
OS2BOS

Magenta ApS

Apr 19, 2021

CONTENTS

1	Introduktion	1
1.1	README	1
1.2	OS2BOS	1
1.3	Technical Documentation	6
1.4	Release History	165
	Python Module Index	179
	HTTP Routing Table	181
	Index	183

INTRODUKTION

BOS er en webapplikation til håndtering af foranstaltninger og bevillinger for kommunens socialrådgivere. Formålet med BOS er at give socialrådgiverne bedre redskaber til at benytte de bedste foranstaltninger og give kommunen som helhed bedre indblik og styringsmulighed i forhold til foranstaltninger, effekter, økonomi og budget.

Navnet BOS dækker over “Bevilling og Styring”.

Nedenstående figur viser et typisk eksempel på en side i systemets brugergrænseflade:

The screenshot displays the BOS web application interface. At the top, there is a header with the logo for Ballerup Kommune, navigation links (Mine sager, Find sager, Betalinger, Administration), and the user profile (IT Guy (admin), S-DIG). Below the header, a sidebar shows the user's name and a search bar. The main content area is divided into two sections. The left section, titled 'Hovedsag 27.24.00-1-G01', contains a table with details about the case, including the case number, the name of the case manager, the date, and the status. The right section, titled 'Familieoversigt', contains a table with details about the family, including the name, the date, and the status. Below these sections is a table titled 'Bevillingsskrivelser' (Funding Decisions) with columns for Status, Foranstaltningssag, Bevillingsparagraf, Supplerende info, Oprettet, Senest ændret, Udgift i år, and Forventet udgift i år. The table contains two rows of data, each representing a funding decision. The first row shows a decision for '§ SEL-02-3.1 Ophold i dagtilbud, fritidshjem, ...' with a status of 'Godkendt' and a value of 2.500,00 kr. The second row shows a decision for '§ SEL-02-3.5 Aftalingsordning' with a status of 'Godkendt' and a value of 10.000,00 kr. At the bottom right of the table, there is a summary row showing the total value of 12.500,00 kr.

Status	Foranstaltningssag	Bevillingsparagraf	Supplerende info	Oprettet	Senest ændret	Udgift i år	Forventet udgift i år
Godkendt	27.36.08-G01-13-14	§ SEL-02-3.1 Ophold i dagtilbud, fritidshjem, ...		16. jan 2020	16. jan 2020	2.500,00 kr.	
Godkendt	27.36.08-G01-13-15	§ SEL-02-3.5 Aftalingsordning	Aftalning	16. jan 2020	16. jan 2020	10.000,00 kr.	
						Samt	12.500,00 kr

Indholdet er opdelt i følgende sektioner:

1.1 README

1.2 OS2BOS

1.2.1 Installation

TL;DR: To get a running development environment run:

```
git clone git@git.magenta.dk:bevillingsplatform/bevillingsplatform.git
cd bevillingsplatform
docker-compose up -d
```

You can now reach the frontend at <http://localhost:8080>. The frontend will proxy all request to the backend.

Run backend tests with:

```
docker-compose exec bev pytest
```

Docker

The repository contains a `Dockerfile`. This is the recommended way to install bevillingsplatform both as a developer and in production.

All releases are pushed to Docker Hub at [magentaaps/bevillingsplatform](https://hub.docker.com/r/magentaaps/bevillingsplatform) under the `latest` tag.

To run bevillingsplatform in docker you need a running docker daemon. To install docker we refer you to the [official documentation](#).

To configure the django inside the image, add your setting to `/code/settings.ini`. This is easiest done by binding a file from the host machine to this file.

The container requires a connection to a postgres database server. It is configured with the `DATABASE_*` settings. The database server must have a user and a database object. It can be created with the help of `/docker/postgres-initdb.d/20-create-db-and-user.sh`. This file can easily be mounted into `/docker-entrypoint-initdb.d/` in the [official postgres docker image](#).

You can start the container with:

```
docker run -p 8000:8000 -v $PWD/dev-settings.ini:/code/settings.ini magentaaps/
↪bevillingsplatform:latest
```

This will pull the image from Docker Hub and starts a container in the foreground. The `-p 8000:8000` binds port 8000 of the host machine to port 8000 on the container. The `-v $PWD/dev-settings.ini:/code/settings.ini` binds the `dev-settings.ini` file into the container at the location where django will pick it up.

If successful you should see the container migrating the database and finally

```
[2019-06-13 09:18:48 +0000] [1] [INFO] Listening at: http://0.0.0.0:8000 (1)
```

when the gunicorn server starts up. You should now be able to reach the server from the host at `http://localhost:8000`.

If you continue to see `01/30 Unable to connect to database.` your database configuration most likely wrong. Remember if you set `DATABASE_HOST=localhost` the container will try to connect to a database in the same container, not the host machine.

Static files

The docker image will serve the static files for the Vue frontend, Django REST framework and Django Admin from gunicorn both in development and in production. Normally it is not good practice to serve static files from gunicorn for security and performance reasons. We use [whitenoise](#) to address most of these concerns and generally don't expect many users. If you still want to serve it from another service, all the files are copied to `/static` on container startup. This can easily be mounted to a webserver.

Logs

For logging we use the builtin logging module used by [Django](#). Logs are written to the `/log/` directory inside the container.

The `/log/` folder is mapped to the host systems `log/` folder in local development and `/var/log/os2bos/` anywhere else.

We log the following:

- Error log - the gunicorn error log is output on `STDERR` as well as in `/log/error.log`. It can be inspected with `docker logs`.
- Access log - The gunicorn access log is output on `STDOUT` as well as in `/log/access.log`. It can be inspected with `docker logs`.
- Django debug log - The django log is written to `/log/django-debug.log`.
- Audit log - We log all non-idempotent requests (`POST/PUT/PATCH/DELETE`) in the audit log specifying the request endpoint and user making the request. The log is written to `/log/audit.log`.
- Export to Prism log - Log related to the PRISM exports management command. The log is written to `/log/export_to_prism.log`.
- Mark Payments Paid log - Log related to the marking payments paid management command. The log is written to `mark_payments_paid.log`.
- Generate Payments Report log - Log related to generating the payment reports. The log is written to `/log/generate_payments_report.log`.
- Cron mail logs - Logs related to the [Django mailer app](#). The logs are written to `cron_mail.log`, `cron_mail_deferred.log`, `cron_mail_purge.log`.

User permissions

The *Dockerfile* creates and runs the application as the *bev* user. This user will own all the files generated by the application. This user has a `UID` and `GID` of 72050.

If you want to use another `UID/GID`, you can specify it as the `--user=uid:gid` [overwrite flag](#). for the `docker run` command or in [docker-compose](#). If you change the `UID/GID`, the `/log` and `/static` volumes may not have the right permissions. It is recommended to only use [bind](#) if you overwrite the user and set the same user as owner of the directory you bind.

If some process inside the container needs to write files to locations other than `/static` or `/log`, you need to mount a volume with the right permissions. An example is `./manage.py makemigrations` trying to write to `/code/backend/core/migrations`. If you bind `/code` to your host system, make sure that the user with `UID 72050` have write permissions to `backend/core/migrations`. This can be done with `chmod o+w migrations` on your host where you grant all user permission to write.

Test

All the requirements for tests included in the docker image. You can run the test from inside a container with `pytest`.

tox

`tox` is also installed, but it tries to create a virtual environments inside the container. This is messy and will fail because the application user does not have permission to write files. Don't use `tox` inside the container.

Docker-compose

You can use `docker-compose` to start up bevillingsplatform and related services such as postgres and postfix.

A `docker-compose.yml` for development is included. It includes the settings to connect them. It starts the following services:

- *frontend*: the vue frontend reachable at <http://localhost:8080>
- *bev*: the django backend
- *db*: a OS2BOS specific [postgres database server](#)
- *bev-cron*: [supercronic](#), a job runner running our cronjobs
- *idp*: a local version of the IdP running our version of [SimpleSAMLphp](#)
- *postfix*: a [postfix email server](#)

Normally the backend image also serves the frontend code, but to ease frontend development, we include a frontend service that run [vue-cli-service serve](#). The frontend proxies requests to the backend. The exact list of proxied endpoints can be seen in `frontend/vue.config.js`.

`docker-compose.yml` also mounts the current directory in the container and automatically restarts the server on changes to the backend files. This enables you to edit the backend files and the server will be reloaded automatically.

To pull the images and start the services run:

```
docker-compose up -d --build
```

The `-d` flag move the services to the background. You can inspect the output of them with `docker-compose logs <name>` where `<name>` is the name of the service in `docker-compose.yml`. The `--build` flag builds the newest docker image for *bevillingsplatform* from the local `Dockerfile`.

To stop the service again run `docker-compose stop`. This will stop the services, but the data will persist. To completely remove the containers and data run `docker-compose down -v`.

1.2.2 Tests and shell access

To run the backend test, execute: `docker-compose exec bev ./manage.py test`. It will connect to the running docker container and execute the tests.

To get shell access to the backend run `docker-compose exec bev bash`.

If you want to write files from inside the container, make sure the *bev* user have permission to do so. See [User permissions](#).

Tests can also be executed locally with `tox`:


```
tox -e test
```

1.2.3 Code coverage

We adhere to a code coverage of 100%.

After running the test-suite a coverage report can be generated locally with tox:

```
tox -e coverage
```

1.2.4 Documentation

The documentation exists at [Read the Docs](#) and can be generated locally with tox:

```
tox -e docs
```

When changes are introduced to the Django models, update and commit the database model graph for use in documentation:

```
tox -e graph
```

1.2.5 Code standards

The Python code is enforced with the following standards:

- [black](#)
- [flake8](#)
- [pydocstyle](#) (PEP257)

Adherence to these standards can be checked locally with tox:

```
tox -e lint
```

1.2.6 Licensing

Copyright (c) 2019 Magenta Aps

Bevillingsplatform is free software; you may use, study, modify and distribute it under the terms of version 2.0 of the Mozilla Public License. See the LICENSE file for details. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

All source code in this and the underlying directories is subject to the terms of the Mozilla Public License, v. 2.0.

The core version of the code is located here: <https://github.com/OS2bos/os2bos/>.

1.3 Technical Documentation

1.3.1 Frontend

BOS frontenden er skrevet i Javascript frameworket [Vue.js](#). Brugeren interagerer med denne og frontenden kommunikerer med backenden via et REST API i JSON format.

1.3.2 Backend

BOS backenden er skrevet i [Python](#) og benytter [Django](#) samt [Django Rest Framework](#). Data bliver persisteret i en [PostgreSQL-database](#) og udstilles og manipuleres via et REST API.

Backend documentation

Authentication

Authentication in OS2BOS is based on SAML 2.0 single sign-on. Support for this is implemented through the module [PySAML2](#).

Once a user has been authenticated, a session is created containing the attributes given to us from the IdP, e.g. the Active Directory groups the user belongs to.

SAML based single sign-on (SSO)

SAML SSO delegates the login process to the IdP, and as such, requires an IdP supporting SAML single sign-on.

Settings related to SAML SSO can be found in the `SAML2_AUTH` object in `settings.py`.

User setup

In local development and test, SAML SSO is used with [SimpleSamlPHP](#) as idP.

Users and user profiles are defined in `dev-environment/authsources.php` for local development. If you wish to add a new user, you need to add an entry to the file following this template:

```
'<brugernavn>:<password>' => array(
    'email' => '<email_adresse>',
    'username' => '<brugernavn>',
    'first_name' => '<fornavn>',
    'last_name' => '<efternavn>',
    'team' => '<team (navn)>',
    'bos_profile' => '<profil>'
),
```

A user that is not allowed to use OS2BOS, must have a SAML-token with an empty or missing `bos_profile` entry.

User profiles

There are five user profiles corresponding to five different levels of permissions.

- “readonly” - the user can only read, not write.
- “edit” - the user can both read and write, but cannot perform grant operations.
- “grant” - the user can both read, write and perform grant operations, but cannot access Django admin.
- “workflow_engine” - the user can both read, write and perform grant operations, and can access only classifications in Django admin.
- “admin” - the user can read, write, perform grant operations and access Django admin.

Integrations

Serviceplatformen

We use Serviceplatformen to fetch information from the CPR register. Support for this is implemented through the module `service_person_stamdata_udvidet`.

Settings relevant to Serviceplatformen can be found in the `SERVICEPLATFORM_UUIDS` object and the `SERVICEPLATFORM_CERTIFICATE_PATH`, `USE_SERVICEPLATFORM`, `USE_SERVICEPLATFORM_PROD` keys in `settings.py`.

PRISME

OS2BOS can export daily payment files for import into the *PRISME* economy system. The format of the files are based on KMD’s interface specification [GF200001Q](#) for creditor records (transaction type G68).

Generating database documentation

The database documentation can be found in an online version here: [os2bosdocs](#).

We generate database documentation using the tool [SchemaSpy](#). The following commands are run from inside the container as root: `docker-compose exec -u 0 bev bash`

Install a JRE:

```
apt-get install default-jre
```

Install graphviz:

```
apt-get install graphviz
```

Download the latest schemaspy.jar

```
wget https://github.com/schemaspy/schemaspy/releases/download/v6.1.0/schemaspy-6.1.0.jar -O schemaspy.jar
```

Download the latest JDBC for PostgreSQL:

```
wget https://jdbc.postgresql.org/download/postgresql-42.2.8.jar -O postgresql.jar
```

Run SchemaSpy against the database:

```
java -jar schemaspy.jar -dp postgresql.jar -t pgsq1 -s public -db bev -host db -u bev_
↪-p bev -o er_html
```

The documentation is now found in the `er_html` folder.

Django debugging

To debug end-to-end you can set the `breakpoint()` you want and use `docker attach`:

```
docker attach bevillingsplatform_bev_1
```

Release procedure

We manage releases using the [Gitflow](#) model.

- Create a release branch for the release, e.g. `release/1.0.0`.
- Bump the version and changelog in `NEWS.rst` and `VERSION` according to [Semantic Versioning](#).
- *Optional*: Check our dependencies for new security bugfixes.
- *Optional*: In case of API changes, save the OpenAPI schema `openapi.yml` from `<OS2BOS URL>/api/openapi/?format=api` for use in documentation.
- *Optional*: In case of model changes, update the database model graph by running `tox -e graph` for use in documentation.
- Test the release and verify that it doesn't contain any breaking bugs or regressions.
- Create an `rc` tag, e.g. `1.0.0-rc1` on the release branch to make it deployable on staging.
- In case of further development, make pull requests against the release branch and create a new `rc` tag.
- Proceed with development and `rc`-tagging on the release branch until the customer is happy.
- Create a pull request from the release branch to `master` for review of `VERSION` and `EWS.rst` bump.
- Create a pull request from the release branch to `develop`.
- Merge the pull requests.
- Create a release tag, e.g. `1.0.0` from `master`.
- Push `develop`, `master` and tags to [the OS2BOS Github](#) as the new core version.
- Delete the release branch.


```
change_list_template = 'core/admin/accountaliasmapping/change_list.html'

changelist_view(*args, **kwargs)
    Override changelist_view adding form to context.

get_urls()
    Override get_urls adding upload_url path.

list_display = ('main_account_number', 'activity_number', 'alias')

property media

upload_csv(request)
    Handle the uploaded CSV file.

property urls
    Override get_urls adding upload_url path.

class core.admin.AccountAliasMappingCSVFileUploadForm(data=None, files=None,
    auto_id='id_%s', pre-
    fix=None, initial=None,
    error_class=<class
    'django.forms.utils.ErrorList'>,
    label_suffix=None,
    empty_permitted=False,
    field_order=None,
    use_required_attribute=None,
    renderer=None)

Bases: django.forms.forms.Form

CSV Upload form for AccountAliasMappingAdmin.

Form fields:

    • csv_file: vælg venligst en fil (FileField)

property media
    Return all media required to render the widgets on this form.

class core.admin.ActivityAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    ModelAdmin for Activity.

    account_alias(obj)
        Get account alias.

    account_alias_new(obj)
        Get new account alias.

    account_number(obj)
        Get account number.

    account_number_new(obj)
        Get new account number.

property media

readonly_fields = ('account_number', 'account_number_new', 'account_alias', 'account_a

class core.admin.ActivityCategoryAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for ActivityCategory.
```

```

    inlines = [<class 'core.admin.SectionInfoActivityCategoryInline'>]
    list_display = ('category_id', 'name')
    property media

class core.admin.ActivityDetailsAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin
    Widgets: Filter_horizontal for many to many links, add search field.
    exclude = ('supplementary_activity_for', 'main_activity_for', 'service_providers')
    filter_horizontal = ('main_activities',)
    inlines = (<class 'core.admin.SectionInfoInline'>, <class 'core.admin.SupplementaryAct.
    list_display = ('name', 'activity_id', 'active')
    property media
    search_fields = ('activity_id', 'name')

class core.admin.AppropriationAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    ModelAdmin for Appropriation.
    property media

class core.admin.ApprovalLevelAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for ApprovalLevel.
    list_display = ('name', 'active')
    property media

class core.admin.CaseAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    ModelAdmin for Case.
    property media

class core.admin.ClassificationAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    ModelAdmin for Classification models.
    has_add_permission(request)
        Override has_add_permission for ModelAdmin.
    has_change_permission(request, obj=None)
        Override has_change_permission for ModelAdmin.
    has_delete_permission(request, obj=None)
        Override has_delete_permission for ModelAdmin.
    has_module_permission(request)
        Override has_model_permission for ModelAdmin.
    has_view_permission(request, obj=None)
        Override has_view_permission for ModelAdmin.
    property media

```

```
class core.admin.ClassificationInline (parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline
    TabularInline for Classification inlines.

    has_add_permission (request)
        Override has_add_permission for InlineModelAdmin.

    has_change_permission (request, obj=None)
        Override has_change_permission for InlineModelAdmin.

    has_delete_permission (request, obj=None)
        Override has_delete_permission for InlineModelAdmin.

    has_view_permission (request, obj=None)
        Override has_view_permission for InlineModelAdmin.

    property media

class core.admin.CustomUserAdmin (model, admin_site)
    Bases: django.contrib.auth.admin.UserAdmin
    Add team to user admin interface.

    has_add_permission (request)
        Override has_add_permission for ModelAdmin.

    has_change_permission (request, obj=None)
        Override has_change_permission for ModelAdmin.

    has_delete_permission (request, obj=None)
        Override has_delete_permission for ModelAdmin.

    has_module_permission (request)
        Override has_model_permission for ModelAdmin.

    has_view_permission (request, obj=None)
        Override has_view_permission for ModelAdmin.

    property media

class core.admin.EffortAdmin (model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for Effort.

    filter_horizontal = ('allowed_for_target_groups',)
    list_display = ('name', 'active')
    property media

class core.admin.EffortStepAdmin (model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for EffortStep.

    inlines = (<class 'core.admin.SectionEffortStepProxyInline'>,)
    list_display = ('name', 'list_sections', 'active')
    list_sections (obj)
        HTML list of sections for Django admin purposes.

    property media

    search_fields = ('name', 'number')
```



```

class core.admin.HistoricalRatePerDateInline (parent_model, admin_site)
    Bases: core.admin.ClassificationInline
    HistoricalRatePerDateInline for VariablerateAdmin.

    can_delete = False

    classes = ('collapse',)

    exclude = ('history_change_reason',)

    extra = 0

    has_add_permission (request)
        Override has_add_permission for HistoricalRatePerDateInline.

    has_change_permission (request, obj=None)
        Override has_change_permission for HistoricalRatePerDateInline.

    property media

    model
        alias of core.proxies.HistoricalRatePerDateProxy

    verbose_name = 'Historisk tekst for datoer'

    verbose_name_plural = 'Historiske takster for datoer'

class core.admin.InternalPaymentRecipientAdmin (model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for InternalPaymentRecipient.

    list_display = ('name',)

    property media

class core.admin.MainActivityDetailsInline (parent_model, admin_site)
    Bases: core.admin.ClassificationInline
    MainActivityDetailsInline for SectionAdmin.

    extra = 0

    property media

    model
        alias of core.models.SectionInfo

    ordering = ('activity_details__name',)

    verbose_name = 'Tilladt hovedydelse'

    verbose_name_plural = 'Tilladte hovedydelser'

class core.admin.MunicipalityAdmin (model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for Municipality.

    list_display = ('name', 'active')

    property media

    search_fields = ('name',)

```

```
class core.admin.PaymentAdmin(model, admin_site)
    Bases: simple_history.admin.SimpleHistoryAdmin

    ModelAdmin for Payment.

    account_alias(obj)
        Get account alias.

    account_alias_new(obj)
        Get new account alias.

    account_string(obj)
        Get account string.

    account_string_new(obj)
        Get account string new.

    list_display = ('id', 'payment_id', 'account_string', 'date', 'paid', 'paid_date', 'pa
    list_filter = ('paid', 'payment_schedule__fictive', 'date', 'paid_date', 'payment_meth
    property media
    payment_id(obj)
        Get payment ID from payment plan.

    payment_schedule_str(obj)
        Get related payment schedule link.

    readonly_fields = ('payment_id', 'account_string', 'account_alias', 'account_string_new
    search_fields = ('payment_schedule__payment_id',)

class core.admin.PaymentDateExclusionAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin

    ModelAdmin for PaymentDateExclusion.

    list_display = ('date', 'weekday')

    list_filter = (('date', <class 'django.contrib.admin.filters.DateFieldListFilter'>),)
    property media
    search_fields = ('date',)

    weekday(obj)
        Return translated weekday of date.

class core.admin.PaymentMethodDetails(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    ModelAdmin for PaymentMethodDetails.

    property media

class core.admin.PaymentScheduleAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    Display read only fields on payment schedule.

    account_alias(obj)
        Get account alias.

    account_alias_new(obj)
        Get new account alias.
```

```

    account_string (obj)
        Get account string.

    account_string_new (obj)
        Get account string.

    list_display = ('id', 'payment_id', 'recipient_type', 'recipient_id', 'recipient_name')
    list_filter = ('payment_method', 'payment_type', 'payment_frequency', 'fictive')
    property media
    readonly_fields = ('payment_id', 'account_string', 'account_alias', 'account_string_new')
    search_fields = ('payment_id',)

class core.admin.PriceAdmin(model, admin_site)
    Bases: core.admin.VariableRateAdmin
    ModelAdmin for Price.

    form
        alias of core.admin.PriceForm

    list_display = ('payment_schedule',)
    property media
    readonly_fields = ('payment_schedule',)

class core.admin.PriceForm(*args, **kwargs)
    Bases: core.admin.VariableRateAdminForm
    PriceForm for PriceAdmin.

    property media
        Return all media required to render the widgets on this form.

class core.admin.RateAdmin(model, admin_site)
    Bases: core.admin.VariableRateAdmin
    ModelAdmin for Rate.

    exclude = ('needs_recalculation',)
    form
        alias of core.admin.RateForm

    list_display = ('name', 'description')
    property media

class core.admin.RateForm(*args, **kwargs)
    Bases: core.admin.VariableRateAdminForm
    RateForm for RateAdmin.

    property media
        Return all media required to render the widgets on this form.

class core.admin.RatePerDateInline(parent_model, admin_site)
    Bases: core.admin.ClassificationInline
    RatePerDateInline for VariableRateAdmin.

    can_delete = False
    extra = 0

```

```
    has_add_permission(request)
        Override has_add_permission for RatePerDateInline.

    property media

    model
        alias of core.models.RatePerDate

    readonly_fields = ['rate', 'start_date', 'end_date']

class core.admin.RelatedPersonAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    ModelAdmin for RelatedPerson.

    property media

class core.admin.SchoolDistrictAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin

    ModelAdmin for SchoolDistrict.

    list_display = ('name', 'active')

    property media

class core.admin.SectionAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin

    Add search field.

    filter_horizontal = ('allowed_for_target_groups', 'allowed_for_steps')

    inlines = (<class 'core.admin.MainActivityDetailsInline'>, <class 'core.admin.SupplementaryActDetailsInline'>)

    list_display = ('paragraph', 'text', 'list_main_activity_for', 'list_supplementary_activity_for')

    list_main_activity_for(obj)
        HTML list of main activities for Django admin purposes.

    list_supplementary_activity_for(obj)
        HTML list of supplementary activities for Django admin purposes.

    property media

    search_fields = ('paragraph', 'text')

class core.admin.SectionEffortStepProxyInline(parent_model, admin_site)
    Bases: core.admin.ClassificationInline

    SectionEffortStepProxyInline for EffortStepAdmin.

    extra = 0

    property media

    model
        alias of core.proxies.SectionEffortStepProxy

    verbose_name = 'Tilladt paragraf'

    verbose_name_plural = 'Tilladte paragraffer'

class core.admin.SectionInfoActivityCategoryInline(parent_model, admin_site)
    Bases: core.admin.ClassificationInline

    SectionInfoInline for ActivityDetailsAdmin.
```

```

    extra = 0
    property media
    model
        alias of core.models.SectionInfo
class core.admin.SectionInfoAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for SectionInfo.
    list_display = ('activity_details', 'section', 'kle_number', 'activity_category')
    property media
class core.admin.SectionInfoInline(parent_model, admin_site)
    Bases: core.admin.ClassificationInline
    SectionInfoInline for ActivityDetailsAdmin.
    extra = 0
    property media
    model
        alias of core.models.SectionInfo
    verbose_name = 'Tilladt paragraf (for hovedydelse)'
    verbose_name_plural = 'Tilladte paragraffer (for hovedydelse)'
class core.admin.ServiceProviderAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin
    Add search fields.
    list_display = ('name', 'cvr_number', 'active')
    property media
    search_fields = ('name', 'cvr_number')
class core.admin.SupplementaryActivityDetailsInline(parent_model, admin_site)
    Bases: core.admin.ClassificationInline
    SupplementaryActivityDetailsInline for SectionAdmin.
    extra = 0
    property media
    model
        alias of core.proxies.ActivityDetailsSectionProxy
    ordering = ('activitydetails__name',)
    verbose_name = 'Tilladt følgeudgift'
    verbose_name_plural = 'Tilladte følgeudgifter'
class core.admin.SupplementaryActivityInline(parent_model, admin_site)
    Bases: core.admin.ClassificationInline
    SupplementaryActivityInline for ActivityDetailsAdmin.
    extra = 0
    property media

```

```
model
    alias of core.proxies.ActivityDetailsSectionProxy
verbose_name = 'Tilladt paragraf (for følgeudgift)'
verbose_name_plural = 'Tilladte paragraffer (for følgeudgift)'
class core.admin.TargetGroupAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for TargetGroup with custom ModelForm.
    fields = ('name', 'required_fields_for_case', 'active')
    form
        alias of core.admin.TargetGroupForm
    list_display = ('name', 'active')
    property media
class core.admin.TargetGroupForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm
    Form for TargetGroup to set required_fields_for_case.
    __init__(*args, **kwargs)
        __init__ for TargetGroupForm.
        Set initial value for required_fields_for_case.
    clean_required_fields_for_case()
        Clean required_fields_for_case as comma-separated string.
    property media
        Return all media required to render the widgets on this form.
    required_fields_for_case_choices()
        Define the choices for the required_fields_for_case field.
class core.admin.TeamAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for Team.
    property media
class core.admin.VariableRateAdmin(model, admin_site)
    Bases: core.admin.ClassificationAdmin
    ModelAdmin for VariableRate subclasses.
    form
        alias of core.admin.VariableRateAdminForm
    inlines = [<class 'core.admin.RatePerDateInline'>, <class 'core.admin.HistoricalRatePe
    property media
    save_model(request, obj, form, change)
        Override save_model to set rate after model save.
class core.admin.VariableRateAdminForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm
    Form for handling the specifics of date dependency.
```

```

__init__ (*args, **kwargs)
    __init__ for VariableRateAdminForm.

    Set initial value for rate and start_date.

property media
    Return all media required to render the widgets on this form.

```

core.apps module

Core app configuration.

```

class core.apps.CoreConfig (app_name, app_module)
    Bases: django.apps.config.AppConfig

    The core Django app.

    name = 'core'

    ready ()
        Defer import of signals until ready.

```

core.authentication module

Classes for handling CSRF authentication.

```

class core.authentication.CsrfExemptSessionAuthentication
    Bases: rest_framework.authentication.SessionAuthentication

    Override standard session to exempt request from CSRF validation.

    This is OK from a security perspective since a new SAML token is issued with each API request.

    enforce_csrf (request)
        Override this function to do nothing.

```

core.filters module

Filters and filtersets for filtering in REST API.

Filters allow us to do basic search for objects on allowed field without adding the complexity of an entire search engine nor of custom queries.

```

class core.filters.ActivityFilter (data=None, queryset=None, *, relationship=None,
                                   **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet

    Filter activities on status.

    auto_filters = {}

    base_filters = {'status': <django_filters.filters.ChoiceFilter object>}

    declared_filters = {}

    related_filters = {}

class core.filters.AllowedForStepsFilter (data=None, queryset=None, *, relation-
                                          ship=None, **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet

```

Filter sections on allowed effort steps.

```
auto_filters = {}

base_filters = {'active': <django_filters.rest_framework.filters.BooleanFilter object>

declared_filters = {'allowed_for_steps': <django_filters.filters.NumberFilter object>

filter_allowed_for_steps(qs, name, value)
    Filter on "allowed_for_steps" relation.

related_filters = {}

class core.filters.AppropriationFilter(data=None, queryset=None, *, relationship=None,
                                     **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet
    Filter appropriation.

    auto_filters = {}

    base_filters = {'case': <rest_framework_filters.filters.RelatedFilter object>, 'creat

    declared_filters = {'case': <rest_framework_filters.filters.RelatedFilter object>, 'm

    related_filters = {'case': <rest_framework_filters.filters.RelatedFilter object>}

class core.filters.CaseFilter(data=None, queryset=None, *, relationship=None, **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet
    Filter cases on the "expired" field.

    auto_filters = {}

    base_filters = {'acting_municipality': <django_filters.filters.ModelChoiceFilter obje

    declared_filters = {'case_worker': <rest_framework_filters.filters.RelatedFilter obje

    filter_expired(queryset, name, value)
        Filter out cases according to value.

    related_filters = {'case_worker': <rest_framework_filters.filters.RelatedFilter obje

class core.filters.CaseForAppropriationFilter(data=None, queryset=None, *, relation-
                                             ship=None, **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet
    Filter cases on CPR number, team and case_worker.

    auto_filters = {}

    base_filters = {'case_worker': <django_filters.filters.ModelChoiceFilter object>, 'ca

    declared_filters = {}

    related_filters = {}

class core.filters.CaseForPaymentFilter(data=None, queryset=None, *, relationship=None,
                                       **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet
    Filter cases on CPR number.

    auto_filters = {}

    base_filters = {'cpr_number': <django_filters.filters.CharFilter object>}

    declared_filters = {}
```



```

    related_filters = {}

class core.filters.PaymentFilter (data=None,    queryset=None,    *,    relationship=None,
                                **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet
    Filter payments on payment plan, activity, case, dates, etc.

    auto_filters = {}

    base_filters = {'activity':    <rest_framework_filters.filters.RelatedFilter object>, 'a
    declared_filters = {'activity':    <rest_framework_filters.filters.RelatedFilter object>

    filter_paid_date_or_date_gte (queryset, name, value)
        Filter on value <= "best known payment date".

    filter_paid_date_or_date_lte (queryset, name, value)
        Filter on value >= "best known payment date".

    filter_paid_date_or_date_month (queryset, name, value)
        Filter best known payment date on previous, current, next month.

    filter_paid_date_or_date_week (queryset, name, value)
        Filter best known payment date on previous, current, next week.

    filter_paid_date_or_date_year (queryset, name, value)
        Filter best known payment date on previous, current, next year.

    generic_time_choices = (('previous', 'Forrige'), ('current', 'Nuværende'), ('next', 'N

    related_filters = {'activity':    <rest_framework_filters.filters.RelatedFilter object>,

class core.filters.PaymentScheduleFilter (data=None,    queryset=None,    *,    relation-
                                         ship=None, **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet
    Filter payment plans on ID.

    auto_filters = {}

    base_filters = {'payment_id':    <django_filters.filters.NumberFilter object>}

    declared_filters = {}

    related_filters = {}

class core.filters.UserForCaseFilter (data=None,    queryset=None,    *,    relationship=None,
                                      **kwargs)
    Bases: rest_framework_filters.filterset.FilterSet
    Filter cases on case worker team.

    auto_filters = {}

    base_filters = {'team':    <django_filters.filters.ModelChoiceFilter object>}

    declared_filters = {}

    related_filters = {}

```

core.managers module

Custom query set managers.

```
class core.managers.ActivityQuerySet (model=None, query=None, using=None, hints=None)  
    Bases: django.db.models.query.QuerySet
```

QuerySet and Manager for the Activity model.

```
expired()
```

Only include expired activities.

```
ongoing()
```

Only include ongoing, i.e. non-expired Activities.

```
class core.managers.AppropriationQuerySet (model=None, query=None, using=None, hints=None)  
    Bases: django.db.models.query.QuerySet
```

QuerySet and Manager for the Appropriation model.

```
annotate_main_activity_details_id()
```

Annotate the main_activity__details__id as a subquery.

```
expired()
```

Only include expired Appropriations.

```
ongoing()
```

Only include ongoing, i.e. non-expired Appropriations.

```
class core.managers.CaseQuerySet (model=None, query=None, using=None, hints=None)  
    Bases: django.db.models.query.QuerySet
```

Distinguish between expired and ongoing cases.

```
expired()
```

Only include expired Cases.

```
ongoing()
```

Only include ongoing, i.e. non-expired Cases.

```
class core.managers.PaymentQuerySet (model=None, query=None, using=None, hints=None)  
    Bases: django.db.models.query.QuerySet
```

Handle payments properly - some are paid and others are not.

In general we should use the scheduled day and amount for unpaid payments and the paid date and paid amount for paid ones.

```
amount_case = <Case: CASE WHEN <Q: (AND: ('paid_amount__isnull', False))> THEN F(paid
```

```
amount_sum()
```

Sum over Payments with paid_amount overruling amount.

```
annotate_paid_date_or_date()
```

Annotate all payments with paid date or payment date.

```
expected_payments_for_report_list()
```

Filter payments for a report of granted AND expected payments.

```
group_by_monthly_amounts()
```

Group by monthly amounts.

The output will look like this:

```
[
    {'date_month': '2019-07', 'amount': Decimal('3000.00')},
    {'date_month': '2019-08', 'amount': Decimal('1500.00')}
]
```

in_this_year()

Filter Payments only in the current year.

paid_date_or_date_case = <Case: CASE WHEN <Q: (AND: ('paid_date__isnull', False))> TH

paid_date_or_date_gte(date)

Return all payments with paid date or payment date >= date.

paid_date_or_date_lte(date)

Return all payments with paid date or payment date <= date.

strict_amount_sum()

Sum over Payments amount.

core.mixins module

Mixins to use in view classes.

class core.mixins.AuditMixin

Bases: object

Allow audit logging by intercepting all API requests.

finalize_response(request, response, *args, **kwargs)

Perform logging and continue normal operations.

log_methods = ('post', 'put', 'patch', 'delete')

logger = <Logger bevellingsplatform.audit (INFO)>

class core.mixins.AuditModelMixin(*args, **kwargs)

Bases: django.db.models.base.Model

Mixin for tracking created/modified datetime and user.

Parameters

- **created**(DateTimeField) – Oprettet
- **modified**(DateTimeField) – Modificeret
- **user_created**(CharField) – Bruger der har oprettet
- **user_modified**(CharField) – Bruger der har modificeret

created

Model field: oprettet

modified

Model field: modificeret

user_created

Model field: bruger der har oprettet

user_modified

Model field: bruger der har modificeret

```
class core.mixins.AuditModelViewSetMixin
    Bases: object

    Mixin for AuditModel Viewsets.

    perform_create (serializer)
        Set user_created on creation.

    perform_update (serializer)
        Set user_modified on modification.

class core.mixins.ClassificationViewSetMixin
    Bases: object

    Superclass for Classification Viewsets only exposing the active.

    get_queryset ()
        Only expose active objects if user is not workflow or admin.
```

core.models module

These are the Django models, defining the database layout.

```
class core.models.AccountAlias (*args, **kwargs)
    Bases: django.db.models.base.Model

    Model that serves as a mapping from an account string to account alias.

    Map account string from PaymentSchedule to an “account alias” used by Ballerup.

    Parameters

    • id (AutoField) – Id
    • section_info (ForeignKey to SectionInfo) – Paragraf-info
    • activity_details (ForeignKey to ActivityDetails) – Aktivitetsdetalje
    • alias (CharField) – Kontoalias

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

activity_details
    Model field: aktivitetsdetalje, accesses the ActivityDetails model.

activity_details_id
    Model field: aktivitetsdetalje

alias
    Model field: kontoalias

id
    Model field: ID

objects = <django.db.models.manager.Manager object>

section_info
    Model field: paragraf-info, accesses the SectionInfo model.
```

section_info_id

Model field: paragraf-info

class `core.models.AccountAliasMapping(*args, **kwargs)`

Bases: `django.db.models.base.Model`

New version of the AccountAlias model.

Works as a lookup between the main_account and activity dimension strings and an alias.

Parameters

- **id** (*AutoField*) – Id
- **main_account_number** (*CharField*) – Hovedkontonummer. dimensionen 'Hovedkonto' i kontostrengen
- **activity_number** (*CharField*) – Aktivitetsnummer. dimensionen 'Aktivitet' i kontostrengen
- **alias** (*CharField*) – Kontoalias

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

activity_number

Model field: aktivitetsnummer

alias

Model field: kontoalias

id

Model field: ID

main_account_number

Model field: hovedkontonummer

objects = `<django.db.models.manager.Manager object>`

class `core.models.Activity(*args, **kwargs)`

Bases: `core.mixins.AuditModelMixin`, `django.db.models.base.Model`

An activity is a specific service provided within an appropriation.

The details object contains the name, tolerance, etc. of the service.

Parameters

- **id** (*AutoField*) – Id
- **created** (*DateTimeField*) – Oprettet
- **modified** (*DateTimeField*) – Modificeret
- **user_created** (*CharField*) – Bruger der har oprettet
- **user_modified** (*CharField*) – Bruger der har modificeret
- **details** (ForeignKey to *ActivityDetails*) – Aktivitetsdetalje
- **status** (*CharField*) – Status
- **approval_level** (ForeignKey to *ApprovalLevel*) – Bevillingsniveau
- **approval_note** (*TextField*) – Evt. bemærkning

- **approval_user** (ForeignKey to *User*) – Bevilget af bruger
- **appropriation_date** (*DateField*) – Bevillingsdato
- **start_date** (*DateField*) – Startdato
- **end_date** (*DateField*) – Slutdato
- **activity_type** (*CharField*) – Type
- **modifies** (ForeignKey to *Activity*) – Justeres af aktivitet
- **appropriation** (ForeignKey to *Appropriation*) – Bevilling
- **service_provider** (ForeignKey to *ServiceProvider*) – Leverandør
- **note** (*TextField*) – Note

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

property account_alias

Calculate the `account_alias` to use with this activity.

property account_alias_new

Calculate the new `account_alias` to use with this activity.

property account_number

Calculate the `account_number` to use with this activity.

property account_number_new

Calculate the `account_number_new` to use with this activity.

TODO: eventually replace `account_number` with this.

property activity_category

Get the activity category of this activity.

activity_type

Model field: type

property applicable_payments

Return payments that are not overruled by expected payments.

appropriation

Model field: bevilling, accesses the *Appropriation* model.

appropriation_date

Model field: bevillingsdato

appropriation_id

Model field: bevilling

approval_level

Model field: bevillingsniveau, accesses the *ApprovalLevel* model.

approval_level_id

Model field: bevillingsniveau

approval_note

Model field: evt. bemærkning

approval_user
Model field: bevilget af bruger, accesses the *User* model.

approval_user_id
Model field: bevilget af bruger

details
Model field: aktivitetsdetalje, accesses the *ActivityDetails* model.

details_id
Model field: aktivitetsdetalje

end_date
Model field: slutdato

get_activity_type_display (*, field=<django.db.models.fields.CharField: activity_type>)
Autogenerated: Shows the label of the *activity_type*

get_all_modified_by_activities ()
 Retrieve all modified_by objects recursively.

get_status_display (*, field=<django.db.models.fields.CharField: status>)
Autogenerated: Shows the label of the *status*

grant (approval_level, approval_note, approval_user)
 Grant this activity - update payment info as needed.

id
Model field: ID

is_valid_activity_start_date ()
 Determine if a date is valid for a activity start_date.

modified_by
Model field: justeres af aktivitet, accesses the M2M *Activity* model.

modifies
Model field: justeres af aktivitet, accesses the *Activity* model.

modifies_id
Model field: justeres af aktivitet

property monthly_payment_plan
 Calculate the payment plan for this activity, grouped by months.

note
Model field: note

objects = <django.db.models.manager.ManagerFromActivityQuerySet object>

payment_plan
Model field: aktivitet, accesses the *PaymentSchedule* model.

save (*args, **kwargs)
 Save activity.
 Also updates “modified” field on appropriation and payment_plan payments.

service_provider
Model field: leverandør, accesses the *ServiceProvider* model.

service_provider_id
Model field: leverandør

start_date

Model field: startdato

status

Model field: status

property total_cost

Calculate the total cost of this activity at all times.

property total_cost_full_year

Retrieve total amount expected for this year.

Extrapolate for the full year (January 1 - December 31).

property total_cost_this_year

Calculate total cost this year for this activity.

property total_expected_this_year

Calculate total amount *expected* this year.

As may be noted, this is redundant and identical to “total cost this year”. Maybe one of these functions should be removed, the redundancy is due to a desire for clarity as to what is returned in different contexts.

property total_granted_this_year

Calculate total amount *granted* on this activity this year.

property triggers_payment_email

Decide if this activity triggers an email when saved.

validate_expected()

Validate this is a correct expected activity.

property vat_factor

Calculate the VAT factor for this activity.

class core.models.ActivityCategory(*args, **kwargs)

Bases: core.models.Classification

ActivityCategory associated with a SectionInfo (main_activity).

Parameters

- **id** (AutoField) – Id
- **active** (BooleanField) – Aktiv
- **category_id** (CharField) – Kategori id
- **name** (CharField) – Navn

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

category_id

Model field: kategori ID

id

Model field: ID

name

Model field: navn

objects = <django.db.models.manager.Manager object>

section_infos

Model field: aktivitetskategori, accesses the M2M *SectionInfo* model.

class `core.models.ActivityDetails(*args, **kwargs)`

Bases: `core.models.Classification`

Class containing all types of activities offered.

Each type is associated with the legal sections for which it is allowed as well as the allowed main activitydetails and service providers.

Parameters

- **id** (*AutoField*) – Id
- **active** (*BooleanField*) – Aktiv
- **name** (*CharField*) – Navn
- **description** (*TextField*) – Beskrivelse
- **activity_id** (*CharField*) – Aktivitets id
- **max_tolerance_in_percent** (*PositiveSmallIntegerField*) – Max tolerance i procent
- **max_tolerance_in_dkk** (*PositiveIntegerField*) – Max tolerance i dkk
- **main_activity_for** (*ManyToManyField*) – Hovedaktivitet for paragraffer
- **supplementary_activity_for** (*ManyToManyField*) – Følgeudgift for paragraffer
- **service_providers** (*ManyToManyField*) – Leverandører
- **main_activities** (*ManyToManyField*) – Hovedydelse. Denne aktivitetsdetalje kan være følgeudgift for disse hovedydelse.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

account_aliases

Model field: aktivitetsdetalje, accesses the M2M *AccountAlias* model.

activity_id

Model field: aktivitets ID

activity_set

Model field: aktivitetsdetalje, accesses the M2M *Activity* model.

description

Model field: beskrivelse

id

Model field: ID

main_activities

Model field: hovedydelse, accesses the M2M *ActivityDetails* model.

main_activity_for

Model field: hovedaktivitet for paragraffer, accesses the M2M *ActivityDetails* model.

max_tolerance_in_dkk
Model field: max tolerance i DKK

max_tolerance_in_percent
Model field: max tolerance i procent

name
Model field: Navn

objects = <django.db.models.manager.Manager object>

sectioninfo_set
Model field: aktivitetsdetalje, accesses the M2M *SectionInfo* model.

service_providers
Model field: leverandører, accesses the M2M *ActivityDetails* model.

supplementary_activities
Model field: hovedydelse, accesses the M2M *ActivityDetails* model.

supplementary_activity_for
Model field: følgeudgift for paragraffer, accesses the M2M *ActivityDetails* model.

class core.models.**Appropriation** (*args, **kwargs)
Bases: *core.mixins.AuditModelMixin*, *django.db.models.base.Model*
An appropriation of funds in a Case - corresponds to a Sag in SBSYS.

Parameters

- **id** (*AutoField*) – Id
- **created** (*DateTimeField*) – Oprettet
- **modified** (*DateTimeField*) – Modificeret
- **user_created** (*CharField*) – Bruger der har oprettet
- **user_modified** (*CharField*) – Bruger der har modificeret
- **sbsys_id** (*CharField*) – Sbsys-id
- **section** (*ForeignKey* to *Section*) – Paragraf
- **case** (*ForeignKey* to *Case*) – Sag
- **note** (*TextField*) – Supplerende oplysninger

exception **DoesNotExist**
Bases: *django.core.exceptions.ObjectDoesNotExist*

exception **MultipleObjectsReturned**
Bases: *django.core.exceptions.MultipleObjectsReturned*

activities
Model field: bevilling, accesses the M2M *Activity* model.

case
Model field: sag, accesses the *Case* model.

case_id
Model field: sag

grant (*activities*, *approval_level*, *approval_note*, *approval_user*)
Grant all the given Activities.

property granted_from_date

Retrieve the start date of the main activity, if granted.

property granted_to_date

Retrieve the end date of the main activity, if granted.

id

Model field: ID

property main_activity

Return main activity, if any.

note

Model field: supplerende oplysninger

objects = <django.db.models.manager.ManagerFromAppropriationQuerySet object>

sbsys_id

Model field: SBSYS-ID

section

Model field: paragraf, accesses the *Section* model.

section_id

Model field: paragraf

property section_info

Return info for law section justifying this appropriation.

property status

Calculate appropriation status from status of activities.

property supplementary_activities

Return supplementary activities, if any.

property total_cost_expected

Retrieve total amount expected.

property total_cost_granted

Retrieve total amount granted.

property total_expected_full_year

Retrieve total amount expected for this year.

Extrapolate for the full year (January 1 - December 31).

property total_expected_this_year

Retrieve total amount expected this year for this Appropriation.

We take into account granted payments but overrule with expected amounts if they modify another activity.

property total_granted_full_year

Retrieve total amount granted for this year.

Extrapolate for the full year (January 1 - December 31).

property total_granted_this_year

Retrieve total amount granted this year for this Appropriation.

This includes both main and supplementary activities.

class core.models.ApprovalLevel (*args, **kwargs)

Bases: *core.models.Classification*

Organizational level on which an appropriation was approved.

Parameters

- **id** (*AutoField*) – Id
- **active** (*BooleanField*) – Aktiv
- **name** (*CharField*) – Navn

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

approved_activities

Model field: bevillingsniveau, accesses the M2M *Activity* model.

id

Model field: ID

name

Model field: navn

objects = `<django.db.models.manager.Manager object>`

class `core.models.Case` (*args, **kwargs)

Bases: `core.mixins.AuditModelMixin`, `django.db.models.base.Model`

A Case, covering one child - corresponding to a Hovedsag in SBSYS.

Parameters

- **id** (*AutoField*) – Id
- **created** (*DateTimeField*) – Oprettet
- **modified** (*DateTimeField*) – Modificeret
- **user_created** (*CharField*) – Bruger der har oprettet
- **user_modified** (*CharField*) – Bruger der har modificeret
- **sbsys_id** (*CharField*) – Sbsys-id
- **cpr_number** (*CharField*) – Cpr-nummer
- **name** (*CharField*) – Navn
- **case_worker** (ForeignKey to *User*) – Sagsbehandler
- **district** (ForeignKey to *SchoolDistrict*) – Skoledistrikt
- **paying_municipality** (ForeignKey to *Municipality*) – Betalingskommune
- **acting_municipality** (ForeignKey to *Municipality*) – Handlekommune
- **residence_municipality** (ForeignKey to *Municipality*) – Bopælskommune
- **target_group** (ForeignKey to *TargetGroup*) – Målgruppe
- **effort_step** (ForeignKey to *EffortStep*) – Indsatstrappe
- **scaling_step** (*PositiveSmallIntegerField*) – Skaleringstrappe
- **assessment_comment** (*TextField*) – Supplerende oplysninger til vurdering
- **note** (*TextField*) – Note
- **efforts** (*ManyToManyField*) – Indsatser

```

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

acting_municipality
    Model field: handlekommune, accesses the Municipality model.

acting_municipality_id
    Model field: handlekommune

appropriations
    Model field: sag, accesses the M2M Appropriation model.

assessment_comment
    Model field: supplerende oplysninger til vurdering

case_worker
    Model field: sagsbehandler, accesses the User model.

case_worker_id
    Model field: sagsbehandler

cpr_number
    Model field: cpr-nummer

district
    Model field: skoledistrikt, accesses the SchoolDistrict model.

district_id
    Model field: skoledistrikt

effort_step
    Model field: indsatsstrappe, accesses the EffortStep model.

effort_step_id
    Model field: indsatsstrappe

efforts
    Model field: indsatser, accesses the M2M Case model.

property expired
    Determine if this case has expired.

    A case is considered to be expired if all its associated activities have end date in the past. If it does not have
    any associated (non-deleted) activities, it is not considered to be expired.

get_scaling_step_display (*, field=<django.db.models.fields.PositiveSmallIntegerField: scaling_step>)
    Autogenerated: Shows the label of the scaling_step

history = <simple_history.manager.HistoryManager object>

id
    Model field: ID

name
    Model field: navn

note
    Model field: note

objects = <django.db.models.manager.ManagerFromCaseQuerySet object>

```

paying_municipality
Model field: betalingskommune, accesses the *Municipality* model.

paying_municipality_id
Model field: betalingskommune

related_persons
Model field: hovedsag, accesses the M2M *RelatedPerson* model.

residence_municipality
Model field: bopælskommune, accesses the *Municipality* model.

residence_municipality_id
Model field: bopælskommune

save_without_historical_record (*args, **kwargs)
Save model without saving a historical record

Make sure you know what you're doing before you use this method.

sbsys_id
Model field: SBSYS-ID

scaling_step
Model field: skaleringstrappe

target_group
Model field: målgruppe, accesses the *TargetGroup* model.

target_group_id
Model field: målgruppe

class core.models.**Classification** (*args, **kwargs)
Bases: django.db.models.base.Model

Abstract base class for Classifications.

Parameters **active** (*BooleanField*) – Aktiv

active
Model field: aktiv

class core.models.**Effort** (*args, **kwargs)
Bases: *core.models.Classification*

Effort for a Case.

Parameters

- **id** (*AutoField*) – Id
- **active** (*BooleanField*) – Aktiv
- **name** (*CharField*) – Navn
- **description** (*CharField*) – Beskrivelse
- **allowed_for_target_groups** (*ManyToManyField*) – Målgrupper

exception **DoesNotExist**
Bases: django.core.exceptions.ObjectDoesNotExist

exception **MultipleObjectsReturned**
Bases: django.core.exceptions.MultipleObjectsReturned

```

allowed_for_target_groups
    Model field: målgrupper, accesses the M2M Effort model.

cases
    Model field: indsatser, accesses the M2M Case model.

description
    Model field: beskrivelse

id
    Model field: ID

name
    Model field: navn

objects = <django.db.models.manager.Manager object>

class core.models.EffortStep(*args, **kwargs)
    Bases: core.models.Classification

    Evaluation step for grading the effort deemed necessary in a Case.

    Parameters
        • id (AutoField) – Id
        • active (BooleanField) – Aktiv
        • name (CharField) – Navn
        • number (PositiveIntegerField) – Nummer

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

case_set
    Model field: indsatstrappe, accesses the M2M Case model.

id
    Model field: ID

name
    Model field: navn

number
    Model field: Nummer

objects = <django.db.models.manager.Manager object>

sections
    Model field: trin i indsatstrappen, accesses the M2M Section model.

class core.models.HistoricalCase(id, created, modified, user_created, user_modified, scaling_step, assessment_comment, case_worker, effort_step, history_id, history_date, history_change_reason, history_type, history_user)
    Bases: simple_history.models.HistoricalChanges, django.db.models.base.Model

    Parameters
        • id (IntegerField) – Id
        • created (DateTimeField) – Oprettet

```

- **modified** (*DateTimeField*) – Modificeret
- **user_created** (*CharField*) – Bruger der har oprettet
- **user_modified** (*CharField*) – Bruger der har modificeret
- **scaling_step** (*PositiveSmallIntegerField*) – Skaleringstrappe
- **assessment_comment** (*TextField*) – Supplerende oplysninger til vurdering
- **case_worker** (ForeignKey to *User*) – Sagsbehandler
- **effort_step** (ForeignKey to *EffortStep*) – Indsatstrappe
- **history_id** (*AutoField*) – History id
- **history_date** (*DateTimeField*) – History date
- **history_change_reason** (*CharField*) – History change reason
- **history_type** (*CharField*) – History type
- **history_user** (ForeignKey to *User*) – History user

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

assessment_comment

Model field: supplerende oplysninger til vurdering

case_worker

Model field: sagsbehandler, accesses the *User* model.

case_worker_id

Model field: sagsbehandler

created

Model field: oprettet

effort_step

Model field: indsatsstrappe, accesses the *EffortStep* model.

effort_step_id

Model field: indsatsstrappe

static get_default_history_user (*instance*)

Returns the user specified by *get_user* method for manually creating historical objects

get_history_type_display (*, *field*=<*django.db.models.fields.CharField: history_type*>)

Autogenerated: Shows the label of the *history_type*

get_next_by_history_date (*, *field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next*=True, ***kwargs*)

Autogenerated: Finds next instance based on *history_date*.

get_previous_by_history_date (*, *field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next*=False, ***kwargs*)

Autogenerated: Finds previous instance based on *history_date*.

get_scaling_step_display (*, *field*=<*django.db.models.fields.PositiveSmallIntegerField: scaling_step*>)

Autogenerated: Shows the label of the *scaling_step*


```

history_change_reason
    Model field: history change reason

history_date
    Model field: history date

history_id
    Model field: history id

history_object

history_type
    Model field: history type

history_user
    Model field: history user, accesses the User model.

history_user_id
    Model field: history user

id
    Model field: ID

property instance

instance_type
    alias of core.models.Case

modified
    Model field: modifieret

property next_record
    Get the next history record for the instance. None if last.

objects = <django.db.models.manager.Manager object>

property prev_record
    Get the previous history record for the instance. None if first.

revert_url ()
    URL for this change in the default admin site.

scaling_step
    Model field: skaleringsstrappe

user_created
    Model field: bruger der har oprettet

user_modified
    Model field: bruger der har modificeret

class core.models.HistoricalPayment (id, paid_amount, paid, paid_date, saved_account_alias,
                                     history_id, history_date, history_change_reason, his-
                                     tory_type, history_user)

Bases: simple_history.models.HistoricalChanges, django.db.models.base.Model

Parameters
    • id (IntegerField) – Id
    • paid_amount (DecimalField) – Betalt beløb
    • paid (BooleanField) – Betalt
    • paid_date (DateField) – Betalt på dato

```

- **saved_account_alias** (*CharField*) – Gemt kontoalias
- **history_id** (*AutoField*) – History id
- **history_date** (*DateTimeField*) – History date
- **history_change_reason** (*CharField*) – History change reason
- **history_type** (*CharField*) – History type
- **history_user** (ForeignKey to *User*) – History user

exception DoesNotExist
Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned
Bases: `django.core.exceptions.MultipleObjectsReturned`

static get_default_history_user (*instance*)
Returns the user specified by *get_user* method for manually creating historical objects

get_history_type_display (*, *field*=<*django.db.models.fields.CharField: history_type*>)
Autogenerated: Shows the label of the *history_type*

get_next_by_history_date (*, *field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next=True*, ***kwargs*)
Autogenerated: Finds next instance based on *history_date*.

get_previous_by_history_date (*, *field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next=False*, ***kwargs*)
Autogenerated: Finds previous instance based on *history_date*.

history_change_reason
Model field: history change reason

history_date
Model field: history date

history_id
Model field: history id

history_object

history_type
Model field: history type

history_user
Model field: history user, accesses the *User* model.

history_user_id
Model field: history user

id
Model field: ID

property instance

instance_type
alias of *core.models.Payment*

property next_record
Get the next history record for the instance. *None* if last.

objects = <*django.db.models.manager.Manager* object>

paid
Model field: betalt

paid_amount
Model field: betalt beløb

paid_date
Model field: betalt på dato

property prev_record
Get the previous history record for the instance. *None* if first.

revert_url()
URL for this change in the default admin site.

saved_account_alias
Model field: gemt kontoalias

```
class core.models.HistoricalRatePerDate(id, rate, start_date, end_date, changed_date,
                                         main_rate, changed_by, history_id, history_date,
                                         history_change_reason, history_type, history_user)
```

Bases: simple_history.models.HistoricalChanges, django.db.models.base.Model

Parameters

- **id** (*IntegerField*) – Id
- **rate** (*DecimalField*) – Takst
- **start_date** (*DateField*) – Startdato
- **end_date** (*DateField*) – Slutdato
- **changed_date** (*DateTimeField*) – Changed date
- **main_rate** (ForeignKey to *VariableRate*) – Main rate
- **changed_by** (ForeignKey to *User*) – Ændret af
- **history_id** (*AutoField*) – History id
- **history_date** (*DateTimeField*) – Historik dato
- **history_change_reason** (*CharField*) – History change reason
- **history_type** (*CharField*) – Historik type
- **history_user** (ForeignKey to *User*) – Historik bruger

exception DoesNotExist
Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
Bases: django.core.exceptions.MultipleObjectsReturned

changed_by
Model field: Ændret af, accesses the *User* model.

changed_by_id
Model field: Ændret af

changed_date
Model field: changed date

end_date
Model field: slutdato

static get_default_history_user (*instance*)
Returns the user specified by *get_user* method for manually creating historical objects

get_history_type_display (*, *field*=<*django.db.models.fields.CharField: history_type*>)
Autogenerated: Shows the label of the *history_type*

get_next_by_history_date (*, *field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next=True*, ***kwargs*)
Autogenerated: Finds next instance based on *history_date*.

get_previous_by_history_date (*, *field*=<*django.db.models.fields.DateTimeField: history_date*>, *is_next=False*, ***kwargs*)
Autogenerated: Finds previous instance based on *history_date*.

history_change_reason
Model field: history change reason

history_date
Model field: historik dato

history_id
Model field: history id

history_object

history_type
Model field: historik type

history_user
Model field: historik bruger, accesses the *User* model.

history_user_id
Model field: historik bruger

id
Model field: ID

property instance

instance_type
alias of *core.models.RatePerDate*

main_rate
Model field: main rate, accesses the *VariableRate* model.

main_rate_id
Model field: main rate

property next_record
Get the next history record for the instance. *None* if last.

objects = <*django.db.models.manager.Manager* object>

property prev_record
Get the previous history record for the instance. *None* if first.

rate
Model field: takst

revert_url ()
URL for this change in the default admin site.

start_date
Model field: startdato

```

class core.models.InternalPaymentRecipient(*args, **kwargs)
    Bases: django.db.models.base.Model

    Recipient model for INTERNAL payment activities.

    Parameters
        • id (AutoField) – Id
        • name (CharField) – Navn

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    id
        Model field: ID

    name
        Model field: navn

    objects = <django.db.models.manager.Manager object>

class core.models.Municipality(*args, **kwargs)
    Bases: core.models.Classification

    Represents a Danish municipality.

    Parameters
        • id (AutoField) – Id
        • active (BooleanField) – Aktiv
        • name (CharField) – Navn

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    acts_on
        Model field: handlekommune, accesses the M2M Case model.

    id
        Model field: ID

    name
        Model field: navn

    objects = <django.db.models.manager.Manager object>

    pays_for
        Model field: betalingskommune, accesses the M2M Case model.

    resident_clients
        Model field: bopælskommune, accesses the M2M Case model.

class core.models.Payment(*args, **kwargs)
    Bases: django.db.models.base.Model

    Represent an amount paid to a supplier - amount, recipient, date.

    These may be entered manually, but ideally they should be imported from an accounts payable system.

```

Parameters

- **id** (*AutoField*) – Id
- **date** (*DateField*) – Betalingsdato
- **recipient_type** (*CharField*) – Betalingsmodtager
- **recipient_id** (*CharField*) – Id
- **recipient_name** (*CharField*) – Navn
- **payment_method** (*CharField*) – Betalingsmåde
- **amount** (*DecimalField*) – Beløb
- **paid_amount** (*DecimalField*) – Betalt beløb
- **paid** (*BooleanField*) – Betalt
- **paid_date** (*DateField*) – Betalt på dato
- **note** (*TextField*) – Note
- **saved_account_string** (*CharField*) – Gemt kontostreng
- **saved_account_alias** (*CharField*) – Gemt kontoalias
- **payment_schedule** (ForeignKey to *PaymentSchedule*) – Betalingsplan

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

property account_alias

Return saved account alias if any, else calculate from schedule.

property account_alias_new

Return saved account alias if any, else calculate from schedule.

TODO: eventually replace `account_alias` with this.

property account_string

Return saved account string if any, else calculate from schedule.

property account_string_new

Return saved account string if any, else calculate from schedule.

TODO: eventually replace `account_string` with this.

amount

Model field: `beløb`

date

Model field: `betalingsdato`

get_next_by_date (*, *field*=<*django.db.models.fields.DateField: date*>, *is_next*=True, **kwargs)

Autogenerated: Finds next instance based on *date*.

get_payment_method_display (*, *field*=<*django.db.models.fields.CharField: payment_method*>)

Autogenerated: Shows the label of the *payment_method*

get_previous_by_date (*, *field*=<*django.db.models.fields.DateField: date*>, *is_next*=False, **kwargs)

Autogenerated: Finds previous instance based on *date*.

```

get_recipient_type_display (*, field=<django.db.models.fields.CharField: recipient_type>)
    Autogenerated: Shows the label of the recipient_type

history = <simple_history.manager.HistoryManager object>

id
    Model field: ID

property_is_payable_manually
    Determine whether it is payable manually (in the frontend).

note
    Model field: note

objects = <django.db.models.manager.ManagerFromPaymentQuerySet object>

paid
    Model field: betalt

paid_amount
    Model field: betalt beløb

paid_date
    Model field: betalt på dato

payment_method
    Model field: betalingsmåde

payment_schedule
    Model field: betalingsplan, accesses the PaymentSchedule model.

payment_schedule_id
    Model field: betalingsplan

recipient_id
    Model field: ID

recipient_name
    Model field: Navn

recipient_type
    Model field: betalingsmodtager

save (*args, **kwargs)
    Save this payment - validate payment method and completeness.

save_without_historical_record (*args, **kwargs)
    Save model without saving a historical record

    Make sure you know what you're doing before you use this method.

saved_account_alias
    Model field: gemt kontoalias

saved_account_string
    Model field: gemt kontostreng

class core.models.PaymentDateExclusion (*args, **kwargs)
    Bases: django.db.models.base.Model

    Model for Prism payment exclusion dates.

    Parameters
    • id (AutoField) – Id

```

- **date** (*DateField*) – Dato

exception DoesNotExist
Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned
Bases: `django.core.exceptions.MultipleObjectsReturned`

date
Model field: `date`

get_next_by_date (*, *field*=<*django.db.models.fields.DateField: date*>, *is_next*=True, ***kwargs*)
Autogenerated: Finds next instance based on *date*.

get_previous_by_date (*, *field*=<*django.db.models.fields.DateField: date*>, *is_next*=False, ***kwargs*)
Autogenerated: Finds previous instance based on *date*.

id
Model field: `ID`

objects = <*django.db.models.manager.Manager object*>

class `core.models.PaymentMethodDetails` (**args*, ***kwargs*)
Bases: `django.db.models.base.Model`
Contains extra information about a payment method.

Parameters

- **id** (*AutoField*) – Id
- **tax_card** (*CharField*) – Skattekort

exception DoesNotExist
Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned
Bases: `django.core.exceptions.MultipleObjectsReturned`

get_tax_card_display (*, *field*=<*django.db.models.fields.CharField: tax_card*>)
Autogenerated: Shows the label of the *tax_card*

id
Model field: `ID`

objects = <*django.db.models.manager.Manager object*>

paymentschedule_set
Model field: betalingsmåde detalje, accesses the M2M *PaymentSchedule* model.

tax_card
Model field: skattekort

tax_card_choices = (('MAIN_CARD', 'Hovedkort'), ('SECONDARY_CARD', 'Bikort'))

class `core.models.PaymentSchedule` (**args*, ***kwargs*)
Bases: `django.db.models.base.Model`
Schedule a payment for an Activity.

Parameters

- **id** (*AutoField*) – Id
- **recipient_type** (*CharField*) – Betalingsmodtager

- **recipient_id** (*CharField*) – Id
- **recipient_name** (*CharField*) – Navn
- **payment_method** (*CharField*) – Betalingsmåde
- **payment_method_details** (ForeignKey to *PaymentMethodDetails*) – Betalingsmåde detalje
- **activity** (OneToOneField to *Activity*) – Aktivitet
- **payment_frequency** (*CharField*) – Betalingsfrekvens
- **payment_date** (*DateField*) – Betalingsdato. dette felt er obligatorisk for og angår kun engangsbetalinger
- **payment_day_of_month** (*IntegerField*) – Betales d. dette felt er obligatorisk for og angår kun månedlige betalinger
- **payment_type** (*CharField*) – Betalingstype
- **payment_units** (*DecimalField*) – Betalingsenheder
- **payment_amount** (*DecimalField*) – Beløb
- **fictive** (*BooleanField*) – Fiktiv
- **payment_id** (*PositiveIntegerField*) – Betalings-id
- **payment_cost_type** (*CharField*) – Betalingspristype
- **payment_rate** (ForeignKey to *Rate*) – Takst

BIWEEKLY = 'BIWEEKLY'

COMPANY = 'COMPANY'

DAILY = 'DAILY'

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

FIXED_PRICE = 'FIXED'

GLOBAL_RATE_PRICE = 'GLOBAL_RATE'

INDIVIDUAL_PAYMENT = 'INDIVIDUAL_PAYMENT'

INTERNAL = 'INTERNAL'

MONTHLY = 'MONTHLY'

exception MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

ONE_TIME_PAYMENT = 'ONE_TIME_PAYMENT'

PERSON = 'PERSON'

PER_UNIT_PRICE = 'PER_UNIT'

RUNNING_PAYMENT = 'RUNNING_PAYMENT'

WEEKLY = 'WEEKLY'

property account_alias

Calculate account alias from activity.

property account_alias_new

Calculate new account alias from activity.

property account_string

Calculate account string from activity.

property account_string_new

Calculate account string from activity.

TODO: eventually replace account_string with this.

activity

Model field: aktivitet, accesses the *Activity* model.

activity_id

Model field: aktivitet

calculate_per_payment_amount (*vat_factor, date*)

Calculate amount from payment type and units.

property can_be_paid

Determine whether payments on this schedule can be paid.

It is only allowed to pay payments associated with an approved activity, i.e. status GRANTED.

create_rrule (*start, **kwargs*)

Create a dateutil.rrule for this schedule specifically.

fictive

Model field: fiktiv

generate_payments (*start, end=None, vat_factor=Decimal('100')*)

Generate payments with a start and end date.

get_payment_cost_type_display (*, *field=<django.db.models.fields.CharField: payment_cost_type>*)

Autogenerated: Shows the label of the *payment_cost_type*

get_payment_frequency_display (*, *field=<django.db.models.fields.CharField: payment_frequency>*)

Autogenerated: Shows the label of the *payment_frequency*

get_payment_method_display (*, *field=<django.db.models.fields.CharField: payment_method>*)

Autogenerated: Shows the label of the *payment_method*

get_payment_type_display (*, *field=<django.db.models.fields.CharField: payment_type>*)

Autogenerated: Shows the label of the *payment_type*

get_recipient_type_display (*, *field=<django.db.models.fields.CharField: recipient_type>*)

Autogenerated: Shows the label of the *recipient_type*

id

Model field: ID

static is_payment_and_recipient_allowed (*payment_method, recipient_type*)

Determine if this combination of method and recipient is allowed.

is_ready_to_generate_payments ()

Don't generate payments unless object is ready.

Note: This is *not* a validation. It's necessary because Django Rest Framework can perform partial saves and the post-save logic needs to be able to cope with that.

property next_payment

Return the next payment due starting from today, if any.

objects = <django.db.models.manager.Manager object>

payment_amount

Model field: beløb

payment_cost_choices = (('FIXED', 'fast beløb'), ('PER_UNIT', 'pris pr. enhed'), ('GLO'))

payment_cost_type

Model field: betalingspristype

payment_date

Model field: betalingsdato

payment_day_of_month

Model field: betales d.

payment_frequency

Model field: betalingsfrekvens

payment_frequency_choices = (('DAILY', 'Dagligt'), ('WEEKLY', 'Ugentligt'), ('BIWEEKLY', 'Ugentligt'))

payment_id

Model field: betalings-ID

payment_method

Model field: betalingsmåde

payment_method_details

Model field: betalingsmåde detalje, accesses the *PaymentMethodDetails* model.

payment_method_details_id

Model field: betalingsmåde detalje

payment_rate

Model field: takst, accesses the *Rate* model.

payment_rate_id

Model field: takst

payment_type

Model field: betalingstype

payment_type_choices = (('ONE_TIME_PAYMENT', 'Engangsudgift'), ('RUNNING_PAYMENT', 'Følgeskud'))

payment_units

Model field: betalingsenheder

payments

Model field: betalingsplan, accesses the M2M *Payment* model.

property per_payment_amount

Amount per payment calculated w/ payment cost type.

price_per_unit

Model field: betalingsplan, accesses the *Price* model.

property rate_or_price_amount

Fetch current rate or per unit price amounts.

recalculate_prices()

Recalculate price on all payments.

```
recipient_choices = (('INTERNAL', 'Intern'), ('PERSON', 'Person'), ('COMPANY', 'Firma'))

recipient_id
    Model field: ID

recipient_name
    Model field: navn

recipient_type
    Model field: betalingsmodtager

save (*args, **kwargs)
    Save this payment schedule after validating payment method.

synchronize_payments (start, end, vat_factor=Decimal('100'))
    Synchronize an existing number of payments for a new end_date.

class core.models.Price (*args, **kwargs)
    Bases: core.models.VariableRate
    An individual price on a payment plan.

    Parameters
        • id (AutoField) – Id
        • variable_rate_ptr (OneToOneField to VariableRate) – Variable rate ptr
        • payment_schedule (OneToOneField to PaymentSchedule) – Betalingsplan

exception DoesNotExist
    Bases: core.models.VariableRate.DoesNotExist

exception MultipleObjectsReturned
    Bases: core.models.VariableRate.MultipleObjectsReturned

payment_schedule
    Model field: betalingsplan, accesses the PaymentSchedule model.

payment_schedule_id
    Model field: betalingsplan

variable_rate_ptr
    Model field: variable rate ptr, accesses the VariableRate model.

variable_rate_ptr_id
    Model field: variable rate ptr

class core.models.Rate (*args, **kwargs)
    Bases: core.models.VariableRate, core.models.Classification
    A centrally fixed rate.

    Parameters
        • id (AutoField) – Id
        • variable_rate_ptr (OneToOneField to VariableRate) – Variable rate ptr
        • active (BooleanField) – Aktiv
        • name (CharField) – Navn
        • description (TextField) – Beskrivelse
        • needs_recalculation (BooleanField) – Skal genberegnes. dette felt sættes automatisk når en takst ændres
```

```

exception DoesNotExist
    Bases: core.models.VariableRate.DoesNotExist

exception MultipleObjectsReturned
    Bases: core.models.VariableRate.MultipleObjectsReturned

description
    Model field: beskrivelse

name
    Model field: navn

needs_recalculation
    Model field: skal genberegnes

paymentschedule_set
    Model field: takst, accesses the M2M PaymentSchedule model.

variablerate_ptr
    Model field: variablerate ptr, accesses the VariableRate model.

variablerate_ptr_id
    Model field: variablerate ptr

class core.models.RatePerDate (*args, **kwargs)
    Bases: django.db.models.base.Model

    Handle the date variation of VariableRates.

    Parameters

    • id (AutoField) – Id
    • rate (DecimalField) – Takst
    • start_date (DateField) – Startdato
    • end_date (DateField) – Slutdato
    • main_rate (ForeignKey to VariableRate) – Main rate
    • changed_date (DateTimeField) – Changed date
    • changed_by (ForeignKey to User) – Ændret af

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

changed_by
    Model field: Ændret af, accesses the User model.

changed_by_id
    Model field: Ændret af

changed_date
    Model field: changed date

end_date
    Model field: slutdato

history = <simple_history.manager.HistoryManager object>

```

id
Model field: ID

main_rate
Model field: main rate, accesses the *VariableRate* model.

main_rate_id
Model field: main rate

objects = <django.db.models.manager.Manager object>

rate
Model field: tekst

save_without_historical_record (*args, **kwargs)
Save model without saving a historical record

Make sure you know what you're doing before you use this method.

start_date
Model field: startdato

class core.models.**RelatedPerson** (*args, **kwargs)
Bases: *core.mixins.AuditModelMixin*, *django.db.models.base.Model*

A person related to a Case, e.g. as a parent or sibling.

Parameters

- **id** (*AutoField*) – Id
- **created** (*DateTimeField*) – Oprettet
- **modified** (*DateTimeField*) – Modificeret
- **user_created** (*CharField*) – Bruger der har oprettet
- **user_modified** (*CharField*) – Bruger der har modificeret
- **relation_type** (*CharField*) – Relation
- **cpr_number** (*CharField*) – Cpr-nummer
- **name** (*CharField*) – Navn
- **related_case** (*CharField*) – Sbsys-sag
- **from_serviceplatformen** (*BooleanField*) – Fra serviceplatformen
- **main_case** (*ForeignKey* to *Case*) – Hovedsag

exception **DoesNotExist**
Bases: *django.core.exceptions.ObjectDoesNotExist*

exception **MultipleObjectsReturned**
Bases: *django.core.exceptions.MultipleObjectsReturned*

cpr_number
Model field: cpr-nummer

from_serviceplatformen
Model field: fra Serviceplatformen

id
Model field: ID

```

main_case
    Model field: hovedsag, accesses the Case model.

main_case_id
    Model field: hovedsag

name
    Model field: navn

objects = <django.db.models.manager.Manager object>

related_case
    Model field: SBSYS-sag

relation_type
    Model field: relation

static serviceplatformen_to_related_person (data)
    Convert data from Serviceplatformen to our RelatedPerson model.

class core.models.SchoolDistrict (*args, **kwargs)
    Bases: core.models.Classification

    Represents a Danish school district.

    Parameters
    • id (AutoField) – Id
    • active (BooleanField) – Aktiv
    • name (CharField) – Navn

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

cases
    Model field: skoledistrikt, accesses the M2M Case model.

id
    Model field: ID

name
    Model field: navn

objects = <django.db.models.manager.Manager object>

class core.models.Section (*args, **kwargs)
    Bases: core.models.Classification

    Law sections and the corresponding KLE codes.

    Each section is associated with the target group for which it is allowed as well as the action steps allowed.

    Parameters
    • id (AutoField) – Id
    • active (BooleanField) – Aktiv
    • paragraph (CharField) – Paragraf
    • text (TextField) – Forklarende tekst

```

- **law_text_name** (*CharField*) – Lov tekst navn
- **allowed_for_target_groups** (*ManyToManyField*) – Målgrupper
- **allowed_for_steps** (*ManyToManyField*) – Trin i indsatsstrappen

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

allowed_for_steps

Model field: trin i indsatsstrappen, accesses the M2M *Section* model.

allowed_for_target_groups

Model field: målgrupper, accesses the M2M *Section* model.

appropriations

Model field: paragraf, accesses the M2M *Appropriation* model.

id

Model field: ID

law_text_name

Model field: lov tekst navn

main_activities

Model field: hovedaktivitet for paragraffer, accesses the M2M *ActivityDetails* model.

objects = `<django.db.models.manager.Manager object>`

paragraph

Model field: paragraf

sectioninfo_set

Model field: paragraf, accesses the M2M *SectionInfo* model.

supplementary_activities

Model field: følgeudgift for paragraffer, accesses the M2M *ActivityDetails* model.

text

Model field: forklarende tekst

class `core.models.SectionInfo (*args, **kwargs)`

Bases: `django.db.models.base.Model`

For a main activity, KLE no. and SBSYS ID for the relevant sections.

Parameters

- **id** (*AutoField*) – Id
- **activity_details** (*ForeignKey* to *ActivityDetails*) – Aktivitetsdetalje
- **section** (*ForeignKey* to *Section*) – Paragraf
- **activity_category** (*ForeignKey* to *ActivityCategory*) – Aktivitetskategori
- **kle_number** (*CharField*) – Kle-nummer
- **sbsys_template_id** (*CharField*) – Sbsys skabelon-id
- **main_activity_main_account_number** (*CharField*) – Hovedkontonummer for hovedaktivitet

- **supplementary_activity_main_account_number** (*CharField*) – Hovedkontonummer for følgeudgift. Et hovedkontonummer der skal bruges, hvis en følgeudgift har denne aktivitetsdetalje som hovedydelse.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

account_aliases

Model field: paragraf-info, accesses the M2M *AccountAlias* model.

activity_category

Model field: aktivitetskategori, accesses the *ActivityCategory* model.

activity_category_id

Model field: aktivitetskategori

activity_details

Model field: aktivitetsdetalje, accesses the *ActivityDetails* model.

activity_details_id

Model field: aktivitetsdetalje

get_main_activity_main_account_number()

Get the main activity main account number.

get_supplementary_activity_main_account_number()

Get the supplementary activity main account number.

If no such number exists, take the one for the main activity.

id

Model field: ID

kle_number

Model field: KLE-nummer

main_activity_main_account_number

Model field: hovedkontonummer for hovedaktivitet

objects = <django.db.models.manager.Manager object>

sbsys_template_id

Model field: SBSYS skabelon-id

section

Model field: paragraf, accesses the *Section* model.

section_id

Model field: paragraf

supplementary_activity_main_account_number

Model field: hovedkontonummer for følgeudgift

class core.models.ServiceProvider(*args, **kwargs)

Bases: *core.models.Classification*

Class containing information for a specific service provider.

Parameters

- **id** (*AutoField*) – Id

- **active** (*BooleanField*) – Aktiv
- **cvr_number** (*CharField*) – Cvr-nummer
- **name** (*CharField*) – Navn
- **vat_factor** (*DecimalField*) – Momsfaktor

exception DoesNotExist
Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned
Bases: `django.core.exceptions.MultipleObjectsReturned`

activities
Model field: leverandør, accesses the M2M *Activity* model.

cvr_number
Model field: cvr-nummer

id
Model field: ID

name
Model field: navn

objects = `<django.db.models.manager.Manager object>`

supplied_activities
Model field: leverandører, accesses the M2M *ActivityDetails* model.

vat_factor
Model field: momsfaktor

class `core.models.TargetGroup(*args, **kwargs)`
Bases: `core.models.Classification`
Target group for a Case.

Parameters

- **id** (*AutoField*) – Id
- **active** (*BooleanField*) – Aktiv
- **name** (*CharField*) – Navn
- **required_fields_for_case** (*CharField*) – Påkrævede felter på sag

exception DoesNotExist
Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned
Bases: `django.core.exceptions.MultipleObjectsReturned`

cases
Model field: målgruppe, accesses the M2M *Case* model.

efforts
Model field: målgrupper, accesses the M2M *Effort* model.

get_required_fields_for_case()
Return `required_fields_for_case` from CSV to list.

id
Model field: ID

```

name
    Model field: navn

objects = <django.db.models.manager.Manager object>

required_fields_for_case
    Model field: påkrævede felter på sag

sections
    Model field: målgrupper, accesses the M2M Section model.

class core.models.Team(*args, **kwargs)
    Bases: django.db.models.base.Model

    Represents a team in the administration.

    Parameters
        • id (AutoField) – Id
        • name (CharField) – Navn
        • leader (ForeignKey to User) – Leder

exception DoesNotExist
    Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
    Bases: django.core.exceptions.MultipleObjectsReturned

id
    Model field: ID

leader
    Model field: leder, accesses the User model.

leader_id
    Model field: leder

name
    Model field: navn

objects = <django.db.models.manager.Manager object>

users
    Model field: team, accesses the M2M User model.

class core.models.User(*args, **kwargs)
    Bases: django.contrib.auth.models.AbstractUser

    Customized user for handling login etc.

    Parameters
        • id (AutoField) – Id
        • password (CharField) – Adgangskode
        • last_login (DateTimeField) – Sidst logget ind
        • is_superuser (BooleanField) – Superbrugerstatus. Bestemmer at denne bruger har alle rettigheder uden at tildele dem eksplicit.
        • username (CharField) – Brugernavn. Påkrævet. Højest 150 tegn. Kun bogstaver og cifre samt @/./+/_/_
        • first_name (CharField) – Fornavn

```

- **last_name** (*CharField*) – Efternavn
- **email** (*EmailField*) – E-mail-adresse
- **is_staff** (*BooleanField*) – Admin-status. Bestemmer om brugeren kan logge ind på dette administrationswebsite.
- **is_active** (*BooleanField*) – Aktiv. Bestemmer om brugeren skal behandles som aktiv. Fravælg dette frem for at slette en konto.
- **date_joined** (*DateTimeField*) – Dato for registrering
- **team** (ForeignKey to *Team*) – Team
- **profile** (*CharField*) – Brugerprofil
- **groups** (*ManyToManyField*) – Grupper. Grupperne som denne bruger hører til. En bruger får alle rettigheder givet til hver af hans/hendes grupper.
- **user_permissions** (*ManyToManyField*) – Rettigheder. Specifikke rettigheder for denne bruger.

ADMIN = 'admin'

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

EDIT = 'edit'

GRANT = 'grant'

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

READONLY = 'readonly'

WORKFLOW_ENGINE = 'workflow_engine'

approved_activities

Model field: bevilget af bruger, accesses the M2M *Activity* model.

cases

Model field: sagsbehandler, accesses the M2M *Case* model.

get_next_by_date_joined (*, *field*=<`django.db.models.fields.DateTimeField: date_joined`>, *is_next*=True, **kwargs)

Autogenerated: Finds next instance based on *date_joined*.

get_previous_by_date_joined (*, *field*=<`django.db.models.fields.DateTimeField: date_joined`>, *is_next*=False, **kwargs)

Autogenerated: Finds previous instance based on *date_joined*.

get_profile_display (*, *field*=<`django.db.models.fields.CharField: profile`>)

Autogenerated: Shows the label of the *profile*

groups

Model field: grupper, accesses the M2M *User* model.

id

Model field: ID

is_workflow_engine_or_admin ()

Return if user has workflow or admin permission.

logentry_set

Model field: bruger, accesses the M2M *LogEntry* model.

managed_teams
Model field: leder, accesses the M2M *Team* model.

classmethod max_profile (*perms*)
 Return the profile (in “perms”) with the highest permissions.
 IdPs can send more than one profile when using SAML. This function returns the “maximum” profile in the sense of the one with the most far-reaching permissions.

profile
Model field: brugerprofil

profile_choices = (('readonly', 'Kun læse'), ('edit', 'Læse og skrive'), ('grant', 'Be

rate_and_price_changes
Model field: Ændret af, accesses the M2M *RatePerDate* model.

team
Model field: team, accesses the *Team* model.

team_id
Model field: team

user_permissions
Model field: rettigheder, accesses the M2M *User* model.

class core.models.VariableRate (*args, **kwargs)
 Bases: django.db.models.base.Model
 Superclass for time-dependent rates and prices.
Parameters id (AutoField) – Id

exception DoesNotExist
 Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned
 Bases: django.core.exceptions.MultipleObjectsReturned

static create_interval (start_date, end_date)
 Create new interval for rates.

get_rate_amount (rate_date=datetime.date(2021, 4, 19))
 Look up period in RatesPerDate.

id
Model field: ID

objects = <django.db.models.manager.Manager object>

price
Model field: variablerate ptr, accesses the *Price* model.

rate
Model field: variablerate ptr, accesses the *Rate* model.

property rate_amount
 Look up period in RatesPerDate.

rates_per_date
Model field: main rate, accesses the M2M *RatePerDate* model.

set_rate_amount (amount, start_date=None, end_date=None)
 Set amount, merge with existing periods.

core.permissions module

Custom permissions used by this app.

```
class core.permissions.DeletePaymentPermission
    Bases: rest_framework.permissions.BasePermission

    Check if a payment can be deleted.

    has_object_permission (request, view, obj)
        Payments on granted activities may not be deleted.

class core.permissions.EditPaymentPermission
    Bases: rest_framework.permissions.BasePermission

    Check if this payment may be edited by the current user.

    has_object_permission (request, view, obj)
        Check that this user is allowed to apply these changes.

class core.permissions.IsUserAllowed
    Bases: rest_framework.permissions.BasePermission

    Determine user's frontend permissions i.e., in all API operations.

    has_permission (request, view)
        Allow depending on operation and user's profile.

class core.permissions.NewPaymentPermission
    Bases: core.permissions.IsUserAllowed

    Check if adding a new payment is allowed.

    has_permission (request, view)
        Allow for individual payment plan and appropriation not granted.
```

core.serializers module

Data serializers used by the REST API.

```
class core.serializers.ActivityCategorySerializer (*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the ActivityCategory model.

class core.serializers.ActivityDetailsSerializer (*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the ActivityDetails model.

class core.serializers.ActivitySerializer (*args, **kwargs)
    Bases: drf_writable_nested.serializers.WritableNestedModelSerializer

    Serializer for the Activity model.

    static setup_eager_loading (queryset)
        Set up eager loading for improved performance.

    validate (data)
        Validate this activity - check end date is after start date, etc.
```

```

class core.serializers.AppropriationSerializer(*args, **kwargs)
    Bases: core.serializers.BaseAppropriationSerializer

    Serializer for a single Appropriation model.

    get_activities(appropriation)
        Get activities on appropriation.

class core.serializers.ApprovalLevelSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the ApprovalLevel model.

class core.serializers.BaseAppropriationSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Base Serializer for the Appropriation model.

    get_num_ongoing_activities(appropriation)
        Get number of ongoing activities.

    get_num_ongoing_draft_or_expected_activities(appropriation)
        Get number of ongoing related draft or expected activities.

    static setup_eager_loading(queryset)
        Set up eager loading for improved performance.

class core.serializers.CaseSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the Case model.

    Note validation to ensure that cases are always valid according to required_fields_for_case

    get_num_ongoing_appropriations(case)
        Get number of related ongoing appropriations.

    get_num_ongoing_draft_or_expected_appropriations(case)
        Get number of related expected or draft ongoing appropriations.

    static setup_eager_loading(queryset)
        Set up eager loading for improved performance.

    validate(data)
        Check if required fields for case are present.

class core.serializers.EffortSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the Effort model.

class core.serializers.EffortStepSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the EffortStep model.

class core.serializers.HistoricalCaseSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the historic/temporal dimension of a Case.

class core.serializers.InternalPaymentRecipientSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the InternalPaymentRecipient model.

```

```
class core.serializers.ListAppropriationSerializer (*args, **kwargs)
    Bases: core.serializers.BaseAppropriationSerializer

    Serializer for the Appropriation model for a list.

class core.serializers.MunicipalitySerializer (*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the Municipality model.

class core.serializers.PaymentMethodDetailsSerializer (*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the PaymentMethodDetails model.

class core.serializers.PaymentScheduleSerializer (*args, **kwargs)
    Bases: drf_writable_nested.serializers.WritableNestedModelSerializer

    Serializer for the PaymentSchedule model.

    static setup_eager_loading (queryset)
        Set up eager loading for improved performance.

    validate (data)
        Validate this payment schedule.

        As for Payment, payment method, recipient type and various other things need to fit together.

class core.serializers.PaymentSerializer (*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the Payment model.

    Note the many options for filtering as well as the non-trivial validate function.

    save ()
        Save instance, catch and sanitize ValidationError.

    validate (data)
        Validate this payment.

class core.serializers.PriceSerializer (*args, **kwargs)
    Bases: drf_writable_nested.serializers.WritableNestedModelSerializer

    Serializer for the Price model.

    create (validated_data)
        Set rate amount on Price.

    update (instance, validated_data)
        Update rate amount on Price for dates.

class core.serializers.RatePerDateSerializer (*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the RatePerDate model.

class core.serializers.RateSerializer (*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the Rate model.

class core.serializers.RelatedPersonSerializer (*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the RelatedPerson model.
```



```

class core.serializers.SchoolDistrictSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the SchoolDistrict model.

class core.serializers.SectionInfoSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the SectionInfo model.

class core.serializers.SectionSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the Section model.

class core.serializers.ServiceProviderSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the ServiceProvider model.

class core.serializers.TargetGroupSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the TargetGroup model.

    to_representation(instance)
        Convert required_fields_for_case to list.

class core.serializers.TeamSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the Team model.

class core.serializers.UserSerializer(*args, **kwargs)
    Bases: rest_framework.serializers.ModelSerializer

    Serializer for the User model.

```

core.signals module

Signals for acting on events occurring on model objects.

```

core.signals.generate_payments_on_post_save(sender, instance, created, **kwargs)
    Generate payments for activity before saving.

core.signals.save_payment_schedule_on_save_price(sender, instance, created, **kwargs)
    Save payment schedule too when saving price.

core.signals.send_activity_created_email_on_paymentschedule_create(sender,
                                                                    instance,
                                                                    created,
                                                                    **kwargs)

    Send activity created email when PaymentSchedule is saved.

core.signals.send_activity_payment_email_on_delete(sender, instance, **kwargs)
    Send payment email when Activity is deleted.

core.signals.send_activity_payment_email_on_save(sender, instance, created, **kwargs)
    Send payment email when Activity is saved.

core.signals.set_needs_recalculation_on_save_rate(sender, instance, created,
                                                  **kwargs)
    Set "needs update" flag when changing a rate.

```

`core.signals.set_payment_id_on_paymentschedule_save` (*sender, instance, created,*
***kwargs*)

Set the payment_id as the PaymentSchedule ID on creation.

`core.signals.set_saved_account_alias_on_payment_save` (*sender, instance, created,*
***kwargs*)

Set the saved_account_alias on Payment save.

`core.signals.set_saved_account_string_on_payment_save` (*sender, instance, **kwargs*)

Set the saved_account_string on Payment save.

core.utils module

Utilities used by other parts of this app.

`core.utils.create_rrule` (*payment_type, payment_frequency, payment_day_of_month, start,*
***kwargs*)

Create a dateutil.rrule to generate dates for a payment schedule.

The rule should be based on payment_type/payment_frequency and start. Takes either “until” or “count” as kwargs.

`core.utils.due_payments_for_prism` (*date*)

Return payments which are due on date and should be sent to PRISM.

`core.utils.due_payments_for_prism_with_exclusions` (*date*)

Process payments with exclusions for PRISME.

We check the day after tomorrow for one or several payment date exclusions and include payments for those found.

`core.utils.export_prism_payments_for_date` (*date=None*)

Fetch due payments for prism and run the export functions.

`core.utils.format_prism_financial_record` (*payment, line_no, record_no,*
new_account_alias=False)

Format a single financial record for PRISM, on a single line.

This follows documentation provided by Ballerup Kommune based on KMD’s interface specification GQ311001Q for financial records (transaction type G69).

`core.utils.format_prism_payment_record` (*payment, line_no, record_no*)

Format a single payment record for PRISM, on a single line.

This follows documentation provided by Ballerup Kommune based on KMD’s interface specification GF200001Q for creditor records (transaction type G68).

`core.utils.generate_payment_date_exclusion_dates` (*years=None*)

Generate “default” dates for payment date exclusions for a number of years.

The default are danish holidays and weekends.

`core.utils.generate_payments_report` ()

Generate a payments report as CSV.

`core.utils.generate_payments_report_list_v0` (*payments, new_account_alias=False*)

Generate a payments report list of payment dicts from payments.

`core.utils.generate_payments_report_list_v1` (*payments*)

Generate payments report list v1 (v0 with new_account_alias changes).

`core.utils.generate_payments_report_list_v2` (*payments*)

Generate payments report list v2 (v1 with note added).

`core.utils.generate_records_for_prism(due_payments, new_account_alias=False)`
Generate the list of records for writing to PRISM file.

`core.utils.get_cpr_data(cpr)`
Get CPR data from Serviceplatformen.

`core.utils.get_cpr_data_mock(cpr)`
Use test data in place of the real 'get_cpr_data' for develop/test.

`core.utils.get_person_info(cpr)`
Get CPR data on a person and his/her relations.

`core.utils.parse_account_alias_mapping_data_from_csv_path(path)`
Helper-function for parsing account alias mappings from a path.

`core.utils.parse_account_alias_mapping_data_from_csv_string(string)`
Parse account alias mapping data from a .csv StringIO.

Returns a list of (main_account_number, activity_number, alias) tuples for example: [
(645511002, 015035, BOS0000109), (528211011, 015038, BOS0000112)
]

`core.utils.saml_before_login(user_data)`
Hook called after userdata is received from IdP, before login.

`core.utils.saml_create_user(user_data)`
Hook called after user is created in DB, before login.

`core.utils.send_activity_created_email(activity)`
Send an email because an activity was created.

`core.utils.send_activity_deleted_email(activity)`
Send an email because an activity has been deleted.

`core.utils.send_activity_email(subject, template, activity)`
Send an email concerning an updated activity.

`core.utils.send_activity_expired_email(activity)`
Send an email because an activity has expired.

`core.utils.send_activity_updated_email(activity)`
Send an email because an activity was updated.

`core.utils.send_appropriation(appropriation, included_activities=None)`
Generate PDF and XML files from appropriation and send them to SBSYS.

Parameters

- **appropriation** – the Appropriation from which to generate PDF and XML.
- **included_activities** – Activities which should be explicitly included.

`core.utils.validate_cvr(cvr)`
Validate a cvr number string.

Note: this is merely a 8-digit number input validation and not a 'real' CVR validation

`core.utils.write_prism_file_v0(filename, date, payments, tomorrow, new_account_alias=False)`
Write the actual PRISM file.

core.views module

Views and viewsets exposed by the REST interface.

```
class core.views.ActivityCategoryViewSet (**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose activity categories in REST API.

    queryset = QuerySet

    serializer_class
        alias of core.serializers.ActivityCategorySerializer

class core.views.ActivityDetailsViewSet (**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose activity details in REST API.

    basename = None

    description = None

    detail = None

    filterset_fields = '__all__'

    name = None

    queryset = QuerySet

    serializer_class
        alias of core.serializers.ActivityDetailsSerializer

    suffix = None

class core.views.ActivityViewSet (**kwargs)
    Bases: core.mixins.AuditModelViewSetMixin, core.views.AuditViewSet
    Expose activities in REST API.

    basename = None

    description = None

    destroy(request, *args, **kwargs)
        Handle deletion according to their business logic.

        Drafts are deleted, expectations are logically deleted, granted activities cannot be deleted.

    detail = None

    filterset_fields = '__all__'

    get_queryset()
        Avoid Django's default lazy loading to improve performance.

    name = None

    serializer_class
        alias of core.serializers.ActivitySerializer

    suffix = None

class core.views.AppropriationViewSet (**kwargs)
    Bases: core.mixins.AuditModelViewSetMixin, core.views.AuditViewSet
    Expose appropriations in REST API.
```

Note the custom action `grant` for approving an appropriation and all its activities.

```

basename = None
description = None
detail = None
filterset_class
    alias of core.filters.AppropriationFilter
get_queryset ()
    Avoid Django's default lazy loading to improve performance.
get_serializer_class ()
    Use a different Serializer depending on the action.
grant (request, pk=None)
    Grant the specified activities on this appropriation.
name = None
serializer_action_classes = {'list': <class 'core.serializers.ListAppropriationSerial
suffix = None

class core.views.ApprovalLevelViewSet (**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose approval levels in REST API.

    basename = None
    description = None
    detail = None
    filterset_fields = '__all__'
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.ApprovalLevelSerializer
    suffix = None

class core.views.AuditViewSet (**kwargs)
    Bases: core.mixins.AuditMixin, rest_framework.viewsets.ModelViewSet
    Superclass for use by all model classes with audit fields.

    All classes that can be written through the REST interface have audit fields.

    authentication_classes = (<class 'core.authentication.CsrfExemptSessionAuthentication'>,)
    permission_classes = (<class 'core.permissions.IsUserAllowed'>,)

class core.views.CaseViewSet (**kwargs)
    Bases: core.mixins.AuditModelViewSetMixin, core.views.AuditViewSet
    Viewset exposing Case in the REST API.

    Note the custom actions history and change_case_worker.

    basename = None

```

change_case_worker (*request*)
Change the case_worker of several Cases.

Parameters

- **case_pks** – A list of case pks.
- **case_worker_pk** – the case worker pk to change to.

description = None

detail = None

filterset_class
alias of *core.filters.CaseFilter*

get_queryset ()
Avoid Django's default lazy loading to improve performance.

history (*request, pk=None*)
Fetch history of HistoricalCases which we use as assessments.

name = None

perform_create (*serializer*)
Create new case - customized to set user.

queryset = **CaseQuerySet**

serializer_class
alias of *core.serializers.CaseSerializer*

suffix = None

class **core.views.EffortStepViewSet** (***kwargs*)
Bases: *core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet*
Expose effort steps in REST API.

basename = None

description = None

detail = None

filterset_fields = '__all__'

name = None

queryset = **QuerySet**

serializer_class
alias of *core.serializers.EffortStepSerializer*

suffix = None

class **core.views.EffortViewSet** (***kwargs*)
Bases: *core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet*
Expose efforts in REST API.

basename = None

description = None

detail = None

filterset_fields = '__all__'

```

    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.EffortSerializer
    suffix = None
class core.views.InternalPaymentRecipientViewSet (**kwargs)
    Bases: core.views.ReadOnlyViewSet
    Expose internal payment recipients in REST API.
    basename = None
    description = None
    detail = None
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.InternalPaymentRecipientSerializer
    suffix = None
class core.views.IsEditingPastPaymentsAllowed (**kwargs)
    Bases: rest_framework.views.APIView
    Expose the ALLOW_EDIT_OF_PAST_PAYMENTS setting in the API.
    authentication_classes = (<class 'core.authentication.CsrfExemptSessionAuthentication'>)
    get (request, format=None)
        Return the Django setting allowing changes to the past.
class core.views.MunicipalityViewSet (**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose municipalities in REST API.
    basename = None
    description = None
    detail = None
    filterset_fields = '__all__'
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.MunicipalitySerializer
    suffix = None
class core.views.PaymentMethodDetailsViewSet (**kwargs)
    Bases: core.views.AuditViewSet
    Expose payment method details in REST API.
    basename = None
    description = None

```

```
    detail = None
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.PaymentMethodDetailsSerializer
    suffix = None

class core.views.PaymentScheduleViewSet (**kwargs)
    Bases: core.views.AuditViewSet
    Expose payment schedules in REST API.

    basename = None
    description = None
    detail = None
    get_queryset ()
        Avoid Django's default lazy loading to improve performance.
    name = None
    serializer_class
        alias of core.serializers.PaymentScheduleSerializer
    suffix = None

class core.views.PaymentViewSet (**kwargs)
    Bases: core.views.AuditViewSet
    Expose payments in REST API.
    Note, this viewset supports pagination.

    basename = None
    description = None
    detail = None
    filterset_class
        alias of core.filters.PaymentFilter
    filterset_fields = '__all__'
    name = None
    pagination_class
        alias of rest_framework.pagination.PageNumberPagination
    permission_classes = (<class 'core.permissions.EditPaymentPermission'>, <class 'core.p
    queryset = PaymentQuerySet
    serializer_class
        alias of core.serializers.PaymentSerializer
    suffix = None

class core.views.PriceViewSet (**kwargs)
    Bases: core.views.AuditViewSet
    Expose Price objects in REST API.
```



```

    basename = None
    description = None
    detail = None
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.PriceSerializer
    suffix = None
class core.views.RateViewSet (**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose rates in REST API.
    basename = None
    description = None
    detail = None
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.RateSerializer
    suffix = None
class core.views.ReadOnlyViewSet (**kwargs)
    Bases: rest_framework.viewsets.ReadOnlyModelViewSet
    Superclass for use model classes that are read only through REST.
    permission_classes = (<class 'core.permissions.IsUserAllowed'>,)
class core.views.RelatedPersonViewSet (**kwargs)
    Bases: core.mixins.AuditModelViewSetMixin, core.views.AuditViewSet
    Expose related persons - typically family relations - in REST API.
    basename = None
    description = None
    detail = None
    fetch_from_serviceplatformen (request)
        Fetch relations for a person using the CPR from Serviceplatformen.
        Returns the data as serialized RelatedPersons data.
        GET params: cpr
    filterset_fields = '__all__'
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.RelatedPersonSerializer

```

```
    suffix = None

class core.views.SchoolDistrictViewSet(**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose school districts in REST API.

    basename = None
    description = None
    detail = None
    filterset_fields = '__all__'
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.SchoolDistrictSerializer
    suffix = None

class core.views.SectionInfoViewSet(**kwargs)
    Bases: core.views.ReadOnlyViewSet
    Expose section infos in REST API.

    basename = None
    description = None
    detail = None
    filterset_fields = '__all__'
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.SectionInfoSerializer
    suffix = None

class core.views.SectionViewSet(**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose law sections in REST API.

    basename = None
    description = None
    detail = None
    filterset_class
        alias of core.filters.AllowedForStepsFilter
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.SectionSerializer
    suffix = None
```

```

class core.views.ServiceProviderViewSet(**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose service providers in REST API.

    basename = None
    description = None
    detail = None
    filterset_fields = '__all__'
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.ServiceProviderSerializer
    suffix = None

class core.views.TargetGroupViewSet(**kwargs)
    Bases: core.mixins.ClassificationViewSetMixin, core.views.ReadOnlyViewSet
    Expose target groups in REST API.

    basename = None
    description = None
    detail = None
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.TargetGroupSerializer
    suffix = None

class core.views.TeamViewSet(**kwargs)
    Bases: core.views.ReadOnlyViewSet
    Expose teams in REST API.

    basename = None
    description = None
    detail = None
    name = None
    queryset = QuerySet
    serializer_class
        alias of core.serializers.TeamSerializer
    suffix = None

class core.views.UserViewSet(**kwargs)
    Bases: core.views.ReadOnlyViewSet
    Expose users in REST API.

    basename = None

```

```
description = None
detail = None
name = None
queryset = QuerySet
serializer_class
    alias of core.serializers.UserSerializer
suffix = None
```

Module contents

‘core’ package containing all functionality.

The bevillingsplatform package - settings and configuration

Submodules

bevillingsplatform.initialize module

Functions for populating the database with initial data.

```
bevillingsplatform.initialize.initialize()
```

Initialize all the basic data we want at start.

Should be able to be run multiple times over without generating duplicates.

```
bevillingsplatform.initialize.initialize_account_alias_mappings()
```

Initialize the account alias mappings.

```
bevillingsplatform.initialize.initialize_account_aliases()
```

Initialize the account aliases.

```
bevillingsplatform.initialize.initialize_activity_categories()
```

Initialize all the relevant activity categories.

Data should be the output of “manage.py dumpdata core.activitycategory

```
bevillingsplatform.initialize.initialize_activity_details()
```

Initialize all the relevant activity details.

Data should be the output of “manage.py dumpdata core.activitydetails”.

```
bevillingsplatform.initialize.initialize_approval_levels()
```

Initialize all the initial approval levels.

Data should be the output of “manage.py dumpdata core.approvallevels”.

```
bevillingsplatform.initialize.initialize_municipalities()
```

Initialize all the danish municipalities.

```
bevillingsplatform.initialize.initialize_payment_method_details()
```

Initialize all the relevant payment method details.

```
bevillingsplatform.initialize.initialize_rates()
```

Initialize the variable rates, rates and rates per date for Ballerup.

```
bevillingsplatform.initialize.initialize_school_districts()
    Initialize all the school districts for Ballerup.

bevillingsplatform.initialize.initialize_section_infos()
    Initialize all the relevant section infos.

    Data should be the output of “manage.py dumpdata core.sectioninfos”.

bevillingsplatform.initialize.initialize_sections()
    Initialize all the relevant law sections.

    Data should be the output of “manage.py dumpdata core.section”.

bevillingsplatform.initialize.initialize_service_providers()
    Initialize all the relevant service providers.

    Data should be the output of “manage.py dumpdata core.serviceprovider”.

bevillingsplatform.initialize.initialize_target_groups()
    Initialize the initial target groups.

    Data should be output of “manage.py dumpdata core.targetgroups”.

bevillingsplatform.initialize.initialize_teams()
    Initialize all the school districts for Ballerup.

bevillingsplatform.initialize.initialize_users()
    Prime the system with some users to get started.

    Data should be the output of “manage.py dumpdata core.User”.
```

bevillingsplatform.settings module

Django settings for bevillingsplatform project.

Generated by “django-admin startproject” using Django 2.2.1.

For more information on this file, see <https://docs.djangoproject.com/en/2.2/topics/settings/>

For the full list of settings and their values, see <https://docs.djangoproject.com/en/2.2/ref/settings/>

bevillingsplatform.urls module

bevillingsplatform URL Configuration.

The *urlpatterns* list routes URLs to views. For more information please see: <https://docs.djangoproject.com/en/2.2/topics/http/urls/>

Examples

Function views

1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')

Including another URLconf

1. Import the `include()` function: `from django.urls import include, path`
2. Add a URL to `urlpatterns`: `path('blog/', include('blog.urls'))`

bevillingsplatform.wsgi module

WSGI config for bevillingsplatform project.

It exposes the WSGI callable as a module-level variable named `application`.

For more information on this file, see <https://docs.djangoproject.com/en/2.2/howto/deployment/wsgi/>

Module contents

'bevillingsplatform' package created by Django.

manage module

Django's command-line utility for administrative tasks.

```
manage.main()  
    Run main function.
```

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

API documentation

GET `/api/cases/`

ViewSet exposing Case in the REST API.

Note the custom actions `history` and `change_case_worker`.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **sbsys_id** (*string*) – sbsys_id
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **case_worker** (*string*) – Sagsbehandler

- **district** (*string*) – district
- **paying_municipality** (*string*) – paying_municipality
- **acting_municipality** (*string*) – acting_municipality
- **residence_municipality** (*string*) – residence_municipality
- **target_group** (*string*) – target_group
- **effort_step** (*string*) – effort_step
- **scaling_step** (*string*) – scaling_step
- **assessment_comment** (*string*) – assessment_comment
- **efforts** (*string*) – efforts
- **note** (*string*) – note
- **expired** (*string*) – Udgået

Example request:

```
GET /api/cases/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "expired": "string",
    "num_ongoing_appropriations": "string",
    "num_ongoing_draft_or_expected_appropriations": "string",
    "team": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "sbsys_id": "string",
    "cpr_number": "string",
    "name": "string",
    "scaling_step": 1,
    "assessment_comment": "string",
    "note": "string",
    "case_worker": 1,
    "district": 1,
    "paying_municipality": 1,
    "acting_municipality": 1,
    "residence_municipality": 1,
    "target_group": 1,
    "effort_step": 1,
    "efforts": [
      1
    ]
  }
]
```

(continues on next page)

(continued from previous page)

```
}  
]
```

POST /api/cases/

Viewset exposing Case in the REST API.

Note the custom actions history and change_case_worker.

Example request:

```
POST /api/cases/ HTTP/1.1  
Host: example.com  
Content-Type: application/json  
  
{  
  "user_created": "string",  
  "user_modified": "string",  
  "sbsys_id": "string",  
  "cpr_number": "string",  
  "name": "string",  
  "scaling_step": 1,  
  "assessment_comment": "string",  
  "note": "string",  
  "case_worker": 1,  
  "district": 1,  
  "paying_municipality": 1,  
  "acting_municipality": 1,  
  "residence_municipality": 1,  
  "target_group": 1,  
  "effort_step": 1,  
  "efforts": [  
    1  
  ]  
}
```

Status Codes

- 201 Created – **Example response:**

```
HTTP/1.1 201 Created  
Content-Type: application/json  
  
{  
  "id": 1,  
  "expired": "string",  
  "num_ongoing_appropriations": "string",  
  "num_ongoing_draft_or_expected_appropriations": "string",  
  "team": "string",  
  "created": "2021-04-19T09:16:55.367826",  
  "modified": "2021-04-19T09:16:55.367826",  
  "user_created": "string",  
  "user_modified": "string",  
  "sbsys_id": "string",  
  "cpr_number": "string",  
  "name": "string",  
  "scaling_step": 1,  
}
```

(continues on next page)

(continued from previous page)

```

    "assessment_comment": "string",
    "note": "string",
    "case_worker": 1,
    "district": 1,
    "paying_municipality": 1,
    "acting_municipality": 1,
    "residence_municipality": 1,
    "target_group": 1,
    "effort_step": 1,
    "efforts": [
        1
    ]
}

```

GET /api/cases/{id}/

Viewset exposing Case in the REST API.

Note the custom actions history and change_case_worker.

Parameters

- **id** (*string*) – A unique integer value identifying this sag.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **sbsys_id** (*string*) – sbsys_id
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **case_worker** (*string*) – Sagsbehandler
- **district** (*string*) – district
- **paying_municipality** (*string*) – paying_municipality
- **acting_municipality** (*string*) – acting_municipality
- **residence_municipality** (*string*) – residence_municipality
- **target_group** (*string*) – target_group
- **effort_step** (*string*) – effort_step
- **scaling_step** (*string*) – scaling_step
- **assessment_comment** (*string*) – assessment_comment
- **efforts** (*string*) – efforts
- **note** (*string*) – note
- **expired** (*string*) – Udgået

Example request:

```
GET /api/cases/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "expired": "string",
  "num_ongoing_appropriations": "string",
  "num_ongoing_draft_or_expected_appropriations": "string",
  "team": "string",
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "cpr_number": "string",
  "name": "string",
  "scaling_step": 1,
  "assessment_comment": "string",
  "note": "string",
  "case_worker": 1,
  "district": 1,
  "paying_municipality": 1,
  "acting_municipality": 1,
  "residence_municipality": 1,
  "target_group": 1,
  "effort_step": 1,
  "efforts": [
    1
  ]
}
```

PUT /api/cases/{id}/

Viewset exposing Case in the REST API.

Note the custom actions `history` and `change_case_worker`.

Parameters

- **id** (*string*) – A unique integer value identifying this sag.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **sbsys_id** (*string*) – sbsys_id
- **cpr_number** (*string*) – cpr_number

- **name** (*string*) – name
- **case_worker** (*string*) – Sagsbehandler
- **district** (*string*) – district
- **paying_municipality** (*string*) – paying_municipality
- **acting_municipality** (*string*) – acting_municipality
- **residence_municipality** (*string*) – residence_municipality
- **target_group** (*string*) – target_group
- **effort_step** (*string*) – effort_step
- **scaling_step** (*string*) – scaling_step
- **assessment_comment** (*string*) – assessment_comment
- **efforts** (*string*) – efforts
- **note** (*string*) – note
- **expired** (*string*) – Udgået

Example request:

```
PUT /api/cases/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "cpr_number": "string",
  "name": "string",
  "scaling_step": 1,
  "assessment_comment": "string",
  "note": "string",
  "case_worker": 1,
  "district": 1,
  "paying_municipality": 1,
  "acting_municipality": 1,
  "residence_municipality": 1,
  "target_group": 1,
  "effort_step": 1,
  "efforts": [
    1
  ]
}
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
```

(continues on next page)

(continued from previous page)

```

    "expired": "string",
    "num_ongoing_appropriations": "string",
    "num_ongoing_draft_or_expected_appropriations": "string",
    "team": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "sbsys_id": "string",
    "cpr_number": "string",
    "name": "string",
    "scaling_step": 1,
    "assessment_comment": "string",
    "note": "string",
    "case_worker": 1,
    "district": 1,
    "paying_municipality": 1,
    "acting_municipality": 1,
    "residence_municipality": 1,
    "target_group": 1,
    "effort_step": 1,
    "efforts": [
      1
    ]
  }

```

PATCH /api/cases/{id}/

Viewset exposing Case in the REST API.

Note the custom actions history and change_case_worker.

Parameters

- **id** (*string*) – A unique integer value identifying this sag.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **sbsys_id** (*string*) – sbsys_id
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **case_worker** (*string*) – Sagsbehandler
- **district** (*string*) – district
- **paying_municipality** (*string*) – paying_municipality
- **acting_municipality** (*string*) – acting_municipality
- **residence_municipality** (*string*) – residence_municipality
- **target_group** (*string*) – target_group

- **effort_step**(*string*) – effort_step
- **scaling_step**(*string*) – scaling_step
- **assessment_comment**(*string*) – assessment_comment
- **efforts**(*string*) – efforts
- **note**(*string*) – note
- **expired**(*string*) – Udgået

Example request:

```

PATCH /api/cases/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "cpr_number": "string",
  "name": "string",
  "scaling_step": 1,
  "assessment_comment": "string",
  "note": "string",
  "case_worker": 1,
  "district": 1,
  "paying_municipality": 1,
  "acting_municipality": 1,
  "residence_municipality": 1,
  "target_group": 1,
  "effort_step": 1,
  "efforts": [
    1
  ]
}

```

Status Codes

- 200 OK – Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "expired": "string",
  "num_ongoing_appropriations": "string",
  "num_ongoing_draft_or_expected_appropriations": "string",
  "team": "string",
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "cpr_number": "string",
  "name": "string",
  "scaling_step": 1,

```

(continues on next page)

(continued from previous page)

```
"assessment_comment": "string",
"note": "string",
"case_worker": 1,
"district": 1,
"paying_municipality": 1,
"acting_municipality": 1,
"residence_municipality": 1,
"target_group": 1,
"effort_step": 1,
"efforts": [
  1
]
}
```

DELETE /api/cases/{id}/

Viewset exposing Case in the REST API.

Note the custom actions history and change_case_worker.

Parameters

- **id** (*string*) – A unique integer value identifying this sag.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **sbsys_id** (*string*) – sbsys_id
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **case_worker** (*string*) – Sagsbehandler
- **district** (*string*) – district
- **paying_municipality** (*string*) – paying_municipality
- **acting_municipality** (*string*) – acting_municipality
- **residence_municipality** (*string*) – residence_municipality
- **target_group** (*string*) – target_group
- **effort_step** (*string*) – effort_step
- **scaling_step** (*string*) – scaling_step
- **assessment_comment** (*string*) – assessment_comment
- **efforts** (*string*) – efforts
- **note** (*string*) – note
- **expired** (*string*) – Udgået

Status Codes

- [204 No Content](#) –

GET /api/cases/{id}/history/

Fetch history of HistoricalCases which we use as assessments.

Parameters

- **id** (*string*) – A unique integer value identifying this sag.

Example request:

```
GET /api/cases/{id}/history/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK** – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "expired": "string",
  "num_ongoing_appropriations": "string",
  "num_ongoing_draft_or_expected_appropriations": "string",
  "team": "string",
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "cpr_number": "string",
  "name": "string",
  "scaling_step": 1,
  "assessment_comment": "string",
  "note": "string",
  "case_worker": 1,
  "district": 1,
  "paying_municipality": 1,
  "acting_municipality": 1,
  "residence_municipality": 1,
  "target_group": 1,
  "effort_step": 1,
  "efforts": [
    1
  ]
}
```

GET /api/appropriations/

Expose appropriations in REST API.

Note the custom action `grant` for approving an appropriation and all its activities.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified

- **sbsys_id**(*string*) – sbsys_id
- **section**(*string*) – section
- **case**(*string*) – Sag
- **note**(*string*) – note
- **main_activity__details__id**(*string*) – Aktivitetsdetalje for hovedaktivitet

Example request:

```
GET /api/appropriations/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "status": "string",
    "granted_from_date": "string",
    "granted_to_date": "string",
    "case__cpr_number": "string",
    "case__name": "string",
    "case__sbsys_id": "string",
    "num_ongoing_draft_or_expected_activities": "string",
    "num_ongoing_activities": "string",
    "main_activity__details__id": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "sbsys_id": "string",
    "note": "string",
    "section": 1,
    "case": 1
  }
]
```

POST /api/appropriations/

Expose appropriations in REST API.

Note the custom action `grant` for approving an appropriation and all its activities.

Example request:

```
POST /api/appropriations/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
```

(continues on next page)

(continued from previous page)

```

"note": "string",
"section": 1,
"case": 1
}

```

Status Codes

- 201 Created – Example response:

```

HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": 1,
  "status": "string",
  "granted_from_date": "string",
  "granted_to_date": "string",
  "case_cpr_number": "string",
  "case_name": "string",
  "case_sbsys_id": "string",
  "num_ongoing_draft_or_expected_activities": "string",
  "num_ongoing_activities": "string",
  "main_activity": {
    "id": 1,
    "monthly_payment_plan": "string",
    "total_cost": "string",
    "total_cost_this_year": "string",
    "total_cost_full_year": "string",
    "total_granted_this_year": "string",
    "total_expected_this_year": "string",
    "payment_plan": {
      "id": 1,
      "payments": [
        {
          "id": 1,
          "account_string": "string",
          "account_alias": "string",
          "payment_schedule_payment_id": "string",
          "case_cpr_number": "string",
          "case_name": "string",
          "activity_id": "string",
          "activity_status": "string",
          "activity_details_id": "string",
          "payment_schedule_fictive": "string",
          "is_payable_manually": "string",
          "date": "2021-04-19",
          "recipient_type": "INTERNAL",
          "recipient_id": "string",
          "recipient_name": "string",
          "payment_method": "CASH",
          "amount": "string",
          "paid_amount": "string",
          "paid": true,
          "paid_date": "2021-04-19",
          "note": "string",
          "payment_schedule": 1
        }
      ]
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

    }
  ],
  "price_per_unit": {
    "id": 1,
    "rates_per_date": [
      {
        "rate": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826"
      },
      {
        "changed_by": 1
      }
    ],
    "current_amount": "string",
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
  },
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "payment_frequency": "DAILY",
  "payment_date": "2021-04-19",
  "payment_day_of_month": 1,
  "payment_type": "ONE_TIME_PAYMENT",
  "payment_units": "string",
  "payment_amount": "string",
  "fictive": true,
  "payment_id": 1,
  "payment_cost_type": "FIXED",
  "payment_method_details": 1,
  "payment_rate": "string"
},
"details__name": "string",
"created": "2021-04-19T09:16:55.367826",
"modified": "2021-04-19T09:16:55.367826",
"user_created": "string",
"user_modified": "string",
"status": "DRAFT",
"approval_note": "string",
"appropriation_date": "2021-04-19",
"start_date": "2021-04-19",
"end_date": "2021-04-19",
"activity_type": "MAIN_ACTIVITY",
"note": "string",
"details": 1,
"approval_level": 1,
"approval_user": 1,
"modifies": 1,
"appropriation": 1,
"service_provider": 1
},
"activities": "string",
"total_granted_this_year": "string",
"total_granted_full_year": "string",

```

(continues on next page)

(continued from previous page)

```

    "total_expected_this_year": "string",
    "total_expected_full_year": "string",
    "total_cost_expected": "string",
    "total_cost_granted": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "sbsys_id": "string",
    "note": "string",
    "section": 1,
    "case": 1
  }

```

GET /api/appropriations/{id}/

Expose appropriations in REST API.

Note the custom action `grant` for approving an appropriation and all its activities.**Parameters**

- **id** (*string*) –

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **sbsys_id** (*string*) – sbsys_id
- **section** (*string*) – section
- **case** (*string*) – Sag
- **note** (*string*) – note
- **main_activity__details__id** (*string*) – Aktivitetsdetalje for hovedaktivitet

Example request:

```

GET /api/appropriations/{id}/ HTTP/1.1
Host: example.com

```

Status Codes

- **200 OK – Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "status": "string",
  "granted_from_date": "string",
  "granted_to_date": "string",
  "case_cpr_number": "string",

```

(continues on next page)

(continued from previous page)

```

"case_name": "string",
"case_sbsys_id": "string",
"num_ongoing_draft_or_expected_activities": "string",
"num_ongoing_activities": "string",
"main_activity": {
  "id": 1,
  "monthly_payment_plan": "string",
  "total_cost": "string",
  "total_cost_this_year": "string",
  "total_cost_full_year": "string",
  "total_granted_this_year": "string",
  "total_expected_this_year": "string",
  "payment_plan": {
    "id": 1,
    "payments": [
      {
        "id": 1,
        "account_string": "string",
        "account_alias": "string",
        "payment_schedule_payment_id": "string",
        "case_cpr_number": "string",
        "case_name": "string",
        "activity_id": "string",
        "activity_status": "string",
        "activity_details_id": "string",
        "payment_schedule_fictive": "string",
        "is_payable_manually": "string",
        "date": "2021-04-19",
        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",
        "payment_method": "CASH",
        "amount": "string",
        "paid_amount": "string",
        "paid": true,
        "paid_date": "2021-04-19",
        "note": "string",
        "payment_schedule": 1
      }
    ],
    "price_per_unit": {
      "id": 1,
      "rates_per_date": [
        {
          "rate": "string",
          "start_date": "2021-04-19",
          "end_date": "2021-04-19",
          "changed_date": "2021-04-19T09:16:55.367826"
        },
        {
          "changed_by": 1
        }
      ],
      "current_amount": "string",
      "amount": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19"
    }
  },
},

```

(continues on next page)

(continued from previous page)

```

        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",
        "payment_method": "CASH",
        "payment_frequency": "DAILY",
        "payment_date": "2021-04-19",
        "payment_day_of_month": 1,
        "payment_type": "ONE_TIME_PAYMENT",
        "payment_units": "string",
        "payment_amount": "string",
        "fictive": true,
        "payment_id": 1,
        "payment_cost_type": "FIXED",
        "payment_method_details": 1,
        "payment_rate": "string"
    },
    "details__name": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "status": "DRAFT",
    "approval_note": "string",
    "appropriation_date": "2021-04-19",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19",
    "activity_type": "MAIN_ACTIVITY",
    "note": "string",
    "details": 1,
    "approval_level": 1,
    "approval_user": 1,
    "modifies": 1,
    "appropriation": 1,
    "service_provider": 1
},
"activities": "string",
"total_granted_this_year": "string",
"total_granted_full_year": "string",
"total_expected_this_year": "string",
"total_expected_full_year": "string",
"total_cost_expected": "string",
"total_cost_granted": "string",
"created": "2021-04-19T09:16:55.367826",
"modified": "2021-04-19T09:16:55.367826",
"user_created": "string",
"user_modified": "string",
"sbsys_id": "string",
"note": "string",
"section": 1,
"case": 1
}

```

PUT /api/appropriations/{id}/

Expose appropriations in REST API.

Note the custom action `grant` for approving an appropriation and all its activities.

Parameters

- **id**(*string*) –

Query Parameters

- **created**(*string*) – created
- **modified**(*string*) – modified
- **user_created**(*string*) – user_created
- **user_modified**(*string*) – user_modified
- **sbsys_id**(*string*) – sbsys_id
- **section**(*string*) – section
- **case**(*string*) – Sag
- **note**(*string*) – note
- **main_activity__details__id**(*string*) – Aktivitetsdetalje for hovedaktivitet

Example request:

```
PUT /api/appropriations/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "note": "string",
  "section": 1,
  "case": 1
}
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "status": "string",
  "granted_from_date": "string",
  "granted_to_date": "string",
  "case_cpr_number": "string",
  "case_name": "string",
  "case_sbsys_id": "string",
  "num_ongoing_draft_or_expected_activities": "string",
  "num_ongoing_activities": "string",
  "main_activity": {
    "id": 1,
    "monthly_payment_plan": "string",
    "total_cost": "string",
    "total_cost_this_year": "string",
    "total_cost_full_year": "string",

```

(continues on next page)

(continued from previous page)

```

"total_granted_this_year": "string",
"total_expected_this_year": "string",
"payment_plan": {
  "id": 1,
  "payments": [
    {
      "id": 1,
      "account_string": "string",
      "account_alias": "string",
      "payment_schedule_payment_id": "string",
      "case_cpr_number": "string",
      "case_name": "string",
      "activity_id": "string",
      "activity_status": "string",
      "activity_details_id": "string",
      "payment_schedule_fictive": "string",
      "is_payable_manually": "string",
      "date": "2021-04-19",
      "recipient_type": "INTERNAL",
      "recipient_id": "string",
      "recipient_name": "string",
      "payment_method": "CASH",
      "amount": "string",
      "paid_amount": "string",
      "paid": true,
      "paid_date": "2021-04-19",
      "note": "string",
      "payment_schedule": 1
    }
  ],
  "price_per_unit": {
    "id": 1,
    "rates_per_date": [
      {
        "rate": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826"
      },
      {
        "changed_by": 1
      }
    ],
    "current_amount": "string",
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
  },
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "payment_frequency": "DAILY",
  "payment_date": "2021-04-19",
  "payment_day_of_month": 1,
  "payment_type": "ONE_TIME_PAYMENT",
  "payment_units": "string",
  "payment_amount": "string",

```

(continues on next page)

(continued from previous page)

```

        "fictive": true,
        "payment_id": 1,
        "payment_cost_type": "FIXED",
        "payment_method_details": 1,
        "payment_rate": "string"
    },
    "details__name": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "status": "DRAFT",
    "approval_note": "string",
    "appropriation_date": "2021-04-19",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19",
    "activity_type": "MAIN_ACTIVITY",
    "note": "string",
    "details": 1,
    "approval_level": 1,
    "approval_user": 1,
    "modifies": 1,
    "appropriation": 1,
    "service_provider": 1
},
"activities": "string",
"total_granted_this_year": "string",
"total_granted_full_year": "string",
"total_expected_this_year": "string",
"total_expected_full_year": "string",
"total_cost_expected": "string",
"total_cost_granted": "string",
"created": "2021-04-19T09:16:55.367826",
"modified": "2021-04-19T09:16:55.367826",
"user_created": "string",
"user_modified": "string",
"sbsys_id": "string",
"note": "string",
"section": 1,
"case": 1
}

```

PATCH /api/appropriations/{id}/

Expose appropriations in REST API.

Note the custom action `grant` for approving an appropriation and all its activities.

Parameters

- **id**(*string*) –

Query Parameters

- **created**(*string*) – created
- **modified**(*string*) – modified
- **user_created**(*string*) – user_created
- **user_modified**(*string*) – user_modified

- **sbsys_id**(*string*) – sbsys_id
- **section**(*string*) – section
- **case**(*string*) – Sag
- **note**(*string*) – note
- **main_activity__details__id**(*string*) – Aktivitetsdetalje for hovedaktivitet

Example request:

```
PATCH /api/appropriations/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json
```

```
{
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "note": "string",
  "section": 1,
  "case": 1
}
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "status": "string",
  "granted_from_date": "string",
  "granted_to_date": "string",
  "case_cpr_number": "string",
  "case_name": "string",
  "case_sbsys_id": "string",
  "num_ongoing_draft_or_expected_activities": "string",
  "num_ongoing_activities": "string",
  "main_activity": {
    "id": 1,
    "monthly_payment_plan": "string",
    "total_cost": "string",
    "total_cost_this_year": "string",
    "total_cost_full_year": "string",
    "total_granted_this_year": "string",
    "total_expected_this_year": "string",
    "payment_plan": {
      "id": 1,
      "payments": [
        {
          "id": 1,
          "account_string": "string",
          "account_alias": "string",
          "payment_schedule_payment_id": "string",
          "case_cpr_number": "string",
          "case_name": "string",

```

(continues on next page)

(continued from previous page)

```

        "activity_id": "string",
        "activity_status": "string",
        "activity_details_id": "string",
        "payment_schedule_fictive": "string",
        "is_payable_manually": "string",
        "date": "2021-04-19",
        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",
        "payment_method": "CASH",
        "amount": "string",
        "paid_amount": "string",
        "paid": true,
        "paid_date": "2021-04-19",
        "note": "string",
        "payment_schedule": 1
    },
    ],
    "price_per_unit": {
        "id": 1,
        "rates_per_date": [
            {
                "rate": "string",
                "start_date": "2021-04-19",
                "end_date": "2021-04-19",
                "changed_date": "2021-04-19T09:16:55.367826"
            },
            {
                "changed_by": 1
            }
        ],
        "current_amount": "string",
        "amount": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19"
    },
    },
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_id": 1,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
},
"details_name": "string",
"created": "2021-04-19T09:16:55.367826",
"modified": "2021-04-19T09:16:55.367826",
"user_created": "string",
"user_modified": "string",
"status": "DRAFT",

```

(continues on next page)

(continued from previous page)

```

    "approval_note": "string",
    "appropriation_date": "2021-04-19",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19",
    "activity_type": "MAIN_ACTIVITY",
    "note": "string",
    "details": 1,
    "approval_level": 1,
    "approval_user": 1,
    "modifies": 1,
    "appropriation": 1,
    "service_provider": 1
  },
  "activities": "string",
  "total_granted_this_year": "string",
  "total_granted_full_year": "string",
  "total_expected_this_year": "string",
  "total_expected_full_year": "string",
  "total_cost_expected": "string",
  "total_cost_granted": "string",
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "note": "string",
  "section": 1,
  "case": 1
}

```

DELETE /api/appropriations/{id}/

Expose appropriations in REST API.

Note the custom action `grant` for approving an appropriation and all its activities.**Parameters**

- **id**(*string*) –

Query Parameters

- **created**(*string*) – created
- **modified**(*string*) – modified
- **user_created**(*string*) – user_created
- **user_modified**(*string*) – user_modified
- **sbsys_id**(*string*) – sbsys_id
- **section**(*string*) – section
- **case**(*string*) – Sag
- **note**(*string*) – note
- **main_activity__details__id**(*string*) – Aktivitetsdetalje for hovedaktivitet

Status Codes

- 204 No Content –

GET /api/activities/
Expose activities in REST API.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **details** (*string*) – details
- **status** (*string*) – status
- **approval_level** (*string*) – approval_level
- **approval_note** (*string*) – approval_note
- **approval_user** (*string*) – approval_user
- **appropriation_date** (*string*) – appropriation_date
- **start_date** (*string*) – start_date
- **end_date** (*string*) – end_date
- **activity_type** (*string*) – activity_type
- **modifies** (*string*) – modifies
- **appropriation** (*string*) – appropriation
- **service_provider** (*string*) – service_provider
- **note** (*string*) – note

Example request:

```
GET /api/activities/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "monthly_payment_plan": "string",
    "total_cost": "string",
    "total_cost_this_year": "string",
    "total_cost_full_year": "string",
    "total_granted_this_year": "string",
    "total_expected_this_year": "string",
    "payment_plan": {
      "id": 1,
      "payments": [
        {
          "id": 1,
```

(continues on next page)

(continued from previous page)

```

        "account_string": "string",
        "account_alias": "string",
        "payment_schedule_payment_id": "string",
        "case_cpr_number": "string",
        "case_name": "string",
        "activity_id": "string",
        "activity_status": "string",
        "activity_details_id": "string",
        "payment_schedule_fictive": "string",
        "is_payable_manually": "string",
        "date": "2021-04-19",
        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",
        "payment_method": "CASH",
        "amount": "string",
        "paid_amount": "string",
        "paid": true,
        "paid_date": "2021-04-19",
        "note": "string",
        "payment_schedule": 1
    },
    ],
    "price_per_unit": {
        "id": 1,
        "rates_per_date": [
            {
                "rate": "string",
                "start_date": "2021-04-19",
                "end_date": "2021-04-19",
                "changed_date": "2021-04-19T09:16:55.367826",
                "changed_by": 1
            }
        ],
        "current_amount": "string",
        "amount": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19"
    },
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_id": 1,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
},
"details_name": "string",

```

(continues on next page)

(continued from previous page)

```

    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "status": "DRAFT",
    "approval_note": "string",
    "appropriation_date": "2021-04-19",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19",
    "activity_type": "MAIN_ACTIVITY",
    "note": "string",
    "details": 1,
    "approval_level": 1,
    "approval_user": 1,
    "modifies": 1,
    "appropriation": 1,
    "service_provider": 1
  }
]

```

POST /api/activities/

Expose activities in REST API.

Example request:

```

POST /api/activities/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "payment_plan": {
    "price_per_unit": {
      "amount": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19"
    },
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
  },
  "user_created": "string",
  "user_modified": "string",
  "status": "DRAFT",
  "approval_note": "string",
  "appropriation_date": "2021-04-19",

```

(continues on next page)

(continued from previous page)

```

"start_date": "2021-04-19",
"end_date": "2021-04-19",
"activity_type": "MAIN_ACTIVITY",
"note": "string",
"details": 1,
"approval_level": 1,
"approval_user": 1,
"modifies": 1,
"appropriation": 1,
"service_provider": 1
}

```

Status Codes

- 201 Created – Example response:

```

HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": 1,
  "monthly_payment_plan": "string",
  "total_cost": "string",
  "total_cost_this_year": "string",
  "total_cost_full_year": "string",
  "total_granted_this_year": "string",
  "total_expected_this_year": "string",
  "payment_plan": {
    "id": 1,
    "payments": [
      {
        "id": 1,
        "account_string": "string",
        "account_alias": "string",
        "payment_schedule_payment_id": "string",
        "case_cpr_number": "string",
        "case_name": "string",
        "activity_id": "string",
        "activity_status": "string",
        "activity_details_id": "string",
        "payment_schedule_fictive": "string",
        "is_payable_manually": "string",
        "date": "2021-04-19",
        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",
        "payment_method": "CASH",
        "amount": "string",
        "paid_amount": "string",
        "paid": true,
        "paid_date": "2021-04-19",
        "note": "string",
        "payment_schedule": 1
      }
    ],
    "price_per_unit": {

```

(continues on next page)

(continued from previous page)

```

        "id": 1,
        "rates_per_date": [
            {
                "rate": "string",
                "start_date": "2021-04-19",
                "end_date": "2021-04-19",
                "changed_date": "2021-04-19T09:16:55.367826",
                "changed_by": 1
            }
        ],
        "current_amount": "string",
        "amount": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19"
    },
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_id": 1,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
},
"details__name": "string",
"created": "2021-04-19T09:16:55.367826",
"modified": "2021-04-19T09:16:55.367826",
"user_created": "string",
"user_modified": "string",
"status": "DRAFT",
"approval_note": "string",
"appropriation_date": "2021-04-19",
"start_date": "2021-04-19",
"end_date": "2021-04-19",
"activity_type": "MAIN_ACTIVITY",
"note": "string",
"details": 1,
"approval_level": 1,
"approval_user": 1,
"modifies": 1,
"appropriation": 1,
"service_provider": 1
}

```

GET /api/activities/{id}/
Expose activities in REST API.

Parameters

- **id**(string)–

Query Parameters

- **created**(*string*) – created
- **modified**(*string*) – modified
- **user_created**(*string*) – user_created
- **user_modified**(*string*) – user_modified
- **details**(*string*) – details
- **status**(*string*) – status
- **approval_level**(*string*) – approval_level
- **approval_note**(*string*) – approval_note
- **approval_user**(*string*) – approval_user
- **appropriation_date**(*string*) – appropriation_date
- **start_date**(*string*) – start_date
- **end_date**(*string*) – end_date
- **activity_type**(*string*) – activity_type
- **modifies**(*string*) – modifies
- **appropriation**(*string*) – appropriation
- **service_provider**(*string*) – service_provider
- **note**(*string*) – note

Example request:

```
GET /api/activities/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "monthly_payment_plan": "string",
  "total_cost": "string",
  "total_cost_this_year": "string",
  "total_cost_full_year": "string",
  "total_granted_this_year": "string",
  "total_expected_this_year": "string",
  "payment_plan": {
    "id": 1,
    "payments": [
      {
        "id": 1,
        "account_string": "string",
        "account_alias": "string",
        "payment_schedule_payment_id": "string",
```

(continues on next page)

(continued from previous page)

```

        "case__cpr_number": "string",
        "case__name": "string",
        "activity__id": "string",
        "activity__status": "string",
        "activity__details__id": "string",
        "payment_schedule__fictive": "string",
        "is_payable_manually": "string",
        "date": "2021-04-19",
        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",
        "payment_method": "CASH",
        "amount": "string",
        "paid_amount": "string",
        "paid": true,
        "paid_date": "2021-04-19",
        "note": "string",
        "payment_schedule": 1
    },
    ],
    "price_per_unit": {
        "id": 1,
        "rates_per_date": [
            {
                "rate": "string",
                "start_date": "2021-04-19",
                "end_date": "2021-04-19",
                "changed_date": "2021-04-19T09:16:55.367826",
                "changed_by": 1
            }
        ],
        "current_amount": "string",
        "amount": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19"
    },
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_id": 1,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
},
"details__name": "string",
"created": "2021-04-19T09:16:55.367826",
"modified": "2021-04-19T09:16:55.367826",
"user_created": "string",
"user_modified": "string",

```

(continues on next page)

(continued from previous page)

```

    "status": "DRAFT",
    "approval_note": "string",
    "appropriation_date": "2021-04-19",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19",
    "activity_type": "MAIN_ACTIVITY",
    "note": "string",
    "details": 1,
    "approval_level": 1,
    "approval_user": 1,
    "modifies": 1,
    "appropriation": 1,
    "service_provider": 1
  }

```

PUT /api/activities/{id}/
Expose activities in REST API.

Parameters

- **id** (*string*) –

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **details** (*string*) – details
- **status** (*string*) – status
- **approval_level** (*string*) – approval_level
- **approval_note** (*string*) – approval_note
- **approval_user** (*string*) – approval_user
- **appropriation_date** (*string*) – appropriation_date
- **start_date** (*string*) – start_date
- **end_date** (*string*) – end_date
- **activity_type** (*string*) – activity_type
- **modifies** (*string*) – modifies
- **appropriation** (*string*) – appropriation
- **service_provider** (*string*) – service_provider
- **note** (*string*) – note

Example request:

```

PUT /api/activities/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

```

(continues on next page)

(continued from previous page)

```
{
  "payment_plan": {
    "price_per_unit": {
      "amount": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19"
    },
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
  },
  "user_created": "string",
  "user_modified": "string",
  "status": "DRAFT",
  "approval_note": "string",
  "appropriation_date": "2021-04-19",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19",
  "activity_type": "MAIN_ACTIVITY",
  "note": "string",
  "details": 1,
  "approval_level": 1,
  "approval_user": 1,
  "modifies": 1,
  "appropriation": 1,
  "service_provider": 1
}
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "monthly_payment_plan": "string",
  "total_cost": "string",
  "total_cost_this_year": "string",
  "total_cost_full_year": "string",
  "total_granted_this_year": "string",
  "total_expected_this_year": "string",
  "payment_plan": {
    "id": 1,
```

(continues on next page)

(continued from previous page)

```

"payments": [
  {
    "id": 1,
    "account_string": "string",
    "account_alias": "string",
    "payment_schedule_payment_id": "string",
    "case_cpr_number": "string",
    "case_name": "string",
    "activity_id": "string",
    "activity_status": "string",
    "activity_details_id": "string",
    "payment_schedule_fictive": "string",
    "is_payable_manually": "string",
    "date": "2021-04-19",
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "amount": "string",
    "paid_amount": "string",
    "paid": true,
    "paid_date": "2021-04-19",
    "note": "string",
    "payment_schedule": 1
  }
],
"price_per_unit": {
  "id": 1,
  "rates_per_date": [
    {
      "rate": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19",
      "changed_date": "2021-04-19T09:16:55.367826",
      "changed_by": 1
    }
  ],
  "current_amount": "string",
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
},
"recipient_type": "INTERNAL",
"recipient_id": "string",
"recipient_name": "string",
"payment_method": "CASH",
"payment_frequency": "DAILY",
"payment_date": "2021-04-19",
"payment_day_of_month": 1,
"payment_type": "ONE_TIME_PAYMENT",
"payment_units": "string",
"payment_amount": "string",
"fictive": true,
"payment_id": 1,
"payment_cost_type": "FIXED",
"payment_method_details": 1,
"payment_rate": "string"

```

(continues on next page)

(continued from previous page)

```
    },
    "details_name": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "status": "DRAFT",
    "approval_note": "string",
    "appropriation_date": "2021-04-19",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19",
    "activity_type": "MAIN_ACTIVITY",
    "note": "string",
    "details": 1,
    "approval_level": 1,
    "approval_user": 1,
    "modifies": 1,
    "appropriation": 1,
    "service_provider": 1
  }
}
```

PATCH /api/activities/{id}/
Expose activities in REST API.

Parameters

- **id** (*string*) –

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **details** (*string*) – details
- **status** (*string*) – status
- **approval_level** (*string*) – approval_level
- **approval_note** (*string*) – approval_note
- **approval_user** (*string*) – approval_user
- **appropriation_date** (*string*) – appropriation_date
- **start_date** (*string*) – start_date
- **end_date** (*string*) – end_date
- **activity_type** (*string*) – activity_type
- **modifies** (*string*) – modifies
- **appropriation** (*string*) – appropriation
- **service_provider** (*string*) – service_provider
- **note** (*string*) – note

Example request:

```

PATCH /api/activities/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "payment_plan": {
    "price_per_unit": {
      "amount": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19"
    },
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
  },
  "user_created": "string",
  "user_modified": "string",
  "status": "DRAFT",
  "approval_note": "string",
  "appropriation_date": "2021-04-19",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19",
  "activity_type": "MAIN_ACTIVITY",
  "note": "string",
  "details": 1,
  "approval_level": 1,
  "approval_user": 1,
  "modifies": 1,
  "appropriation": 1,
  "service_provider": 1
}

```

Status Codes

- 200 OK – Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "monthly_payment_plan": "string",
  "total_cost": "string",
  "total_cost_this_year": "string",
  "total_cost_full_year": "string",

```

(continues on next page)

(continued from previous page)

```

"total_granted_this_year": "string",
"total_expected_this_year": "string",
"payment_plan": {
  "id": 1,
  "payments": [
    {
      "id": 1,
      "account_string": "string",
      "account_alias": "string",
      "payment_schedule_payment_id": "string",
      "case_cpr_number": "string",
      "case_name": "string",
      "activity_id": "string",
      "activity_status": "string",
      "activity_details_id": "string",
      "payment_schedule_fictive": "string",
      "is_payable_manually": "string",
      "date": "2021-04-19",
      "recipient_type": "INTERNAL",
      "recipient_id": "string",
      "recipient_name": "string",
      "payment_method": "CASH",
      "amount": "string",
      "paid_amount": "string",
      "paid": true,
      "paid_date": "2021-04-19",
      "note": "string",
      "payment_schedule": 1
    }
  ],
  "price_per_unit": {
    "id": 1,
    "rates_per_date": [
      {
        "rate": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826",
        "changed_by": 1
      }
    ],
    "current_amount": "string",
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
  },
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "payment_frequency": "DAILY",
  "payment_date": "2021-04-19",
  "payment_day_of_month": 1,
  "payment_type": "ONE_TIME_PAYMENT",
  "payment_units": "string",
  "payment_amount": "string",
  "fictive": true,

```

(continues on next page)

(continued from previous page)

```

        "payment_id": 1,
        "payment_cost_type": "FIXED",
        "payment_method_details": 1,
        "payment_rate": "string"
    },
    "details_name": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "status": "DRAFT",
    "approval_note": "string",
    "appropriation_date": "2021-04-19",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19",
    "activity_type": "MAIN_ACTIVITY",
    "note": "string",
    "details": 1,
    "approval_level": 1,
    "approval_user": 1,
    "modifies": 1,
    "appropriation": 1,
    "service_provider": 1
}

```

DELETE /api/activities/{id}/

Handle deletion according to their business logic.

Drafts are deleted, expectations are logically deleted, granted activities cannot be deleted.

Parameters

- **id** (*string*) –

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **details** (*string*) – details
- **status** (*string*) – status
- **approval_level** (*string*) – approval_level
- **approval_note** (*string*) – approval_note
- **approval_user** (*string*) – approval_user
- **appropriation_date** (*string*) – appropriation_date
- **start_date** (*string*) – start_date
- **end_date** (*string*) – end_date
- **activity_type** (*string*) – activity_type
- **modifies** (*string*) – modifies

- **appropriation** (*string*) – appropriation
- **service_provider** (*string*) – service_provider
- **note** (*string*) – note

Status Codes

- 204 No Content –

GET /api/payment_schedules/

Expose payment schedules in REST API.

Example request:

```
GET /api/payment_schedules/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "payments": [
      {
        "id": 1,
        "account_string": "string",
        "account_alias": "string",
        "payment_schedule_payment_id": "string",
        "case_cpr_number": "string",
        "case_name": "string",
        "activity_id": "string",
        "activity_status": "string",
        "activity_details_id": "string",
        "payment_schedule_fictive": "string",
        "is_payable_manually": "string",
        "date": "2021-04-19",
        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",
        "payment_method": "CASH",
        "amount": "string",
        "paid_amount": "string",
        "paid": true,
        "paid_date": "2021-04-19",
        "note": "string",
        "payment_schedule": 1
      }
    ],
    "price_per_unit": {
      "id": 1,
      "rates_per_date": [
        {
          "rate": "string",
          "start_date": "2021-04-19",

```

(continues on next page)

(continued from previous page)

```

        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826",
        "changed_by": 1
    }
],
"current_amount": "string",
"amount": "string",
"start_date": "2021-04-19",
"end_date": "2021-04-19"
},
"recipient_type": "INTERNAL",
"recipient_id": "string",
"recipient_name": "string",
"payment_method": "CASH",
"payment_frequency": "DAILY",
"payment_date": "2021-04-19",
"payment_day_of_month": 1,
"payment_type": "ONE_TIME_PAYMENT",
"payment_units": "string",
"payment_amount": "string",
"fictive": true,
"payment_id": 1,
"payment_cost_type": "FIXED",
"payment_method_details": 1,
"payment_rate": "string"
}
]

```

POST /api/payment_schedules/

Expose payment schedules in REST API.

Example request:

```

POST /api/payment_schedules/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "price_per_unit": {
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
  },
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "payment_frequency": "DAILY",
  "payment_date": "2021-04-19",
  "payment_day_of_month": 1,
  "payment_type": "ONE_TIME_PAYMENT",
  "payment_units": "string",
  "payment_amount": "string",
  "fictive": true,
  "payment_cost_type": "FIXED",
  "payment_method_details": 1,

```

(continues on next page)

(continued from previous page)

```

    "payment_rate": "string"
  }

```

Status Codes

- 201 Created – Example response:

```

HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": 1,
  "payments": [
    {
      "id": 1,
      "account_string": "string",
      "account_alias": "string",
      "payment_schedule_payment_id": "string",
      "case_cpr_number": "string",
      "case_name": "string",
      "activity_id": "string",
      "activity_status": "string",
      "activity_details_id": "string",
      "payment_schedule_fictive": "string",
      "is_payable_manually": "string",
      "date": "2021-04-19",
      "recipient_type": "INTERNAL",
      "recipient_id": "string",
      "recipient_name": "string",
      "payment_method": "CASH",
      "amount": "string",
      "paid_amount": "string",
      "paid": true,
      "paid_date": "2021-04-19",
      "note": "string",
      "payment_schedule": 1
    }
  ],
  "price_per_unit": {
    "id": 1,
    "rates_per_date": [
      {
        "rate": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826",
        "changed_by": 1
      }
    ],
    "current_amount": "string",
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
  },
  "recipient_type": "INTERNAL",
  "recipient_id": "string",

```

(continues on next page)

(continued from previous page)

```

    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_id": 1,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
  }

```

GET /api/payment_schedules/{id}/
Expose payment schedules in REST API.

Parameters

- **id** (*string*) –

Example request:

```

GET /api/payment_schedules/{id}/ HTTP/1.1
Host: example.com

```

Status Codes

- **200 OK** – Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "payments": [
    {
      "id": 1,
      "account_string": "string",
      "account_alias": "string",
      "payment_schedule_payment_id": "string",
      "case_cpr_number": "string",
      "case_name": "string",
      "activity_id": "string",
      "activity_status": "string",
      "activity_details_id": "string",
      "payment_schedule_fictive": "string",
      "is_payable_manually": "string",
      "date": "2021-04-19",
      "recipient_type": "INTERNAL",
      "recipient_id": "string",
      "recipient_name": "string",
      "payment_method": "CASH",
      "amount": "string",
      "paid_amount": "string",
      "paid": true,

```

(continues on next page)

(continued from previous page)

```

        "paid_date": "2021-04-19",
        "note": "string",
        "payment_schedule": 1
    }
],
"price_per_unit": {
    "id": 1,
    "rates_per_date": [
        {
            "rate": "string",
            "start_date": "2021-04-19",
            "end_date": "2021-04-19",
            "changed_date": "2021-04-19T09:16:55.367826",
            "changed_by": 1
        }
    ],
    "current_amount": "string",
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
},
"recipient_type": "INTERNAL",
"recipient_id": "string",
"recipient_name": "string",
"payment_method": "CASH",
"payment_frequency": "DAILY",
"payment_date": "2021-04-19",
"payment_day_of_month": 1,
"payment_type": "ONE_TIME_PAYMENT",
"payment_units": "string",
"payment_amount": "string",
"fictive": true,
"payment_id": 1,
"payment_cost_type": "FIXED",
"payment_method_details": 1,
"payment_rate": "string"
}

```

PUT /api/payment_schedules/{id}/
Expose payment schedules in REST API.

Parameters

- **id**(string)–

Example request:

```

PUT /api/payment_schedules/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "price_per_unit": {
        "amount": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19"
    },

```

(continues on next page)

(continued from previous page)

```

"recipient_type": "INTERNAL",
"recipient_id": "string",
"recipient_name": "string",
"payment_method": "CASH",
"payment_frequency": "DAILY",
"payment_date": "2021-04-19",
"payment_day_of_month": 1,
"payment_type": "ONE_TIME_PAYMENT",
"payment_units": "string",
"payment_amount": "string",
"fictive": true,
"payment_cost_type": "FIXED",
"payment_method_details": 1,
"payment_rate": "string"
}

```

Status Codes

- 200 OK – Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "payments": [
    {
      "id": 1,
      "account_string": "string",
      "account_alias": "string",
      "payment_schedule_payment_id": "string",
      "case_cpr_number": "string",
      "case_name": "string",
      "activity_id": "string",
      "activity_status": "string",
      "activity_details_id": "string",
      "payment_schedule_fictive": "string",
      "is_payable_manually": "string",
      "date": "2021-04-19",
      "recipient_type": "INTERNAL",
      "recipient_id": "string",
      "recipient_name": "string",
      "payment_method": "CASH",
      "amount": "string",
      "paid_amount": "string",
      "paid": true,
      "paid_date": "2021-04-19",
      "note": "string",
      "payment_schedule": 1
    }
  ],
  "price_per_unit": {
    "id": 1,
    "rates_per_date": [
      {
        "rate": "string",

```

(continues on next page)

(continued from previous page)

```

        "start_date": "2021-04-19",
        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826",
        "changed_by": 1
    },
    ],
    "current_amount": "string",
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
},
"recipient_type": "INTERNAL",
"recipient_id": "string",
"recipient_name": "string",
"payment_method": "CASH",
"payment_frequency": "DAILY",
"payment_date": "2021-04-19",
"payment_day_of_month": 1,
"payment_type": "ONE_TIME_PAYMENT",
"payment_units": "string",
"payment_amount": "string",
"fictive": true,
"payment_id": 1,
"payment_cost_type": "FIXED",
"payment_method_details": 1,
"payment_rate": "string"
}

```

PATCH /api/payment_schedules/{id}/

Expose payment schedules in REST API.

Parameters

- **id**(string)–

Example request:

```

PATCH /api/payment_schedules/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "price_per_unit": {
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
  },
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "payment_frequency": "DAILY",
  "payment_date": "2021-04-19",
  "payment_day_of_month": 1,
  "payment_type": "ONE_TIME_PAYMENT",
  "payment_units": "string",
  "payment_amount": "string",

```

(continues on next page)

(continued from previous page)

```

"fictive": true,
"payment_cost_type": "FIXED",
"payment_method_details": 1,
"payment_rate": "string"
}

```

Status Codes

- 200 OK – Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "payments": [
    {
      "id": 1,
      "account_string": "string",
      "account_alias": "string",
      "payment_schedule_payment_id": "string",
      "case_cpr_number": "string",
      "case_name": "string",
      "activity_id": "string",
      "activity_status": "string",
      "activity_details_id": "string",
      "payment_schedule_fictive": "string",
      "is_payable_manually": "string",
      "date": "2021-04-19",
      "recipient_type": "INTERNAL",
      "recipient_id": "string",
      "recipient_name": "string",
      "payment_method": "CASH",
      "amount": "string",
      "paid_amount": "string",
      "paid": true,
      "paid_date": "2021-04-19",
      "note": "string",
      "payment_schedule": 1
    }
  ],
  "price_per_unit": {
    "id": 1,
    "rates_per_date": [
      {
        "rate": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826",
        "changed_by": 1
      }
    ],
    "current_amount": "string",
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
  }
}

```

(continues on next page)

(continued from previous page)

```
    },
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "payment_frequency": "DAILY",
    "payment_date": "2021-04-19",
    "payment_day_of_month": 1,
    "payment_type": "ONE_TIME_PAYMENT",
    "payment_units": "string",
    "payment_amount": "string",
    "fictive": true,
    "payment_id": 1,
    "payment_cost_type": "FIXED",
    "payment_method_details": 1,
    "payment_rate": "string"
  }
}
```

DELETE /api/payment_schedules/{id}/

Expose payment schedules in REST API.

Parameters

- **id** (*string*) –

Status Codes

- 204 No Content –

GET /api/rates/

Expose rates in REST API.

Example request:

```
GET /api/rates/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – **Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "rates_per_date": [
      {
        "rate": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826",
        "changed_by": 1
      }
    ],
    "active": true,
    "name": "string",
  }
]
```

(continues on next page)

(continued from previous page)

```

        "description": "string",
        "needs_recalculation": true
    }
]

```

GET /api/rates/{id}/
Expose rates in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this takst.

Example request:

```

GET /api/rates/{id}/ HTTP/1.1
Host: example.com

```

Status Codes

- **200 OK – Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "rates_per_date": [
    {
      "rate": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19",
      "changed_date": "2021-04-19T09:16:55.367826",
      "changed_by": 1
    }
  ],
  "active": true,
  "name": "string",
  "description": "string",
  "needs_recalculation": true
}

```

GET /api/prices/
Expose Price objects in REST API.

Example request:

```

GET /api/prices/ HTTP/1.1
Host: example.com

```

Status Codes

- **200 OK – Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

```

(continues on next page)

(continued from previous page)

```
[
  {
    "id": 1,
    "rates_per_date": [
      {
        "rate": "string",
        "start_date": "2021-04-19",
        "end_date": "2021-04-19",
        "changed_date": "2021-04-19T09:16:55.367826",
        "changed_by": 1
      }
    ],
    "current_amount": "string",
    "amount": "string",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19"
  }
]
```

POST /api/prices/

Expose Price objects in REST API.

Example request:

```
POST /api/prices/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
}
```

Status Codes

- 201 Created – Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": 1,
  "rates_per_date": [
    {
      "rate": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19",
      "changed_date": "2021-04-19T09:16:55.367826",
      "changed_by": 1
    }
  ],
  "current_amount": "string",
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
}
```

(continues on next page)

(continued from previous page)

```
}
  "end_date": "2021-04-19"
```

GET /api/prices/{id}/

Expose Price objects in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this pris.

Example request:

```
GET /api/prices/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "rates_per_date": [
    {
      "rate": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19",
      "changed_date": "2021-04-19T09:16:55.367826",
      "changed_by": 1
    }
  ],
  "current_amount": "string",
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
}
```

PUT /api/prices/{id}/

Expose Price objects in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this pris.

Example request:

```
PUT /api/prices/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
}
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "rates_per_date": [
    {
      "rate": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19",
      "changed_date": "2021-04-19T09:16:55.367826",
      "changed_by": 1
    }
  ],
  "current_amount": "string",
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
}
```

PATCH /api/prices/{id}/

Expose Price objects in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this pris.

Example request:

```
PATCH /api/prices/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
}
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "rates_per_date": [
    {
      "rate": "string",
      "start_date": "2021-04-19",
      "end_date": "2021-04-19",
      "changed_date": "2021-04-19T09:16:55.367826",
      "changed_by": 1
    }
  ],
  "current_amount": "string",
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
}
```

(continues on next page)

(continued from previous page)

```

    }
  ],
  "current_amount": "string",
  "amount": "string",
  "start_date": "2021-04-19",
  "end_date": "2021-04-19"
}

```

DELETE /api/prices/{id}/

Expose Price objects in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this pris.

Status Codes

- 204 No Content –

GET /api/payment_method_details/

Expose payment method details in REST API.

Example request:

```

GET /api/payment_method_details/ HTTP/1.1
Host: example.com

```

Status Codes

- 200 OK – **Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "tax_card": "MAIN_CARD"
  }
]

```

POST /api/payment_method_details/

Expose payment method details in REST API.

Example request:

```

POST /api/payment_method_details/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "tax_card": "MAIN_CARD"
}

```

Status Codes

- 201 Created – **Example response:**

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": 1,
  "tax_card": "MAIN_CARD"
}
```

GET /api/payment_method_details/{id}/
Expose payment method details in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this betalingsmåde detalje.

Example request:

```
GET /api/payment_method_details/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "tax_card": "MAIN_CARD"
}
```

PUT /api/payment_method_details/{id}/
Expose payment method details in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this betalingsmåde detalje.

Example request:

```
PUT /api/payment_method_details/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "tax_card": "MAIN_CARD"
}
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
```

(continues on next page)

(continued from previous page)

```
"tax_card": "MAIN_CARD"
}
```

PATCH /api/payment_method_details/{id}/

Expose payment method details in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this betalingsmåde detalje.

Example request:

```
PATCH /api/payment_method_details/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "tax_card": "MAIN_CARD"
}
```

Status Codes

- **200 OK** – **Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "tax_card": "MAIN_CARD"
}
```

DELETE /api/payment_method_details/{id}/

Expose payment method details in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this betalingsmåde detalje.

Status Codes

- **204 No Content** –

GET /api/payments/

Expose payments in REST API.

Note, this viewset supports pagination.

Query Parameters

- **page** (*integer*) – A page number within the paginated result set.
- **date** (*string*) – date
- **recipient_type** (*string*) – recipient_type
- **recipient_id** (*string*) – recipient_id
- **recipient_name** (*string*) – recipient_name
- **payment_method** (*string*) – payment_method

- **amount** (*string*) – amount
- **paid_amount** (*string*) – paid_amount
- **paid** (*string*) – paid
- **paid_date** (*string*) – paid_date
- **note** (*string*) – note
- **saved_account_string** (*string*) – saved_account_string
- **saved_account_alias** (*string*) – saved_account_alias
- **payment_schedule** (*string*) – payment_schedule
- **case** (*string*) – Sag
- **activity** (*string*) – Aktivitet
- **paid_date__gte** (*string*) – Betalingsdato større eller lig med
- **paid_date__lte** (*string*) – Betalingsdato mindre eller lig med
- **date__gte** (*string*) – Dato større eller lig med
- **date__lte** (*string*) – Dato mindre eller lig med
- **paid_date_or_date__gte** (*string*) – Betalingsdato eller Dato større eller lig med
- **paid_date_or_date__lte** (*string*) – Betalingsdato eller Dato mindre eller lig med
- **paid_date_or_date_week** (*string*) – Betalingsdato eller Dato for uge
- **paid_date_or_date_month** (*string*) – Betalingsdato eller Dato for måned
- **paid_date_or_date_year** (*string*) – Betalingsdato eller Dato for år

Example request:

```
GET /api/payments/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "count": 1,
  "next": "https://example.com",
  "previous": "https://example.com",
  "results": [
    {
      "id": 1,
      "account_string": "string",
      "account_alias": "string",
      "payment_schedule_payment_id": "string",
      "case_cpr_number": "string",
      "case_name": "string",
      "activity_id": "string",
      "activity_status": "string",
      "activity_details_id": "string",
```

(continues on next page)

(continued from previous page)

```

        "payment_schedule_fictive": "string",
        "is_payable_manually": "string",
        "date": "2021-04-19",
        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",
        "payment_method": "CASH",
        "amount": "string",
        "paid_amount": "string",
        "paid": true,
        "paid_date": "2021-04-19",
        "note": "string",
        "payment_schedule": 1
    }
}

```

POST /api/payments/

Expose payments in REST API.

Note, this viewset supports pagination.

Example request:

```

POST /api/payments/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "date": "2021-04-19",
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "amount": "string",
  "paid_amount": "string",
  "paid": true,
  "paid_date": "2021-04-19",
  "note": "string",
  "payment_schedule": 1
}

```

Status Codes

- 201 Created – Example response:

```

HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": 1,
  "account_string": "string",
  "account_alias": "string",
  "payment_schedule_payment_id": "string",
  "case_cpr_number": "string",
  "case_name": "string",

```

(continues on next page)

(continued from previous page)

```
"activity__id": "string",
"activity__status": "string",
"activity__details__id": "string",
"payment_schedule__fictive": "string",
"is_payable_manually": "string",
"date": "2021-04-19",
"recipient_type": "INTERNAL",
"recipient_id": "string",
"recipient_name": "string",
"payment_method": "CASH",
"amount": "string",
"paid_amount": "string",
"paid": true,
"paid_date": "2021-04-19",
"note": "string",
"payment_schedule": 1
}
```

GET /api/payments/{id}/

Expose payments in REST API.

Note, this viewset supports pagination.

Parameters

- **id** (*string*) – A unique integer value identifying this betaling.

Query Parameters

- **date** (*string*) – date
- **recipient_type** (*string*) – recipient_type
- **recipient_id** (*string*) – recipient_id
- **recipient_name** (*string*) – recipient_name
- **payment_method** (*string*) – payment_method
- **amount** (*string*) – amount
- **paid_amount** (*string*) – paid_amount
- **paid** (*string*) – paid
- **paid_date** (*string*) – paid_date
- **note** (*string*) – note
- **saved_account_string** (*string*) – saved_account_string
- **saved_account_alias** (*string*) – saved_account_alias
- **payment_schedule** (*string*) – payment_schedule
- **case** (*string*) – Sag
- **activity** (*string*) – Aktivitet
- **paid_date__gte** (*string*) – Betalingsdato større eller lig med
- **paid_date__lte** (*string*) – Betalingsdato mindre eller lig med
- **date__gte** (*string*) – Dato større eller lig med

- **date__lte**(*string*) – Dato mindre eller lig med
- **paid_date_or_date__gte**(*string*) – Betalingsdato eller Dato større eller lig med
- **paid_date_or_date__lte**(*string*) – Betalingsdato eller Dato mindre eller lig med
- **paid_date_or_date__week**(*string*) – Betalingsdato eller Dato for uge
- **paid_date_or_date__month**(*string*) – Betalingsdato eller Dato for måned
- **paid_date_or_date__year**(*string*) – Betalingsdato eller Dato for år

Example request:

```
GET /api/payments/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "account_string": "string",
  "account_alias": "string",
  "payment_schedule_payment_id": "string",
  "case_cpr_number": "string",
  "case_name": "string",
  "activity_id": "string",
  "activity_status": "string",
  "activity_details_id": "string",
  "payment_schedule_fictive": "string",
  "is_payable_manually": "string",
  "date": "2021-04-19",
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "amount": "string",
  "paid_amount": "string",
  "paid": true,
  "paid_date": "2021-04-19",
  "note": "string",
  "payment_schedule": 1
}
```

PUT /api/payments/{id}/
Expose payments in REST API.

Note, this viewset supports pagination.

Parameters

- **id**(*string*) – A unique integer value identifying this betaling.

Query Parameters

- **date**(*string*) – date

- **recipient_type** (*string*) – recipient_type
- **recipient_id** (*string*) – recipient_id
- **recipient_name** (*string*) – recipient_name
- **payment_method** (*string*) – payment_method
- **amount** (*string*) – amount
- **paid_amount** (*string*) – paid_amount
- **paid** (*string*) – paid
- **paid_date** (*string*) – paid_date
- **note** (*string*) – note
- **saved_account_string** (*string*) – saved_account_string
- **saved_account_alias** (*string*) – saved_account_alias
- **payment_schedule** (*string*) – payment_schedule
- **case** (*string*) – Sag
- **activity** (*string*) – Aktivitet
- **paid_date_gte** (*string*) – Betalingsdato større eller lig med
- **paid_date_lte** (*string*) – Betalingsdato mindre eller lig med
- **date_gte** (*string*) – Dato større eller lig med
- **date_lte** (*string*) – Dato mindre eller lig med
- **paid_date_or_date_gte** (*string*) – Betalingsdato eller Dato større eller lig med
- **paid_date_or_date_lte** (*string*) – Betalingsdato eller Dato mindre eller lig med
- **paid_date_or_date_week** (*string*) – Betalingsdato eller Dato for uge
- **paid_date_or_date_month** (*string*) – Betalingsdato eller Dato for måned
- **paid_date_or_date_year** (*string*) – Betalingsdato eller Dato for år

Example request:

```
PUT /api/payments/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "date": "2021-04-19",
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "amount": "string",
  "paid_amount": "string",
  "paid": true,
  "paid_date": "2021-04-19",
  "note": "string",
  "payment_schedule": 1
}
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "account_string": "string",
  "account_alias": "string",
  "payment_schedule_payment_id": "string",
  "case_cpr_number": "string",
  "case_name": "string",
  "activity_id": "string",
  "activity_status": "string",
  "activity_details_id": "string",
  "payment_schedule_fictive": "string",
  "is_payable_manually": "string",
  "date": "2021-04-19",
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "amount": "string",
  "paid_amount": "string",
  "paid": true,
  "paid_date": "2021-04-19",
  "note": "string",
  "payment_schedule": 1
}
```

PATCH /api/payments/{id}/

Expose payments in REST API.

Note, this viewset supports pagination.

Parameters

- **id** (*string*) – A unique integer value identifying this betaling.

Query Parameters

- **date** (*string*) – date
- **recipient_type** (*string*) – recipient_type
- **recipient_id** (*string*) – recipient_id
- **recipient_name** (*string*) – recipient_name
- **payment_method** (*string*) – payment_method
- **amount** (*string*) – amount
- **paid_amount** (*string*) – paid_amount
- **paid** (*string*) – paid
- **paid_date** (*string*) – paid_date
- **note** (*string*) – note
- **saved_account_string** (*string*) – saved_account_string
- **saved_account_alias** (*string*) – saved_account_alias

- **payment_schedule** (*string*) – payment_schedule
- **case** (*string*) – Sag
- **activity** (*string*) – Aktivitet
- **paid_date__gte** (*string*) – Betalingsdato større eller lig med
- **paid_date__lte** (*string*) – Betalingsdato mindre eller lig med
- **date__gte** (*string*) – Dato større eller lig med
- **date__lte** (*string*) – Dato mindre eller lig med
- **paid_date_or_date__gte** (*string*) – Betalingsdato eller Dato større eller lig med
- **paid_date_or_date__lte** (*string*) – Betalingsdato eller Dato mindre eller lig med
- **paid_date_or_date__week** (*string*) – Betalingsdato eller Dato for uge
- **paid_date_or_date__month** (*string*) – Betalingsdato eller Dato for måned
- **paid_date_or_date__year** (*string*) – Betalingsdato eller Dato for år

Example request:

```
PATCH /api/payments/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "date": "2021-04-19",
  "recipient_type": "INTERNAL",
  "recipient_id": "string",
  "recipient_name": "string",
  "payment_method": "CASH",
  "amount": "string",
  "paid_amount": "string",
  "paid": true,
  "paid_date": "2021-04-19",
  "note": "string",
  "payment_schedule": 1
}
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "account_string": "string",
  "account_alias": "string",
  "payment_schedule_payment_id": "string",
  "case_cpr_number": "string",
  "case_name": "string",
  "activity_id": "string",
  "activity_status": "string",
  "activity_details_id": "string",
  "payment_schedule_fictive": "string",

```

(continues on next page)

(continued from previous page)

```

    "is_payable_manually": "string",
    "date": "2021-04-19",
    "recipient_type": "INTERNAL",
    "recipient_id": "string",
    "recipient_name": "string",
    "payment_method": "CASH",
    "amount": "string",
    "paid_amount": "string",
    "paid": true,
    "paid_date": "2021-04-19",
    "note": "string",
    "payment_schedule": 1
}

```

DELETE /api/payments/{id}/

Expose payments in REST API.

Note, this viewset supports pagination.

Parameters

- **id** (*string*) – A unique integer value identifying this betaling.

Query Parameters

- **date** (*string*) – date
- **recipient_type** (*string*) – recipient_type
- **recipient_id** (*string*) – recipient_id
- **recipient_name** (*string*) – recipient_name
- **payment_method** (*string*) – payment_method
- **amount** (*string*) – amount
- **paid_amount** (*string*) – paid_amount
- **paid** (*string*) – paid
- **paid_date** (*string*) – paid_date
- **note** (*string*) – note
- **saved_account_string** (*string*) – saved_account_string
- **saved_account_alias** (*string*) – saved_account_alias
- **payment_schedule** (*string*) – payment_schedule
- **case** (*string*) – Sag
- **activity** (*string*) – Aktivitet
- **paid_date__gte** (*string*) – Betalingsdato større eller lig med
- **paid_date__lte** (*string*) – Betalingsdato mindre eller lig med
- **date__gte** (*string*) – Dato større eller lig med
- **date__lte** (*string*) – Dato mindre eller lig med
- **paid_date_or_date__gte** (*string*) – Betalingsdato eller Dato større eller lig med
- **paid_date_or_date__lte** (*string*) – Betalingsdato eller Dato mindre eller lig med

- **paid_date_or_date_week** (*string*) – Betalingsdato eller Dato for uge
- **paid_date_or_date_month** (*string*) – Betalingsdato eller Dato for måned
- **paid_date_or_date_year** (*string*) – Betalingsdato eller Dato for år

Status Codes

- 204 No Content –

GET /api/related_persons/

Expose related persons - typically family relations - in REST API.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **relation_type** (*string*) – relation_type
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **related_case** (*string*) – related_case
- **from_serviceplatformen** (*string*) – from_serviceplatformen
- **main_case** (*string*) – main_case

Example request:

```
GET /api/related_persons/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "relation_type": "string",
    "cpr_number": "string",
    "name": "string",
    "related_case": "string",
    "from_serviceplatformen": true,
    "main_case": 1
  }
]
```

POST /api/related_persons/

Expose related persons - typically family relations - in REST API.

Example request:

```
POST /api/related_persons/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "user_created": "string",
  "user_modified": "string",
  "relation_type": "string",
  "cpr_number": "string",
  "name": "string",
  "related_case": "string",
  "from_serviceplatformen": true,
  "main_case": 1
}
```

Status Codes

- 201 Created – Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": 1,
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",
  "user_modified": "string",
  "relation_type": "string",
  "cpr_number": "string",
  "name": "string",
  "related_case": "string",
  "from_serviceplatformen": true,
  "main_case": 1
}
```

GET /api/related_persons/fetch_from_serviceplatformen/

Fetch relations for a person using the CPR from Serviceplatformen.

Returns the data as serialized RelatedPersons data.

GET params: cpr

Example request:

```
GET /api/related_persons/fetch_from_serviceplatformen/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",
  "user_modified": "string",
  "relation_type": "string",
  "cpr_number": "string",
  "name": "string",
  "related_case": "string",
  "from_serviceplatformen": true,
  "main_case": 1
}
```

GET `/api/related_persons/{id}/`

Expose related persons - typically family relations - in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this related person.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **relation_type** (*string*) – relation_type
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **related_case** (*string*) – related_case
- **from_serviceplatformen** (*string*) – from_serviceplatformen
- **main_case** (*string*) – main_case

Example request:

```
GET /api/related_persons/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK** – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "created": "2021-04-19T09:16:55.367826",
```

(continues on next page)

(continued from previous page)

```

    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "relation_type": "string",
    "cpr_number": "string",
    "name": "string",
    "related_case": "string",
    "from_serviceplatformen": true,
    "main_case": 1
  }

```

PUT /api/related_persons/{id}/

Expose related persons - typically family relations - in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this related person.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **relation_type** (*string*) – relation_type
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **related_case** (*string*) – related_case
- **from_serviceplatformen** (*string*) – from_serviceplatformen
- **main_case** (*string*) – main_case

Example request:

```

PUT /api/related_persons/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "user_created": "string",
  "user_modified": "string",
  "relation_type": "string",
  "cpr_number": "string",
  "name": "string",
  "related_case": "string",
  "from_serviceplatformen": true,
  "main_case": 1
}

```

Status Codes

- **200 OK** – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",
  "user_modified": "string",
  "relation_type": "string",
  "cpr_number": "string",
  "name": "string",
  "related_case": "string",
  "from_serviceplatformen": true,
  "main_case": 1
}
```

PATCH /api/related_persons/{id}/

Expose related persons - typically family relations - in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this related person.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **relation_type** (*string*) – relation_type
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **related_case** (*string*) – related_case
- **from_serviceplatformen** (*string*) – from_serviceplatformen
- **main_case** (*string*) – main_case

Example request:

```
PATCH /api/related_persons/{id}/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "user_created": "string",
  "user_modified": "string",
  "relation_type": "string",
  "cpr_number": "string",
  "name": "string",
  "related_case": "string",
  "from_serviceplatformen": true,
  "main_case": 1
}
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",
  "user_modified": "string",
  "relation_type": "string",
  "cpr_number": "string",
  "name": "string",
  "related_case": "string",
  "from_serviceplatformen": true,
  "main_case": 1
}
```

DELETE /api/related_persons/{id}/

Expose related persons - typically family relations - in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this related person.

Query Parameters

- **created** (*string*) – created
- **modified** (*string*) – modified
- **user_created** (*string*) – user_created
- **user_modified** (*string*) – user_modified
- **relation_type** (*string*) – relation_type
- **cpr_number** (*string*) – cpr_number
- **name** (*string*) – name
- **related_case** (*string*) – related_case
- **from_serviceplatformen** (*string*) – from_serviceplatformen
- **main_case** (*string*) – main_case

Status Codes

- 204 No Content –

GET /api/municipalities/

Expose municipalities in REST API.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name

Example request:

```
GET /api/municipalities/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "active": true,
    "name": "string"
  }
]
```

GET /api/municipalities/{id}/
Expose municipalities in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this kommune.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name

Example request:

```
GET /api/municipalities/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "name": "string"
}
```

GET /api/school_districts/
Expose school districts in REST API.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name

Example request:


```
GET /api/school_districts/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "active": true,
    "name": "string"
  }
]
```

GET /api/school_districts/{id}/
Expose school districts in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this distrikt.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name

Example request:

```
GET /api/school_districts/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "name": "string"
}
```

GET /api/teams/
Expose teams in REST API.

Example request:

```
GET /api/teams/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "name": "string",
    "leader": 1
  }
]
```

GET /api/teams/{id}/
Expose teams in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this team.

Example request:

```
GET /api/teams/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "name": "string",
  "leader": 1
}
```

GET /api/sections/
Expose law sections in REST API.

Query Parameters

- **active** (*string*) – active
- **paragraph** (*string*) – paragraph
- **text** (*string*) – text
- **allowed_for_target_groups** (*string*) – allowed_for_target_groups
- **allowed_for_steps** (*string*) – allowed_for_steps
- **law_text_name** (*string*) – law_text_name

Example request:

```
GET /api/sections/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "active": true,
    "paragraph": "string",
    "text": "string",
    "law_text_name": "string",
    "allowed_for_target_groups": [
      1
    ],
    "allowed_for_steps": [
      1
    ]
  }
]
```

GET /api/sections/{id}/
Expose law sections in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this paragraph.

Query Parameters

- **active** (*string*) – active
- **paragraph** (*string*) – paragraph
- **text** (*string*) – text
- **allowed_for_target_groups** (*string*) – allowed_for_target_groups
- **allowed_for_steps** (*string*) – allowed_for_steps
- **law_text_name** (*string*) – law_text_name

Example request:

```
GET /api/sections/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "paragraph": "string",
  "text": "string",
  "law_text_name": "string",
  "allowed_for_target_groups": [
```

(continues on next page)

(continued from previous page)

```

        1
    ],
    "allowed_for_steps": [
        1
    ]
}

```

GET /api/sectioninfos/
Expose section infos in REST API.

Query Parameters

- **activity_details** (*string*) – activity_details
- **section** (*string*) – section
- **activity_category** (*string*) – activity_category
- **kle_number** (*string*) – kle_number
- **sbsys_template_id** (*string*) – sbsys_template_id
- **main_activity_main_account_number** (*string*) – main_activity_main_account_number
- **supplementary_activity_main_account_number** (*string*) – supplementary_activity_main_account_number

Example request:

```

GET /api/sectioninfos/ HTTP/1.1
Host: example.com

```

Status Codes

- **200 OK** – Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "kle_number": "string",
    "sbsys_template_id": "string",
    "main_activity_main_account_number": "string",
    "supplementary_activity_main_account_number": "string",
    "activity_details": 1,
    "section": 1,
    "activity_category": 1
  }
]

```

GET /api/sectioninfos/{id}/
Expose section infos in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this paragraph-info.

Query Parameters

- **activity_details** (*string*) – activity_details
- **section** (*string*) – section
- **activity_category** (*string*) – activity_category
- **kle_number** (*string*) – kle_number
- **sbsys_template_id** (*string*) – sbsys_template_id
- **main_activity_main_account_number** (*string*) – main_activity_main_account_number
- **supplementary_activity_main_account_number** (*string*) – supplementary_activity_main_account_number

Example request:

```
GET /api/sectioninfos/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "kle_number": "string",
  "sbsys_template_id": "string",
  "main_activity_main_account_number": "string",
  "supplementary_activity_main_account_number": "string",
  "activity_details": 1,
  "section": 1,
  "activity_category": 1
}
```

GET /api/activity_details/

Expose activity details in REST API.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name
- **description** (*string*) – description
- **activity_id** (*string*) – activity_id
- **max_tolerance_in_percent** (*string*) – max_tolerance_in_percent
- **max_tolerance_in_dkk** (*string*) – max_tolerance_in_dkk
- **main_activity_for** (*string*) – main_activity_for
- **supplementary_activity_for** (*string*) – supplementary_activity_for
- **service_providers** (*string*) – service_providers
- **main_activities** (*string*) – main_activities

Example request:

```
GET /api/activity_details/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "active": true,
    "name": "string",
    "description": "string",
    "activity_id": "string",
    "max_tolerance_in_percent": 1,
    "max_tolerance_in_dkk": 1,
    "main_activity_for": [
      "string"
    ],
    "supplementary_activity_for": [
      1
    ],
    "service_providers": [
      1
    ],
    "main_activities": [
      1
    ]
  }
]
```

GET /api/activity_details/{id}/
Expose activity details in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this aktivitetsdetalje.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name
- **description** (*string*) – description
- **activity_id** (*string*) – activity_id
- **max_tolerance_in_percent** (*string*) – max_tolerance_in_percent
- **max_tolerance_in_dkk** (*string*) – max_tolerance_in_dkk
- **main_activity_for** (*string*) – main_activity_for
- **supplementary_activity_for** (*string*) – supplementary_activity_for
- **service_providers** (*string*) – service_providers

- **main_activities** (*string*) – main_activities

Example request:

```
GET /api/activity_details/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "name": "string",
  "description": "string",
  "activity_id": "string",
  "max_tolerance_in_percent": 1,
  "max_tolerance_in_dkk": 1,
  "main_activity_for": [
    "string"
  ],
  "supplementary_activity_for": [
    1
  ],
  "service_providers": [
    1
  ],
  "main_activities": [
    1
  ]
}
```

GET /api/service_providers/

Expose service providers in REST API.

Query Parameters

- **active** (*string*) – active
- **cvr_number** (*string*) – cvr_number
- **name** (*string*) – name
- **vat_factor** (*string*) – vat_factor

Example request:

```
GET /api/service_providers/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "active": true,
    "cvr_number": "string",
    "name": "string",
    "vat_factor": "string"
  }
]
```

GET /api/service_providers/{id}/

Expose service providers in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this leverandør.

Query Parameters

- **active** (*string*) – active
- **cvr_number** (*string*) – cvr_number
- **name** (*string*) – name
- **vat_factor** (*string*) – vat_factor

Example request:

```
GET /api/service_providers/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK** – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "cvr_number": "string",
  "name": "string",
  "vat_factor": "string"
}
```

GET /api/approval_levels/

Expose approval levels in REST API.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name

Example request:


```
GET /api/approval_levels/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "active": true,
    "name": "string"
  }
]
```

GET /api/approval_levels/{id}/
Expose approval levels in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this bevillingsniveau.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name

Example request:

```
GET /api/approval_levels/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "name": "string"
}
```

GET /api/users/
Expose users in REST API.

Example request:

```
GET /api/users/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "username": "string",
    "first_name": "string",
    "last_name": "string",
    "cases": [
      1
    ],
    "team": 1,
    "profile": "readonly"
  }
]
```

GET /api/users/{id}/
Expose users in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this bruger.

Example request:

```
GET /api/users/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "username": "string",
  "first_name": "string",
  "last_name": "string",
  "cases": [
    1
  ],
  "team": 1,
  "profile": "readonly"
}
```

GET /api/effort_steps/
Expose effort steps in REST API.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name
- **number** (*string*) – number

Example request:

```
GET /api/effort_steps/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "active": true,
    "name": "string",
    "number": 1
  }
]
```

GET /api/effort_steps/{id}/
Expose effort steps in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this indsatrappetrin.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name
- **number** (*string*) – number

Example request:

```
GET /api/effort_steps/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "name": "string",
  "number": 1
}
```

GET /api/target_groups/
Expose target groups in REST API.

Example request:

```
GET /api/target_groups/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "active": true,
    "name": "string",
    "required_fields_for_case": "string"
  }
]
```

GET /api/target_groups/{id}/
Expose target groups in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this målgruppe.

Example request:

```
GET /api/target_groups/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "name": "string",
  "required_fields_for_case": "string"
}
```

GET /api/internal_payment_recipients/
Expose internal payment recipients in REST API.

Example request:

```
GET /api/internal_payment_recipients/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "id": 1,
    "name": "string"
  }
]
```

GET /api/internal_payment_recipients/{id}/
Expose internal payment recipients in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this intern betalingsmodtager.

Example request:

```
GET /api/internal_payment_recipients/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "name": "string"
}
```

GET /api/efforts/
Expose efforts in REST API.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name
- **description** (*string*) – description
- **allowed_for_target_groups** (*string*) – allowed_for_target_groups

Example request:

```
GET /api/efforts/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
[
  {
    "id": 1,
    "active": true,
    "name": "string",
    "description": "string",
    "allowed_for_target_groups": [
      1
    ]
  }
]
```

GET /api/efforts/{id}/
Expose efforts in REST API.

Parameters

- **id** (*string*) – A unique integer value identifying this indsats.

Query Parameters

- **active** (*string*) – active
- **name** (*string*) – name
- **description** (*string*) – description
- **allowed_for_target_groups** (*string*) – allowed_for_target_groups

Example request:

```
GET /api/efforts/{id}/ HTTP/1.1
Host: example.com
```

Status Codes

- **200 OK** – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "active": true,
  "name": "string",
  "description": "string",
  "allowed_for_target_groups": [
    1
  ]
}
```

GET /api/editing_past_payments_allowed/
Return the Django setting allowing changes to the past.

Example request:

```
GET /api/editing_past_payments_allowed/ HTTP/1.1
Host: example.com
```

Status Codes

- 200 OK – Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {}
]
```

POST /api/token/

Takes a set of user credentials and returns an access and refresh JSON web token pair to prove the authentication of those credentials.

Example request:

```
POST /api/token/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "username": "string",
  "password": "string"
}
```

Status Codes

- 201 Created – Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "username": "string",
  "password": "string"
}
```

POST /api/token/refresh/

Takes a refresh type JSON web token and returns an access type JSON web token if the refresh token is valid.

Example request:

```
POST /api/token/refresh/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "refresh": "string"
}
```

Status Codes

- 201 Created – Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "refresh": "string"
}
```

PATCH /api/cases/change_case_worker/

Change the case_worker of several Cases.

Parameters

- **case_pks** – A list of case pks.
- **case_worker_pk** – the case worker pk to change to.

Example request:

```
PATCH /api/cases/change_case_worker/ HTTP/1.1
Host: example.com
Content-Type: application/json
```

```
{
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "cpr_number": "string",
  "name": "string",
  "scaling_step": 1,
  "assessment_comment": "string",
  "note": "string",
  "case_worker": 1,
  "district": 1,
  "paying_municipality": 1,
  "acting_municipality": 1,
  "residence_municipality": 1,
  "target_group": 1,
  "effort_step": 1,
  "efforts": [
    1
  ]
}
```

Status Codes

- **200 OK – Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "expired": "string",
  "num_ongoing_appropriations": "string",
  "num_ongoing_draft_or_expected_appropriations": "string",
  "team": "string",
  "created": "2021-04-19T09:16:55.367826",
  "modified": "2021-04-19T09:16:55.367826",
  "user_created": "string",

```

(continues on next page)

(continued from previous page)

```

    "user_modified": "string",
    "sbsys_id": "string",
    "cpr_number": "string",
    "name": "string",
    "scaling_step": 1,
    "assessment_comment": "string",
    "note": "string",
    "case_worker": 1,
    "district": 1,
    "paying_municipality": 1,
    "acting_municipality": 1,
    "residence_municipality": 1,
    "target_group": 1,
    "effort_step": 1,
    "efforts": [
        1
    ]
}

```

PATCH /api/appropriations/{id}/grant/
Grant the specified activities on this appropriation.

Parameters

- **id**(string) –

Example request:

```

PATCH /api/appropriations/{id}/grant/ HTTP/1.1
Host: example.com
Content-Type: application/json

{
  "user_created": "string",
  "user_modified": "string",
  "sbsys_id": "string",
  "note": "string",
  "section": 1,
  "case": 1
}

```

Status Codes

- **200 OK – Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": 1,
  "status": "string",
  "granted_from_date": "string",
  "granted_to_date": "string",
  "case_cpr_number": "string",
  "case_name": "string",
  "case_sbsys_id": "string",
  "num_ongoing_draft_or_expected_activities": "string",

```

(continues on next page)

(continued from previous page)

```

    "num_ongoing_activities": "string",
    "main_activity": {
      "id": 1,
      "monthly_payment_plan": "string",
      "total_cost": "string",
      "total_cost_this_year": "string",
      "total_cost_full_year": "string",
      "total_granted_this_year": "string",
      "total_expected_this_year": "string",
      "payment_plan": {
        "id": 1,
        "payments": [
          {
            "id": 1,
            "account_string": "string",
            "account_alias": "string",
            "payment_schedule_payment_id": "string",
            "case_cpr_number": "string",
            "case_name": "string",
            "activity_id": "string",
            "activity_status": "string",
            "activity_details_id": "string",
            "payment_schedule_fictive": "string",
            "is_payable_manually": "string",
            "date": "2021-04-19",
            "recipient_type": "INTERNAL",
            "recipient_id": "string",
            "recipient_name": "string",
            "payment_method": "CASH",
            "amount": "string",
            "paid_amount": "string",
            "paid": true,
            "paid_date": "2021-04-19",
            "note": "string",
            "payment_schedule": 1
          }
        ],
        "price_per_unit": {
          "id": 1,
          "rates_per_date": [
            {
              "rate": "string",
              "start_date": "2021-04-19",
              "end_date": "2021-04-19",
              "changed_date": "2021-04-19T09:16:55.367826",
              "changed_by": 1
            }
          ],
          "current_amount": "string",
          "amount": "string",
          "start_date": "2021-04-19",
          "end_date": "2021-04-19"
        },
        "recipient_type": "INTERNAL",
        "recipient_id": "string",
        "recipient_name": "string",

```

(continues on next page)

(continued from previous page)

```

        "payment_method": "CASH",
        "payment_frequency": "DAILY",
        "payment_date": "2021-04-19",
        "payment_day_of_month": 1,
        "payment_type": "ONE_TIME_PAYMENT",
        "payment_units": "string",
        "payment_amount": "string",
        "fictive": true,
        "payment_id": 1,
        "payment_cost_type": "FIXED",
        "payment_method_details": 1,
        "payment_rate": "string"
    },
    "details__name": "string",
    "created": "2021-04-19T09:16:55.367826",
    "modified": "2021-04-19T09:16:55.367826",
    "user_created": "string",
    "user_modified": "string",
    "status": "DRAFT",
    "approval_note": "string",
    "appropriation_date": "2021-04-19",
    "start_date": "2021-04-19",
    "end_date": "2021-04-19",
    "activity_type": "MAIN_ACTIVITY",
    "note": "string",
    "details": 1,
    "approval_level": 1,
    "approval_user": 1,
    "modifies": 1,
    "appropriation": 1,
    "service_provider": 1
},
"activities": "string",
"total_granted_this_year": "string",
"total_granted_full_year": "string",
"total_expected_this_year": "string",
"total_expected_full_year": "string",
"total_cost_expected": "string",
"total_cost_granted": "string",
"created": "2021-04-19T09:16:55.367826",
"modified": "2021-04-19T09:16:55.367826",
"user_created": "string",
"user_modified": "string",
"sbsys_id": "string",
"note": "string",
"section": 1,
"case": 1
}

```

Frontend documentation

The frontend GUI is a javascript single page application using the [VueJS framework](#). It comprises the following components:

- [VueJS](#) for templating and UI logic (view layer)
- [Vue Router](#) for routing between views
- [Vuex](#) for sharing data between Vue components
- [axios](#) for communicating with the backend REST API
- A modified version of [Semstrap](#) for basic CSS styling
- [Vue CLI](#) for build process management
- [Testcafe](#) for E2E testing

Getting started

If you have [docker](#) installed and followed the [installation instructions](#) , you already have the frontend up and running.

Open your browser and point to `localhost:8080` to see the GUI in action. You should be able to log in with username **admin** and password **admin**.

In this configuration, the frontend lives inside a docker container running the Vue CLI server. When you edit files located in `/frontend/src`, Vue CLI server uses hot reload to immediately reload the changes in your browser.

Note: This setup is only suitable for development use. You should make a proper production build for any public use.

Building

In development mode, the frontend is hosted inside a docker container running Vue CLI. **In production mode**, you don't need this since you just need to host the bundled javascript and CSS files.

To access the docker container, run `docker exec -it bevillingsplatform_frontend_1 bash` from the project root on your host machine. Now you can run various commands using [npm](#)

Building for development

Run `npm run serve` inside the frontend docker container to start Vue CLI. You rarely need to do this since Vue CLI starts as soon as the container is initiated.

Building for production

Run `npm run build` inside the frontend docker container to create a production build. If the build is successful, you'll find the finished bundle in the `/frontend/dist` folder.

E2E testing

The testing tools have been isolated from the frontend proper. This way, you don't need to enter a docker container to run tests while developing. The files and tools for E2E testing are located in `/frontend-test`

To run a test, make sure you have [NodeJS](#) and [npm](#) installed on your host machine. Then change to the `/frontend-test` folder and run `npm run test` from a command line.

Default browser for testing

The test will run inside a [Chrome browser](#) so we assume you have the Chrome browser installed. If you want to run the test in another browser, open `/frontend-test/package.json` and edit the line for the `test` script to use another browser like `firefox` or `safari`.

Testing on external devices

You can test your local development instance in virtual machines and on other devices. This is useful when you need to do UI tests on mobile devices or in browsers that can't be installed on your system (like old versions of Internet Explorer). Unfortunately, this doesn't come out of the box. You'll need to set it up.

Pointing your browser to `localhost:8080` will show you the UI but this will only work on your host machine. In your virtual guest OS or mobile device, `localhost` points to something else. The key is to use your host's IP address rather than `localhost`. Devices *that are on your local network* are able to detect your host system IP and display the UI. Virtual machines will generally also be able to pinpoint your development instance by IP.

Set it up

First, find your host system IP address on your local network. There are various ways to do this depending on your system. Your IP could be something like `10.0.0.23`

Open `/docker-compose.yml` and find the following bit of text:

```
environment:
  - SIMPLESAMLPHP_BASEURLPATH=http://localhost:8080/simplesaml/
  - SIMPLESAMLPHP_SP_ENTITY_ID=http://localhost:8080
  - SIMPLESAMLPHP_SP_ASSERTION_CONSUMER_SERVICE=http://localhost:8080/api/saml2_auth/
  ↪ acs/
```

Replace all mentions of `localhost` with your IP like this:

```
environment:
  - SIMPLESAMLPHP_BASEURLPATH=http://10.0.0.23:8080/simplesaml/
  - SIMPLESAMLPHP_SP_ENTITY_ID=http://10.0.0.23:8080
  - SIMPLESAMLPHP_SP_ASSERTION_CONSUMER_SERVICE=http://10.0.0.23:8080/api/saml2_auth/
  ↪ acs/
```

Open `/dev-environment/dev-settings.ini` and find the following bit of text:

```
ALLOWED_HOSTS=localhost,bev
```

Add your IP to the line like this:

```
ALLOWED_HOSTS=localhost,bev,10.0.0.23
```

In the command line, kill and reboot your development instance to use the new configuration. First with `CTRL+C` and then:

```
docker-compose down -v
docker-compose up
```

You should now be able to point a browser in your VM or mobile device to `10.0.0.23:8080` and see the frontend working.

Tutorial: Change the logo

This tutorial will give you a simple introduction to customizing the frontend for your own purposes.

Say, you want to change the logo image that rests in the top left corner of every page. You'll need to do the following:

Add a new image to the frontend source files # Identify the component that displays the logo # Edit the corresponding template and maybe add some styles # Build the frontend

Let's go through these steps in greater detail.

Add a new image

Static resources like images, icons, and multimedia are found in the `/frontend/public` folder.

If you look inside the folder, you'll find a `logo.png` file in there already. Now you could easily swap it for a different image with the same filename but that's no fun. Instead, get a new `my-logo.png` image and drop it into `/frontend/public`.

Find the component that displays the logo

You probably saw an `index.html` file while looking into the `/frontend/public` folder. It doesn't contain much of interest since all of the references to javascript and css files are created in the build step. But if you want to add external stylesheets or a new favicon, this `index.html` is the place to do it.

The real heart of our single page application is in `/frontend/src/App.vue`. All the dynamically generated views are piped into `App.vue`. If you open the file, you'll see a `<template>` section, a `<script>` section, and a `<style>` section. These are typical sections in a Vue component file.

Notice the line with `<router-view v-if="auth" />` in the `<template>` section. The `router-view` element is the entrypoint for Vue Router to pipe in content. When you navigate between views in the single page application, it's really just the contents of `router-view` that are being swapped. Looking into `/frontend/src/router.js` will tell you what components are displayed for each route. This is a good starting point for understanding what components are being displayed at any given time.

But since the logo is present on every page, it is not displayed using `router-view`. The logo resides in the header so you must look into `<app-header />`. If you read about [vue components](#), you'll know that this element corresponds to the `import` statement in the `<script>` section that references `/frontend/src/components/header/Header.vue`. Opening `Header.vue` will show you where the logo image is integrated.

Edit template

In *Header.vue* you'll find that the logo is found in this line ``. Change the *src* attribute of the *img* element to `"/my-logo.png"`. Notice that the image src URL considers `/frontend/public` folder to be the root folder for static files. So the proper way to point to `/frontend/public/my-logo.png` is to enter `/my-logo.png`.

Maybe your new logo has some different dimensions so you might want to change the logo's styles. The *img* element has class *global-logo*. Look in the `<styles>` section of *Header.vue* to find the CSS rules for `.global-logo` and change it accordingly. Styles that only apply to a certain component are usually added directly to Vue components like this. Global styles are added in *App.vue* or in some of the CSS files in `/frontend/src/assets/css`

Build the frontend

If you are running in development mode, your changes should display in the browser right after you save your changes. For a production build, you'd run the `npm run build` command and host the built files from `/frontend/dist` somewhere. Now you've customized the frontend :)

Code documentation

municipalityId2name (*id*)

Convert municipality id to human readable name

Arguments

- **id** (*number*) – municipality ID

Returns **string** – municipality name

districtId2name (*id*)

Convert district id to human readable name

Arguments

- **id** (*number*) – district ID

Returns **string** – district name

activityId2name (*id*)

Convert activity id to human readable name

Arguments

- **id** (*number*) – activity ID

Returns **string** – activity name

sectionId2name (*id*)

Convert section id to human readable name

Arguments

- **id** (*number*) – section ID

Returns **string** – section name

userId2name (*id*)

Convert user id to human readable name

Arguments

- **id** (*number*) – user ID

Returns **string** – user's name

teamId2name (*id*)

Convert team id to human readable name

Arguments

- **id** (*number*) – team ID

Returns **string** – team's name

displayStatus (*id*)

Return a human readable status

Arguments

- **id** (*string*) – system status name

Returns **string** – Readable status in Danish

displayEffort (*id*)

Convert effort step id to human readable name

Arguments

- **id** (*number*) – effort step ID

Returns **string** – effort step name

testSpin ()

Test if a spinner is already in the GUI

Returns **boolean** – True if a spinner element is in the DOM

spinOn ()

Start a GUI spinner

spinOff ()

Stop a GUI spinner

getCases ()

Get list of cases currently in the store.

Returns **array** – List of cases

Examples:

```
this.$store.getters.getCases()
```

getCase ()

Get case info currently in the store.

Returns **object** – A single case object

Examples:

```
this.$store.getters.getCase()
```

fetchCases (*state*)

Get a list of cases from API. Use getCases to read the list.

Arguments

- **state** (*object*) – OPTIONAL An object containing keys and values for a query string.

Examples:

```
this.$store.dispatch('fetchCases') or this.$store.dispatch('fetchCases', {
  queryKey: 'queryValue'})
```

fetchCase (*cas_id*)

Get a single case from API. Use `getCase` to read the list.

Arguments

- **cas_id** (*number*) – Case ID

Returns **void** – Use `store.getter.getCase` to read the info.

Examples:

```
this.$store.dispatch('fetchCase', 42)
```

1.4 Release History

1.4.1 Version 3.4.3, 2021-03-23

New in this version:

Features

- Restrict editing activity type for expected activities modifying another.
- Update OS2Forms.xml with new KLE, OS2FormsID and SbsysCaseFileNumber.
- Add Payment note to payments report.
- Update thirdparty dependencies.

1.4.2 Version 3.4.2, 2021-03-12

Hotfix release

Bug fixes

- Correctly use activity-component of `account_alias_new`.

1.4.3 Version 3.4.1, 2021-03-11

New in this version:

Features

- Persist the new account_string and account_alias when paying payments.
- Fix the calculation of payment sums by excluding deleted activities.
- Consolidate prometheus logging to a single setting.
- Update third party dependencies.

1.4.4 Version 3.4.0, 2021-02-11

New in this version:

Features

- Add parallel account_string using the new activity category models.
- Add parallel account_alias using the new account alias mapping models.
- Update payment list when a payment is changed (for example paid).
- Publish database documentation on build.
- Update third party dependencies.

1.4.5 Version 3.3.1, 2021-01-25

Hotfix release

Bug fixes

- Run PRISM export also on Fridays

1.4.6 Version 3.3.0, 2020-12-17

New in this version:

Features

- Add redirect after SSO login.
- Save filters across page transitions.
- Save filters in URL to make them bookmarkable.
- Change “reset” functionality of overview pages.
- Remove unneeded Team on Case and instead display and filter on Team of the case worker.
- Validate CVR number for recipient_id on PaymentSchedule when recipient is “Firma”.
- Add generic time intervals for payments (previous, current, next - week, month, year).
- Added fields to payments report.
- Changed the flow of emails when manipulating activities.

- Add child name to CPR number of Payments overview.
- Add Users and Teams to workflow users Admin page.
- Update third party dependencies.

Bug fixes

- Fix a bug where updating a payment caused a PATCH twice.
- Various fixes to frontend tests.

1.4.7 Version 3.2.5, 2020-12-02

Hotfix release

Bug fixes

- Use correct date limits for supplementary activity creation.

1.4.8 Version 3.2.4, 2020-11-24

New in this version:

Bug fixes

- Remove upper time limit for generated payment reports.
- Fix calculation of earliest start date when creating an activity.
- Remove redundant PRISM file generation so only one is generated.
- Don't automatically mark fictive invoice payments as paid.
- Add a warning on the supplementary activities when shortening a main activity.
- Update various dependencies.

1.4.9 Version 3.2.3, 2020-11-16

Hotfix release

Bug fixes

- Correctly initialize SAML in settings.py
- Update SAML dependencies to allow POST SingleSignOnService binding

1.4.10 Version 3.2.2, 2020-10-22

New in this version:

Bug fixes

- When generating payments report, catch exception if payment date is before case was created.

1.4.11 Version 3.2.1, 2020-10-13

New in this version:

Bug fixes

- Allow granting activities with no payments.
- Fix generating payments for activities that started with no payments.
- Disallow editing payments for a payment plan that is not individual.
- Update various dependencies.

1.4.12 Version 3.2.0, 2020-09-30

New in this version:

Bug fixes

- Don't send payment emails to internal recipients.
- Allow payments of 0 kroner.
- One time payment activities should not have main activity end date set when granting.
- Proper data cleanup in GUI when user changes payment method or type.
- Allow display of Prices with no start date.
- Don't allow individual payments outside of main activity's period.
- Layout/hyphenation issue fixed.
- Date dependent price per unit must be valid at least from activity's start date.

Features

- Individual payment plans.
- Stop using Postgres-specific DB field for "required fields for target group".
- Python packages are upgraded to include the latest security fixes.
- Warn users that future changes to SD and Cash payments will be overwritten.
- Allow relevant users to edit payments of type Cash and SD Løn.
- Only allow creation of new payments for individual payment plan and only on drafts or expectations.

- UX for Activity creation updated - user goes to the activity in read only mode after creating, not to the appropriation.
- Account alias and account string are shown in the same column.
- Delete button is not shown for granted payments and deleting granted payments is prevented by the backend.
- Appropriation PDF is updated to include individual payments.
- Audit information, responsible user and time stamp is added to rates and prices.
- Price history in fronted is updated to include audit fields.
- Payment per kilometer etc. are cleaned up and replaced by “running payment” in the database.
- Handle expectations for activities with individual payment plan.
- Update recipient on future payments when saving drafts and expectations.
- Backend restrictions on editing payments: Admin and workflow users can edit SD or Cash - other users can only mark non-paid payments (of the other types) as paid. Nobody can edit the amount of granted payments.
- CSV export files are modified to support the changes introduced in Phase 3.
- PRISME export will output to files, one with account alias, one without.
- Don’t allow granting an activity with no payments.
- Don’t allow user to add new payments if activity is in edit mode.

1.4.13 Version 3.1.1, 2020-08-31

New in this version:

Bug fixes

- Fix instance of prices being displayed with non-Danish decimal separator.
- Approximate payment calculator now uses current rate if rates are used.
- Enable input field validation in browser when creating activities.
- Updates list of service providers in UI based on current activity detail.
- Include global rate and price per unit information in payment email and PDF.

1.4.14 Version 3.1.0, 2020-07-09

New in this version:

Features

- Add Prices and Rates.
- Add counts for draft, expected and ongoing activities.
- Don't send activity emails for one time payments.
- Prism payments account for holidays and weekends with PaymentDateExclusions.
- Main account refactoring, use new account string and remove old Account model.
- Emphasize new activities in appropriation email.
- Add pagination for Appropriation PDF.
- Add notes for Activity.
- Make Appropriation drafts deleteable.
- Display payment method when recipient is internal or company.
- Add child name, and case worker fields to activity emails.
- Set creation and modification user correctly for Case, Appropriation, Activity, RelatedPerson.
- Remove Service Providers from ActivityDetails Admin.
- Numerous styling fixes.
- Add labels to Docker files.
- Update dependencies to new versions.

1.4.15 Version 3.0.0, 2020-06-03

New in this version:

Features

- New Django Admin user interface and permission profile for handling classifications.
- Frontend overviews have been improved.
- Account number have been refactored.
- Classifications can be marked active on/off.
- Efforts are now a classification.
- Target groups are now a classification.
- ActivityDetails now have a description.
- Related persons are now editable and can be marked 'manual' or 'from Serviceplatformen'.
- Allow hiding expired activities in the frontend.
- Improved search for payments.
- Fixed dropdown menus with only one choice.
- Frontend and Appropriation endpoint performance improvements.
- Many smaller fixes to texts.

- Update dependencies to new versions.

1.4.16 Version 2.6.1, 2020-04-03

Hotfix release

Bug fixes

- Use correct dates for PRISM exports for Sunday and Monday.

1.4.17 Version 2.6.0, 2020-03-31

New in this version:

Features

- Delete payment schedules and payments when an activity is deleted.
- Send an email when an activity is expired.
- Change subject on activity deleted email.
- Change text string in frontend for closed cases.
- Add status in payments report.
- Change prism payment for Saturday, Sunday and Monday to be exported Friday.
- Add coverage and tests for management commands.
- Update dependencies to new versions.

Bug fixes

- Remove validation for monthly expected adjustments.

1.4.18 Version 2.5.0, 2020-03-06

New in this version:

Features

- Add section, section_text, payment_schedule__payment_id and main_activity_name to CSV Payments report.
- Return a validation error when trying to create an invalid monthly payment schedule.
- Use create_rrule for all the places we check generated payments.
- Add tests for the frontend.
- Update README with logging documentation.
- Update documentation for generating database documentation.
- Add shell linting and docker file linting and lint fixes.

- Add automatic deployment for develop branch.
- Allow the docker backend service to be debuggable with docker attach.
- Update dependencies to new versions.

1.4.19 Version 2.4.2, 2020-02-24

Hotfix release

Bug fixes

- Fix fonts urlpattern for loading fonts as assets.

1.4.20 Version 2.4.1, 2020-02-24

New in this version:

Features

- Store google fonts as assets instead of fetching them from google servers.

Bug fixes

- Fix duplicate payments generation.
- Remove duplicate payments in a migration.
- Add database constraint which prevents duplicate payments on date.

1.4.21 Version 2.4.0, 2020-01-24

New in this version:

Features

- Add warning in GUI if a payment date is earlier than two days from today.
- Add restriction in GUI so an Activity can only have one expected Activity.
- Add Actual-state CSV generation for Payments.
- Improve documentation all-around.
- Change payment file default date to tomorrow.
- Update Django from 2.2.4 to 2.2.9

Bug fixes

- Fix CPR search for “Find sager”.
- Change field 17 of PRISM file to the unique Payment pk.
- Handle missing effort steps gracefully in GUI.

1.4.22 Version 2.3.0, 2020-01-09

New in this version:

Features

- Modify the URL for the rate tabel (taksttabel) to a more general one.
- Nice-ify django admin for Payments and PaymentSchedules and allow search on payment id.
- Add pydocstyle compliance.
- Add sphinx docs generation.

Bug fixes

- Fix incorrect tests dependent on current year.

1.4.23 Version 2.2.3, 2019-12-12

New in this version:

- Changes to PRISM file generation.
- Enforce rules for activities on grant.
- Disable edit for appropriation fields on granted activities.
- Enable date validation for activities.
- Add filtering on payment type.
- Fix payment CPR filtering.
- Small improvements to logging.
- Make tox work locally.
- Add frontend documentation.
- Add cronjobs for docker.
- Fix date filtering.
- Fix failing tests.
- Fix paths in settings.
- Update Django from 2.2.1 to 2.2.4

1.4.24 Version 2.2.2, 2019-11-28

Hotfix

- Fix broken migration.

1.4.25 Version 2.2.1, 2019-11-25

New in this version:

Features

- Mark payments for SD Løn along with fictive ones.

Bug fixes

- Fix hover text.
- Display of Indsatstrappen fixed.
- Fix ordering of Indsatstrappen.
- Recipient info stayed in GUI even though payment method was changed to “internal”.
- Empty “not found” text when displaying “Mine sager”.

1.4.26 Version 2.2.0, 2019-11-21

New in this version:

Features

- It is now possible to find payments from a payment ID.
- Case worker can now be changed on several cases in one action.
- A log of all pending and sent emails is now kept and accessible in the Django admin interface.
- Payment ID and account string is displayed in the Django admin interface.
- Generally improved interface for searching and displaying cases.
- Fictive payments are clearly marked as fictive in payment plans.
- Fictive payments are marked as paid in the database on the day they are due.
- Field added in API to indicate whether a payment can be paid manually or not.
- Payments that are paid as Salary (through SD-Løn) or cash or are fictive may not be edited manually.
- Payments are paginated to avoid too long loading times.
- Payments are now sorted by payment date.
- Payments are sorted by *ascending* payment date.
- Indsatstrappen is now a classification to be maintained in the Django Admin interface.

- Section (of the law, from the appropriation) is added to the payment emails.
- Emails are sent for all approved payments, for all combinations of payment and recipient types.
- Complex logic for generation of account string.
- Payment dialog improved.
- Integration to KMD PRISME accounting system.
- Information about citizen included in display of appropriation.

Bug fixes

- Don't throw an exception if users attempt to access the API without logging in, just deny access.
- If more than one user profile is sent from SAML IdP, don't crash - choose the *highest* one.
- "Mixed content error" on some pages (on internal test server).
- Many small and big improvements to styling and usability.
- Function deciding if case is expired also looked at DELETED activities.

1.4.27 Version 2.0.1, 2019-11-11

New in this version:

- Add support for Service Provider certificates through PySaml2.

1.4.28 Version 2.0.0, 2019-11-06

New in this version:

- Implement SAML SSO login.
- Implement user rights levels.
- Add preliminary Prism file generation.
- Implement GUI for editing payments.
- Add support for "fictive" payments.
- Add support for negative and zero payments.
- Add support for paid amounts and paid date for payments.
- Update payment summation to include paid amounts when able.
- Add new payment ID for payment plans.
- Add account strings for payments.
- Add API filtering for several endpoints.
- Remove the "udbetaling til firma" payment option.
- Fix a bug when creating an activity.
- Fix redirect when setting a payment paid.
- Add missing verbose names in Django admin.

1.4.29 Version 1.1.1, 2019-10-30

Hotfix release.

New in this version:

- Deleted main activity no longer blocks for creating a new main activity.
- Granted activities are now explicitly included in the appropriation PDF.
- Fix activities still being checked for granting when closing the grant dialog.
- Fix not being able to grant an expected main activity.
- Fix invalid XML in OS2forms.xml.
- Add missing constraint for creating supplementary activities based on allowed main activities.

1.4.30 Version 1.1.0, 2019-10-04

New in this version:

- Fixed approval button when there's nothing to approve.
- Fixed missing activities from appropriation PDF.
- Fix spelling error in logout message.
- For payment to a person with SDLøn, tax card is mandatory.
- Use user first_name and last_name instead of initials for Sagsbehandler dropdown.
- Fix stop dates on supplementary activities.
- Fix link to rates document.
- Correctly calculate the expected amount for expected activities.
- Correct forms for modifying effort steps (Indsatstrappe) in Djang Admin.
- Clear frontend errors correctly.
- Rearrange autologin scripts in frontend.
- Change recommended browser text.
- Suppress not writeable warning from ipython.

1.4.31 Version 1.0.0, 2019-09-27

First production release. New in this version:

- KLE number and SBYS template info moved from Section to new SectionInfo class in the ManyToMany relation.
- Activities are granted individually, not all at once for each appropriation.
- Missing logo fixed/supplied.
- Various GUI and UX improvements.
- Prevent expected changes from starting in the past.
- Make user supply day of month for monthly payments - handle month end correctly.
- Browser compatibility fixes.

- Fix missing update of family relations.
- Improved handling of backend error messages.
- New API fields for expected and granted totals for activities.
- Appropriation PDF nicified and adapted to the new approval scheme.
- SBSYS integration (os2forms.xml) fixed.
- Cases *must* have a team, this field is now non-nullable.
- Activities with status EXPECTED are now soft-deleted.
- Status label for appropriations fixed.
- Wrong validation of KLE numbers fixed.
- Stop date of supplementary activities must be no later than stop date of main activity.
- End-to-end tests for accessibility added.
- Classifications updated, now production ready.
- Bad validation that expectation must be after next payment date removed.
- Allow units to be charged, e.g. dates, to be a decimal number.
- Unicorn is now run single-threaded.
- Updates to Docker configuration.
- It is now possible to make expectations for the entire appropriation period even though the main activity is split.
- DB representation of effort steps (Indsatstrappe) changed to integer.

1.4.32 Version 0.5.0, 2019-09-05

New in this version:

- initial release

PYTHON MODULE INDEX

b

- `bevillingsplatform`, [74](#)
- `bevillingsplatform.initialize`, [72](#)
- `bevillingsplatform.settings`, [73](#)
- `bevillingsplatform.urls`, [73](#)
- `bevillingsplatform.wsgi`, [74](#)

c

- `core`, [72](#)
- `core.admin`, [9](#)
- `core.apps`, [19](#)
- `core.authentication`, [19](#)
- `core.filters`, [19](#)
- `core.managers`, [22](#)
- `core.mixins`, [23](#)
- `core.models`, [24](#)
- `core.permissions`, [58](#)
- `core.serializers`, [58](#)
- `core.signals`, [61](#)
- `core.utils`, [62](#)
- `core.views`, [64](#)

m

- `manage`, [74](#)

HTTP ROUTING TABLE

/api

GET /api/activities/, 95	GET /api/sectioninfos/{id}/, 144
GET /api/activities/{id}/, 100	GET /api/sections/, 142
GET /api/activity_details/, 145	GET /api/sections/{id}/, 143
GET /api/activity_details/{id}/, 146	GET /api/service_providers/, 147
GET /api/appropriations/, 83	GET /api/service_providers/{id}/, 148
GET /api/appropriations/{id}/, 87	GET /api/target_groups/, 151
GET /api/approval_levels/, 148	GET /api/target_groups/{id}/, 152
GET /api/approval_levels/{id}/, 149	GET /api/teams/, 141
GET /api/cases/, 74	GET /api/teams/{id}/, 142
GET /api/cases/{id}/, 77	GET /api/users/, 149
GET /api/cases/{id}/history/, 82	GET /api/users/{id}/, 150
GET /api/editing_past_payments_allowed/, 154	POST /api/activities/, 98
GET /api/effort_steps/, 150	POST /api/appropriations/, 84
GET /api/effort_steps/{id}/, 151	POST /api/cases/, 76
GET /api/efforts/, 153	POST /api/payment_method_details/, 123
GET /api/efforts/{id}/, 154	POST /api/payment_schedules/, 111
GET /api/internal_payment_recipients/, 152	POST /api/payments/, 127
GET /api/internal_payment_recipients/{id}/, 153	POST /api/prices/, 120
GET /api/municipalities/, 139	POST /api/related_persons/, 134
GET /api/municipalities/{id}/, 140	POST /api/token/, 155
GET /api/payment_method_details/, 123	POST /api/token/refresh/, 155
GET /api/payment_method_details/{id}/, 124	PUT /api/activities/{id}/, 103
GET /api/payment_schedules/, 110	PUT /api/appropriations/{id}/, 89
GET /api/payment_schedules/{id}/, 113	PUT /api/cases/{id}/, 78
GET /api/payments/, 125	PUT /api/payment_method_details/{id}/, 124
GET /api/payments/{id}/, 128	PUT /api/payment_schedules/{id}/, 114
GET /api/prices/, 119	PUT /api/payments/{id}/, 129
GET /api/prices/{id}/, 121	PUT /api/prices/{id}/, 121
GET /api/rates/, 118	PUT /api/related_persons/{id}/, 137
GET /api/rates/{id}/, 119	DELETE /api/activities/{id}/, 109
GET /api/related_persons/, 134	DELETE /api/appropriations/{id}/, 95
GET /api/related_persons/fetch_from_serviceplatform/, 135	DELETE /api/cases/{id}/, 82
GET /api/related_persons/{id}/, 136	DELETE /api/payment_method_details/{id}/, 125
GET /api/school_districts/, 140	DELETE /api/payment_schedules/{id}/, 118
GET /api/school_districts/{id}/, 141	DELETE /api/payments/{id}/, 133
GET /api/sectioninfos/, 144	DELETE /api/prices/{id}/, 123
	DELETE /api/related_persons/{id}/, 139
	PATCH /api/activities/{id}/, 106
	PATCH /api/appropriations/{id}/, 92
	PATCH /api/appropriations/{id}/grant/,

157
PATCH /api/cases/change_case_worker/,
156
PATCH /api/cases/{id}/, 80
PATCH /api/payment_method_details/{id}/,
125
PATCH /api/payment_schedules/{id}/, 116
PATCH /api/payments/{id}/, 131
PATCH /api/prices/{id}/, 122
PATCH /api/related_persons/{id}/, 138

Symbols

`__init__()` (*core.admin.TargetGroupForm* method), 18
`__init__()` (*core.admin.VariableRateAdminForm* method), 18

A

`account_alias()` (*core.admin.ActivityAdmin* method), 10
`account_alias()` (*core.admin.PaymentAdmin* method), 14
`account_alias()` (*core.admin.PaymentScheduleAdmin* method), 14
`account_alias()` (*core.models.Activity* property), 26
`account_alias()` (*core.models.Payment* property), 42
`account_alias()` (*core.models.PaymentSchedule* property), 45
`account_alias_new()` (*core.admin.ActivityAdmin* method), 10
`account_alias_new()` (*core.admin.PaymentAdmin* method), 14
`account_alias_new()` (*core.admin.PaymentScheduleAdmin* method), 14
`account_alias_new()` (*core.models.Activity* property), 26
`account_alias_new()` (*core.models.Payment* property), 42
`account_alias_new()` (*core.models.PaymentSchedule* property), 45
`account_aliases` (*core.models.ActivityDetails* attribute), 29
`account_aliases` (*core.models.SectionInfo* attribute), 53
`account_number()` (*core.admin.ActivityAdmin* method), 10
`account_number()` (*core.models.Activity* property), 26

`account_number_new()` (*core.admin.ActivityAdmin* method), 10
`account_number_new()` (*core.models.Activity* property), 26
`account_string()` (*core.admin.PaymentAdmin* method), 14
`account_string()` (*core.admin.PaymentScheduleAdmin* method), 14
`account_string()` (*core.models.Payment* property), 42
`account_string()` (*core.models.PaymentSchedule* property), 46
`account_string_new()` (*core.admin.PaymentAdmin* method), 14
`account_string_new()` (*core.admin.PaymentScheduleAdmin* method), 15
`account_string_new()` (*core.models.Payment* property), 42
`account_string_new()` (*core.models.PaymentSchedule* property), 46
`AccountAlias` (class in *core.models*), 24
`AccountAlias.DoesNotExist`, 24
`AccountAlias.MultipleObjectsReturned`, 24
`AccountAliasAdmin` (class in *core.admin*), 9
`AccountAliasMapping` (class in *core.models*), 25
`AccountAliasMapping.DoesNotExist`, 25
`AccountAliasMapping.MultipleObjectsReturned`, 25
`AccountAliasMappingAdmin` (class in *core.admin*), 9
`AccountAliasMappingCSVFileUploadForm` (class in *core.admin*), 10
`acting_municipality` (*core.models.Case* attribute), 33
`acting_municipality_id` (*core.models.Case* attribute), 33
`active` (*core.models.Classification* attribute), 34
`activities` (*core.models.Appropriation* attribute), 30
`activities` (*core.models.ServiceProvider* attribute),

- 54
- Activity (class in core.models), 25
- activity (core.models.PaymentSchedule attribute), 46
- Activity.DoesNotExist, 26
- Activity.MultipleObjectsReturned, 26
- activity_category (core.models.SectionInfo attribute), 53
- activity_category() (core.models.Activity property), 26
- activity_category_id (core.models.SectionInfo attribute), 53
- activity_details (core.models.AccountAlias attribute), 24
- activity_details (core.models.SectionInfo attribute), 53
- activity_details_id (core.models.AccountAlias attribute), 24
- activity_details_id (core.models.SectionInfo attribute), 53
- activity_id (core.models.ActivityDetails attribute), 29
- activity_id (core.models.PaymentSchedule attribute), 46
- activity_number (core.models.AccountAliasMapping attribute), 25
- activity_set (core.models.ActivityDetails attribute), 29
- activity_type (core.models.Activity attribute), 26
- ActivityAdmin (class in core.admin), 10
- ActivityCategory (class in core.models), 28
- ActivityCategory.DoesNotExist, 28
- ActivityCategory.MultipleObjectsReturned, 28
- ActivityCategoryAdmin (class in core.admin), 10
- ActivityCategorySerializer (class in core.serializers), 58
- ActivityCategoryViewSet (class in core.views), 64
- ActivityDetails (class in core.models), 29
- ActivityDetails.DoesNotExist, 29
- ActivityDetails.MultipleObjectsReturned, 29
- ActivityDetailsAdmin (class in core.admin), 11
- ActivityDetailsSerializer (class in core.serializers), 58
- ActivityDetailsViewSet (class in core.views), 64
- ActivityFilter (class in core.filters), 19
- activityId2name() (built-in function), 163
- ActivityQuerySet (class in core.managers), 22
- ActivitySerializer (class in core.serializers), 58
- ActivityViewSet (class in core.views), 64
- acts_on (core.models.Municipality attribute), 41
- ADMIN (core.models.User attribute), 56
- alias (core.models.AccountAlias attribute), 24
- alias (core.models.AccountAliasMapping attribute), 25
- allowed_for_steps (core.models.Section attribute), 52
- allowed_for_target_groups (core.models.Effort attribute), 34
- allowed_for_target_groups (core.models.Section attribute), 52
- AllowedForStepsFilter (class in core.filters), 19
- amount (core.models.Payment attribute), 42
- amount_case (core.managers.PaymentQuerySet attribute), 22
- amount_sum() (core.managers.PaymentQuerySet method), 22
- annotate_main_activity_details_id() (core.managers.AppropriationQuerySet method), 22
- annotate_paid_date_or_date() (core.managers.PaymentQuerySet method), 22
- applicable_payments() (core.models.Activity property), 26
- Appropriation (class in core.models), 30
- appropriation (core.models.Activity attribute), 26
- Appropriation.DoesNotExist, 30
- Appropriation.MultipleObjectsReturned, 30
- appropriation_date (core.models.Activity attribute), 26
- appropriation_id (core.models.Activity attribute), 26
- AppropriationAdmin (class in core.admin), 11
- AppropriationFilter (class in core.filters), 20
- AppropriationQuerySet (class in core.managers), 22
- appropriations (core.models.Case attribute), 33
- appropriations (core.models.Section attribute), 52
- AppropriationSerializer (class in core.serializers), 58
- AppropriationViewSet (class in core.views), 64
- approval_level (core.models.Activity attribute), 26
- approval_level_id (core.models.Activity attribute), 26
- approval_note (core.models.Activity attribute), 26
- approval_user (core.models.Activity attribute), 26
- approval_user_id (core.models.Activity attribute), 27
- ApprovalLevel (class in core.models), 31
- ApprovalLevel.DoesNotExist, 32
- ApprovalLevel.MultipleObjectsReturned, 32
- ApprovalLevelAdmin (class in core.admin), 11
- ApprovalLevelSerializer (class in core.serializers), 59
- ApprovalLevelViewSet (class in core.views), 65
- approved_activities (core.models.ApprovalLevel

- attribute), 32
 approved_activities (core.models.User attribute), 56
 assessment_comment (core.models.Case attribute), 33
 assessment_comment (core.models.HistoricalCase attribute), 36
 AuditMixin (class in core.mixins), 23
 AuditModelMixin (class in core.mixins), 23
 AuditModelViewSetMixin (class in core.mixins), 23
 AuditViewSet (class in core.views), 65
 authentication_classes (core.views.AuditViewSet attribute), 65
 authentication_classes (core.views.IsEditingPastPaymentsAllowed attribute), 67
 auto_filters (core.filters.ActivityFilter attribute), 19
 auto_filters (core.filters.AllowedForStepsFilter attribute), 20
 auto_filters (core.filters.AppropriationFilter attribute), 20
 auto_filters (core.filters.CaseFilter attribute), 20
 auto_filters (core.filters.CaseForAppropriationFilter attribute), 20
 auto_filters (core.filters.CaseForPaymentFilter attribute), 20
 auto_filters (core.filters.PaymentFilter attribute), 21
 auto_filters (core.filters.PaymentScheduleFilter attribute), 21
 auto_filters (core.filters.UserForCaseFilter attribute), 21
- ## B
- base_filters (core.filters.ActivityFilter attribute), 19
 base_filters (core.filters.AllowedForStepsFilter attribute), 20
 base_filters (core.filters.AppropriationFilter attribute), 20
 base_filters (core.filters.CaseFilter attribute), 20
 base_filters (core.filters.CaseForAppropriationFilter attribute), 20
 base_filters (core.filters.CaseForPaymentFilter attribute), 20
 base_filters (core.filters.PaymentFilter attribute), 21
 base_filters (core.filters.PaymentScheduleFilter attribute), 21
 base_filters (core.filters.UserForCaseFilter attribute), 21
 BaseAppropriationSerializer (class in core.serializers), 59
 basename (core.views.ActivityDetailsViewSet attribute), 64
 basename (core.views.ActivityViewSet attribute), 64
 basename (core.views.AppropriationViewSet attribute), 65
 basename (core.views.ApprovalLevelViewSet attribute), 65
 basename (core.views.CaseViewSet attribute), 65
 basename (core.views.EffortStepViewSet attribute), 66
 basename (core.views.EffortViewSet attribute), 66
 basename (core.views.InternalPaymentRecipientViewSet attribute), 67
 basename (core.views.MunicipalityViewSet attribute), 67
 basename (core.views.PaymentMethodDetailsViewSet attribute), 67
 basename (core.views.PaymentScheduleViewSet attribute), 68
 basename (core.views.PaymentViewSet attribute), 68
 basename (core.views.PriceViewSet attribute), 68
 basename (core.views.RateViewSet attribute), 69
 basename (core.views.RelatedPersonViewSet attribute), 69
 basename (core.views.SchoolDistrictViewSet attribute), 70
 basename (core.views.SectionInfoViewSet attribute), 70
 basename (core.views.SectionViewSet attribute), 70
 basename (core.views.ServiceProviderViewSet attribute), 71
 basename (core.views.TargetGroupViewSet attribute), 71
 basename (core.views.TeamViewSet attribute), 71
 basename (core.views.UserViewSet attribute), 71
 bevillingsplatform
 module, 74
 bevillingsplatform.initialize
 module, 72
 bevillingsplatform.settings
 module, 73
 bevillingsplatform.urls
 module, 73
 bevillingsplatform.wsgi
 module, 74
 BIWEEKLY (core.models.PaymentSchedule attribute), 45
- ## C
- calculate_per_payment_amount()
 (core.models.PaymentSchedule method), 46
 can_be_paid() (core.models.PaymentSchedule property), 46
 can_delete (core.admin.HistoricalRatePerDateInline attribute), 13

- `can_delete` (*core.admin.RatePerDateInline* attribute), 15
 - `Case` (class in *core.models*), 32
 - `case` (*core.models.Appropriation* attribute), 30
 - `Case.DoesNotExist`, 32
 - `Case.MultipleObjectsReturned`, 33
 - `case_id` (*core.models.Appropriation* attribute), 30
 - `case_set` (*core.models.EffortStep* attribute), 35
 - `case_worker` (*core.models.Case* attribute), 33
 - `case_worker` (*core.models.HistoricalCase* attribute), 36
 - `case_worker_id` (*core.models.Case* attribute), 33
 - `case_worker_id` (*core.models.HistoricalCase* attribute), 36
 - `CaseAdmin` (class in *core.admin*), 11
 - `CaseFilter` (class in *core.filters*), 20
 - `CaseForAppropriationFilter` (class in *core.filters*), 20
 - `CaseForPaymentFilter` (class in *core.filters*), 20
 - `CaseQuerySet` (class in *core.managers*), 22
 - `cases` (*core.models.Effort* attribute), 35
 - `cases` (*core.models.SchoolDistrict* attribute), 51
 - `cases` (*core.models.TargetGroup* attribute), 54
 - `cases` (*core.models.User* attribute), 56
 - `CaseSerializer` (class in *core.serializers*), 59
 - `CaseViewSet` (class in *core.views*), 65
 - `category_id` (*core.models.ActivityCategory* attribute), 28
 - `change_case_worker` () (*core.views.CaseViewSet* method), 65
 - `change_list_template` (*core.admin.AccountAliasMappingAdmin* attribute), 9
 - `changed_by` (*core.models.HistoricalRatePerDate* attribute), 39
 - `changed_by` (*core.models.RatePerDate* attribute), 49
 - `changed_by_id` (*core.models.HistoricalRatePerDate* attribute), 39
 - `changed_by_id` (*core.models.RatePerDate* attribute), 49
 - `changed_date` (*core.models.HistoricalRatePerDate* attribute), 39
 - `changed_date` (*core.models.RatePerDate* attribute), 49
 - `changelist_view` () (*core.admin.AccountAliasMappingAdmin* method), 10
 - `classes` (*core.admin.HistoricalRatePerDateInline* attribute), 13
 - `Classification` (class in *core.models*), 34
 - `ClassificationAdmin` (class in *core.admin*), 11
 - `ClassificationInline` (class in *core.admin*), 11
 - `ClassificationViewSetMixin` (class in *core.mixins*), 24
 - `clean_required_fields_for_case` () (*core.admin.TargetGroupForm* method), 18
 - `COMPANY` (*core.models.PaymentSchedule* attribute), 45
 - `core`
 - module, 72
 - `core.admin`
 - module, 9
 - `core.apps`
 - module, 19
 - `core.authentication`
 - module, 19
 - `core.filters`
 - module, 19
 - `core.managers`
 - module, 22
 - `core.mixins`
 - module, 23
 - `core.models`
 - module, 24
 - `core.permissions`
 - module, 58
 - `core.serializers`
 - module, 58
 - `core.signals`
 - module, 61
 - `core.utils`
 - module, 62
 - `core.views`
 - module, 64
 - `CoreConfig` (class in *core.apps*), 19
 - `cpr_number` (*core.models.Case* attribute), 33
 - `cpr_number` (*core.models.RelatedPerson* attribute), 50
 - `create` () (*core.serializers.PriceSerializer* method), 60
 - `create_interval` () (*core.models.VariableRate* static method), 57
 - `create_rrule` () (*core.models.PaymentSchedule* method), 46
 - `create_rrule` () (in module *core.utils*), 62
 - `created` (*core.mixins.AuditModelMixin* attribute), 23
 - `created` (*core.models.HistoricalCase* attribute), 36
 - `CsrfExemptSessionAuthentication` (class in *core.authentication*), 19
 - `CustomUserAdmin` (class in *core.admin*), 12
 - `cvr_number` (*core.models.ServiceProvider* attribute), 54
- ## D
- `DAILY` (*core.models.PaymentSchedule* attribute), 45
 - `date` (*core.models.Payment* attribute), 42
 - `date` (*core.models.PaymentDateExclusion* attribute), 44
 - `declared_filters` (*core.filters.ActivityFilter* attribute), 19

declared_filters (*core.filters.AllowedForStepsFilter attribute*), 20
 declared_filters (*core.filters.AppropriationFilter attribute*), 20
 declared_filters (*core.filters.CaseFilter attribute*), 20
 declared_filters (*core.filters.CaseForAppropriationFilter attribute*), 20
 declared_filters (*core.filters.CaseForPaymentFilter attribute*), 20
 declared_filters (*core.filters.PaymentFilter attribute*), 21
 declared_filters (*core.filters.PaymentScheduleFilter attribute*), 21
 declared_filters (*core.filters.UserForCaseFilter attribute*), 21
 DeletePaymentPermission (class in *core.permissions*), 58
 description (*core.models.ActivityDetails attribute*), 29
 description (*core.models.Effort attribute*), 35
 description (*core.models.Rate attribute*), 49
 description (*core.views.ActivityDetailsViewSet attribute*), 64
 description (*core.views.ActivityViewSet attribute*), 64
 description (*core.views.AppropriationViewSet attribute*), 65
 description (*core.views.ApprovalLevelViewSet attribute*), 65
 description (*core.views.CaseViewSet attribute*), 66
 description (*core.views.EffortStepViewSet attribute*), 66
 description (*core.views.EffortViewSet attribute*), 66
 description (*core.views.InternalPaymentRecipientViewSet attribute*), 67
 description (*core.views.MunicipalityViewSet attribute*), 67
 description (*core.views.PaymentMethodDetailsViewSet attribute*), 67
 description (*core.views.PaymentScheduleViewSet attribute*), 68
 description (*core.views.PaymentViewSet attribute*), 68
 description (*core.views.PriceViewSet attribute*), 69
 description (*core.views.RateViewSet attribute*), 69
 description (*core.views.RelatedPersonViewSet attribute*), 69
 description (*core.views.SchoolDistrictViewSet attribute*), 70
 description (*core.views.SectionInfoViewSet attribute*), 70
 description (*core.views.SectionViewSet attribute*), 70
 description (*core.views.ServiceProviderViewSet attribute*), 71
 description (*core.views.TargetGroupViewSet attribute*), 71
 description (*core.views.TeamViewSet attribute*), 71
 description (*core.views.UserViewSet attribute*), 71
 destroy () (*core.views.ActivityViewSet method*), 64
 detail (*core.views.ActivityDetailsViewSet attribute*), 64
 detail (*core.views.ActivityViewSet attribute*), 64
 detail (*core.views.AppropriationViewSet attribute*), 65
 detail (*core.views.ApprovalLevelViewSet attribute*), 65
 detail (*core.views.CaseViewSet attribute*), 66
 detail (*core.views.EffortStepViewSet attribute*), 66
 detail (*core.views.EffortViewSet attribute*), 66
 detail (*core.views.InternalPaymentRecipientViewSet attribute*), 67
 detail (*core.views.MunicipalityViewSet attribute*), 67
 detail (*core.views.PaymentMethodDetailsViewSet attribute*), 67
 detail (*core.views.PaymentScheduleViewSet attribute*), 68
 detail (*core.views.PaymentViewSet attribute*), 68
 detail (*core.views.PriceViewSet attribute*), 69
 detail (*core.views.RateViewSet attribute*), 69
 detail (*core.views.RelatedPersonViewSet attribute*), 69
 detail (*core.views.SchoolDistrictViewSet attribute*), 70
 detail (*core.views.SectionInfoViewSet attribute*), 70
 detail (*core.views.SectionViewSet attribute*), 70
 detail (*core.views.ServiceProviderViewSet attribute*), 71
 detail (*core.views.TargetGroupViewSet attribute*), 71
 detail (*core.views.TeamViewSet attribute*), 71
 detail (*core.views.UserViewSet attribute*), 72
 details (*core.models.Activity attribute*), 27
 details_id (*core.models.Activity attribute*), 27
 displayEffort () (*built-in function*), 164
 displayStatus () (*built-in function*), 164
 district (*core.models.Case attribute*), 33
 district_id (*core.models.Case attribute*), 33
 districtId2name () (*built-in function*), 163
 due_payments_for_prism () (in module *core.utils*), 62
 due_payments_for_prism_with_exclusions () (in module *core.utils*), 62

E

EDIT (*core.models.User attribute*), 56
 EditPaymentPermission (class in *core.permissions*), 58
 Effort (class in *core.models*), 34
 Effort.DoesNotExist, 34
 Effort.MultipleObjectsReturned, 34
 effort_step (*core.models.Case attribute*), 33

effort_step (*core.models.HistoricalCase* attribute), 36
 effort_step_id (*core.models.Case* attribute), 33
 effort_step_id (*core.models.HistoricalCase* attribute), 36
 EffortAdmin (*class in core.admin*), 12
 efforts (*core.models.Case* attribute), 33
 efforts (*core.models.TargetGroup* attribute), 54
 EffortSerializer (*class in core.serializers*), 59
 EffortStep (*class in core.models*), 35
 EffortStep.DoesNotExist, 35
 EffortStep.MultipleObjectsReturned, 35
 EffortStepAdmin (*class in core.admin*), 12
 EffortStepSerializer (*class in core.serializers*), 59
 EffortStepViewSet (*class in core.views*), 66
 EffortViewSet (*class in core.views*), 66
 end_date (*core.models.Activity* attribute), 27
 end_date (*core.models.HistoricalRatePerDate* attribute), 39
 end_date (*core.models.RatePerDate* attribute), 49
 enforce_csrf() (*core.authentication.CsrfExemptSessionAuthentication* method), 19
 exclude (*core.admin.ActivityDetailsAdmin* attribute), 11
 exclude (*core.admin.HistoricalRatePerDateInline* attribute), 13
 exclude (*core.admin.RateAdmin* attribute), 15
 expected_payments_for_report_list() (*core.managers.PaymentQuerySet* method), 22
 expired() (*core.managers.ActivityQuerySet* method), 22
 expired() (*core.managers.AppropriationQuerySet* method), 22
 expired() (*core.managers.CaseQuerySet* method), 22
 expired() (*core.models.Case* property), 33
 export_prism_payments_for_date() (*in module core.utils*), 62
 extra (*core.admin.HistoricalRatePerDateInline* attribute), 13
 extra (*core.admin.MainActivityDetailsInline* attribute), 13
 extra (*core.admin.RatePerDateInline* attribute), 15
 extra (*core.admin.SectionEffortStepProxyInline* attribute), 16
 extra (*core.admin.SectionInfoActivityCategoryInline* attribute), 16
 extra (*core.admin.SectionInfoInline* attribute), 17
 extra (*core.admin.SupplementaryActivityDetailsInline* attribute), 17
 extra (*core.admin.SupplementaryActivityInline* attribute), 17

F

fetch_from_serviceplatformen() (*core.views.RelatedPersonViewSet* method), 69
 fetchCase() (*built-in function*), 165
 fetchCases() (*built-in function*), 164
 fictive (*core.models.PaymentSchedule* attribute), 46
 fields (*core.admin.TargetGroupAdmin* attribute), 18
 filter_allowed_for_steps() (*core.filters.AllowedForStepsFilter* method), 20
 filter_expired() (*core.filters.CaseFilter* method), 20
 filter_horizontal (*core.admin.ActivityDetailsAdmin* attribute), 11
 filter_horizontal (*core.admin.EffortAdmin* attribute), 12
 filter_horizontal (*core.admin.SectionAdmin* attribute), 16
 filter_paid_date_or_date_gte() (*core.filters.PaymentFilter* method), 21
 filter_paid_date_or_date_lte() (*core.filters.PaymentFilter* method), 21
 filter_paid_date_or_date_month() (*core.filters.PaymentFilter* method), 21
 filter_paid_date_or_date_week() (*core.filters.PaymentFilter* method), 21
 filter_paid_date_or_date_year() (*core.filters.PaymentFilter* method), 21
 filterset_class (*core.views.AppropriationViewSet* attribute), 65
 filterset_class (*core.views.CaseViewSet* attribute), 66
 filterset_class (*core.views.PaymentViewSet* attribute), 68
 filterset_class (*core.views.SectionViewSet* attribute), 70
 filterset_fields (*core.views.ActivityDetailsViewSet* attribute), 64
 filterset_fields (*core.views.ActivityViewSet* attribute), 64
 filterset_fields (*core.views.ApprovalLevelViewSet* attribute), 65
 filterset_fields (*core.views.EffortStepViewSet* attribute), 66
 filterset_fields (*core.views.EffortViewSet* attribute), 66
 filterset_fields (*core.views.MunicipalityViewSet* attribute), 67
 filterset_fields (*core.views.PaymentViewSet* attribute), 68
 filterset_fields (*core.views.RelatedPersonViewSet* attribute), 69
 filterset_fields (*core.views.SchoolDistrictViewSet* attribute), 70

`filterset_fields (core.views.SectionInfoViewSet attribute), 70`
`filterset_fields (core.views.ServiceProviderViewSet attribute), 71`
`finalize_response() (core.mixins.AuditMixin method), 23`
`FIXED_PRICE (core.models.PaymentSchedule attribute), 45`
`form (core.admin.PriceAdmin attribute), 15`
`form (core.admin.RateAdmin attribute), 15`
`form (core.admin.TargetGroupAdmin attribute), 18`
`form (core.admin.VariableRateAdmin attribute), 18`
`format_prism_financial_record() (in module core.utils), 62`
`format_prism_payment_record() (in module core.utils), 62`
`from_serviceplatformen (core.models.RelatedPerson attribute), 50`

G

`generate_payment_date_exclusion_dates() (in module core.utils), 62`
`generate_payments() (core.models.PaymentSchedule method), 46`
`generate_payments_on_post_save() (in module core.signals), 61`
`generate_payments_report() (in module core.utils), 62`
`generate_payments_report_list_v0() (in module core.utils), 62`
`generate_payments_report_list_v1() (in module core.utils), 62`
`generate_payments_report_list_v2() (in module core.utils), 62`
`generate_records_for_prism() (in module core.utils), 63`
`generic_time_choices (core.filters.PaymentFilter attribute), 21`
`get() (core.views.IsEditingPastPaymentsAllowed method), 67`
`get_activities() (core.serializers.AppropriationSerializer method), 59`
`get_activity_type_display() (core.models.Activity method), 27`
`get_all_modified_by_activities() (core.models.Activity method), 27`
`get_cpr_data() (in module core.utils), 63`
`get_cpr_data_mock() (in module core.utils), 63`
`get_default_history_user() (core.models.HistoricalCase static method), 36`
`get_default_history_user() (core.models.HistoricalPayment static method), 38`
`get_default_history_user() (core.models.HistoricalRatePerDate static method), 39`
`get_history_type_display() (core.models.HistoricalCase method), 36`
`get_history_type_display() (core.models.HistoricalPayment method), 38`
`get_history_type_display() (core.models.HistoricalRatePerDate method), 40`
`get_main_activity_main_account_number() (core.models.SectionInfo method), 53`
`get_next_by_date() (core.models.Payment method), 42`
`get_next_by_date() (core.models.PaymentDateExclusion method), 44`
`get_next_by_date_joined() (core.models.User method), 56`
`get_next_by_history_date() (core.models.HistoricalCase method), 36`
`get_next_by_history_date() (core.models.HistoricalPayment method), 38`
`get_next_by_history_date() (core.models.HistoricalRatePerDate method), 40`
`get_num_ongoing_activities() (core.serializers.BaseAppropriationSerializer method), 59`
`get_num_ongoing_appropriations() (core.serializers.CaseSerializer method), 59`
`get_num_ongoing_draft_or_expected_activities() (core.serializers.BaseAppropriationSerializer method), 59`
`get_num_ongoing_draft_or_expected_appropriations() (core.serializers.CaseSerializer method), 59`
`get_payment_cost_type_display() (core.models.PaymentSchedule method), 46`
`get_payment_frequency_display() (core.models.PaymentSchedule method), 46`
`get_payment_method_display() (core.models.Payment method), 42`
`get_payment_method_display() (core.models.PaymentSchedule method), 46`
`get_payment_type_display() (core.models.PaymentSchedule method), 46`
`get_person_info() (in module core.utils), 63`

`get_previous_by_date()` (*core.models.Payment method*), 42
`get_previous_by_date()` (*core.models.PaymentDateExclusion method*), 44
`get_previous_by_date_joined()` (*core.models.User method*), 56
`get_previous_by_history_date()` (*core.models.HistoricalCase method*), 36
`get_previous_by_history_date()` (*core.models.HistoricalPayment method*), 38
`get_previous_by_history_date()` (*core.models.HistoricalRatePerDate method*), 40
`get_profile_display()` (*core.models.User method*), 56
`get_queryset()` (*core.mixins.ClassificationViewSetMixin method*), 24
`get_queryset()` (*core.views.ActivityViewSet method*), 64
`get_queryset()` (*core.views.AppropriationViewSet method*), 65
`get_queryset()` (*core.views.CaseViewSet method*), 66
`get_queryset()` (*core.views.PaymentScheduleViewSet method*), 68
`get_rate_amount()` (*core.models.VariableRate method*), 57
`get_recipient_type_display()` (*core.models.Payment method*), 42
`get_recipient_type_display()` (*core.models.PaymentSchedule method*), 46
`get_required_fields_for_case()` (*core.models.TargetGroup method*), 54
`get_scaling_step_display()` (*core.models.Case method*), 33
`get_scaling_step_display()` (*core.models.HistoricalCase method*), 36
`get_serializer_class()` (*core.views.AppropriationViewSet method*), 65
`get_status_display()` (*core.models.Activity method*), 27
`get_supplementary_activity_main_account_number()` (*core.models.SectionInfo method*), 53
`get_tax_card_display()` (*core.models.PaymentMethodDetails method*), 44
`get_urls()` (*core.admin.AccountAliasMappingAdmin method*), 10
`getCase()` (*built-in function*), 164
`getCases()` (*built-in function*), 164
`GLOBAL_RATE_PRICE` (*core.models.PaymentSchedule attribute*), 45
`GRANT` (*core.models.User attribute*), 56
`grant()` (*core.models.Activity method*), 27
`grant()` (*core.models.Appropriation method*), 30
`grant()` (*core.views.AppropriationViewSet method*), 65
`granted_from_date()` (*core.models.Appropriation property*), 30
`granted_to_date()` (*core.models.Appropriation property*), 31
`group_by_monthly_amounts()` (*core.managers.PaymentQuerySet method*), 22
`groups` (*core.models.User attribute*), 56

H

`has_add_permission()` (*core.admin.ClassificationAdmin method*), 11
`has_add_permission()` (*core.admin.ClassificationInline method*), 12
`has_add_permission()` (*core.admin.CustomUserAdmin method*), 12
`has_add_permission()` (*core.admin.HistoricalRatePerDateInline method*), 13
`has_add_permission()` (*core.admin.RatePerDateInline method*), 15
`has_change_permission()` (*core.admin.ClassificationAdmin method*), 11
`has_change_permission()` (*core.admin.ClassificationInline method*), 12
`has_change_permission()` (*core.admin.CustomUserAdmin method*), 12
`has_change_permission()` (*core.admin.HistoricalRatePerDateInline method*), 13
`has_delete_permission()` (*core.admin.ClassificationAdmin method*), 11
`has_delete_permission()` (*core.admin.ClassificationInline method*), 12
`has_delete_permission()` (*core.admin.CustomUserAdmin method*), 12
`has_module_permission()` (*core.admin.ClassificationAdmin method*), 11

`has_module_permission()`
 (*core.admin.CustomUserAdmin* *method*),
 12
`has_object_permission()`
 (*core.permissions.DeletePaymentPermission*
 method), 58
`has_object_permission()`
 (*core.permissions.EditPaymentPermission*
 method), 58
`has_permission()` (*core.permissions.IsUserAllowed*
 method), 58
`has_permission()` (*core.permissions.NewPaymentPermission*
 method), 58
`has_view_permission()`
 (*core.admin.ClassificationAdmin* *method*),
 11
`has_view_permission()`
 (*core.admin.ClassificationInline* *method*),
 12
`has_view_permission()`
 (*core.admin.CustomUserAdmin* *method*),
 12
HistoricalCase (*class in core.models*), 35
HistoricalCase.DoesNotExist, 36
HistoricalCase.MultipleObjectsReturned,
 36
HistoricalCaseSerializer (*class in*
 core.serializers), 59
HistoricalPayment (*class in core.models*), 37
HistoricalPayment.DoesNotExist, 38
HistoricalPayment.MultipleObjectsReturned,
 38
HistoricalRatePerDate (*class in core.models*), 39
HistoricalRatePerDate.DoesNotExist, 39
HistoricalRatePerDate.MultipleObjectsReturned,
 39
HistoricalRatePerDateInline (*class in*
 core.admin), 12
history (*core.models.Case* *attribute*), 33
history (*core.models.Payment* *attribute*), 43
history (*core.models.RatePerDate* *attribute*), 49
history() (*core.views.CaseViewSet* *method*), 66
history_change_reason
 (*core.models.HistoricalCase* *attribute*), 36
history_change_reason
 (*core.models.HistoricalPayment* *attribute*),
 38
history_change_reason
 (*core.models.HistoricalRatePerDate* *attribute*),
 40
history_date (*core.models.HistoricalCase* *at-*
 tribute), 37
history_date (*core.models.HistoricalPayment*
 attribute), 38
history_date (*core.models.HistoricalRatePerDate*
 attribute), 40
history_id (*core.models.HistoricalCase* *attribute*),
 37
history_id (*core.models.HistoricalPayment* *at-*
 tribute), 38
history_id (*core.models.HistoricalRatePerDate* *at-*
 tribute), 40
history_object (*core.models.HistoricalCase*
 attribute), 37
history_object (*core.models.HistoricalPayment* *at-*
 tribute), 38
history_object (*core.models.HistoricalRatePerDate*
 attribute), 40
history_type (*core.models.HistoricalCase* *at-*
 tribute), 37
history_type (*core.models.HistoricalPayment*
 attribute), 38
history_type (*core.models.HistoricalRatePerDate*
 attribute), 40
history_user (*core.models.HistoricalCase* *at-*
 tribute), 37
history_user (*core.models.HistoricalPayment*
 attribute), 38
history_user (*core.models.HistoricalRatePerDate*
 attribute), 40
history_user_id (*core.models.HistoricalCase* *at-*
 tribute), 37
history_user_id (*core.models.HistoricalPayment*
 attribute), 38
history_user_id (*core.models.HistoricalRatePerDate*
 attribute), 40
id (*core.models.AccountAlias* *attribute*), 24
id (*core.models.AccountAliasMapping* *attribute*), 25
id (*core.models.Activity* *attribute*), 27
id (*core.models.ActivityCategory* *attribute*), 28
id (*core.models.ActivityDetails* *attribute*), 29
id (*core.models.Appropriation* *attribute*), 31
id (*core.models.ApprovalLevel* *attribute*), 32
id (*core.models.Case* *attribute*), 33
id (*core.models.Effort* *attribute*), 35
id (*core.models.EffortStep* *attribute*), 35
id (*core.models.HistoricalCase* *attribute*), 37
id (*core.models.HistoricalPayment* *attribute*), 38
id (*core.models.HistoricalRatePerDate* *attribute*), 40
id (*core.models.InternalPaymentRecipient* *attribute*), 41
id (*core.models.Municipality* *attribute*), 41
id (*core.models.Payment* *attribute*), 43
id (*core.models.PaymentDateExclusion* *attribute*), 44
id (*core.models.PaymentMethodDetails* *attribute*), 44
id (*core.models.PaymentSchedule* *attribute*), 46
id (*core.models.RatePerDate* *attribute*), 49

- `id (core.models.RelatedPerson attribute)`, 50
 - `id (core.models.SchoolDistrict attribute)`, 51
 - `id (core.models.Section attribute)`, 52
 - `id (core.models.SectionInfo attribute)`, 53
 - `id (core.models.ServiceProvider attribute)`, 54
 - `id (core.models.TargetGroup attribute)`, 54
 - `id (core.models.Team attribute)`, 55
 - `id (core.models.User attribute)`, 56
 - `id (core.models.VariableRate attribute)`, 57
 - `in_this_year() (core.managers.PaymentQuerySet method)`, 23
 - `INDIVIDUAL_PAYMENT (core.models.PaymentSchedule attribute)`, 45
 - `initialize() (in module bevillingsplatform.initialize)`, 72
 - `initialize_account_alias_mappings() (in module bevillingsplatform.initialize)`, 72
 - `initialize_account_aliases() (in module bevillingsplatform.initialize)`, 72
 - `initialize_activity_categories() (in module bevillingsplatform.initialize)`, 72
 - `initialize_activity_details() (in module bevillingsplatform.initialize)`, 72
 - `initialize_approval_levels() (in module bevillingsplatform.initialize)`, 72
 - `initialize_municipalities() (in module bevillingsplatform.initialize)`, 72
 - `initialize_payment_method_details() (in module bevillingsplatform.initialize)`, 72
 - `initialize_rates() (in module bevillingsplatform.initialize)`, 72
 - `initialize_school_districts() (in module bevillingsplatform.initialize)`, 72
 - `initialize_section_infos() (in module bevillingsplatform.initialize)`, 73
 - `initialize_sections() (in module bevillingsplatform.initialize)`, 73
 - `initialize_service_providers() (in module bevillingsplatform.initialize)`, 73
 - `initialize_target_groups() (in module bevillingsplatform.initialize)`, 73
 - `initialize_teams() (in module bevillingsplatform.initialize)`, 73
 - `initialize_users() (in module bevillingsplatform.initialize)`, 73
 - `inlines (core.admin.ActivityCategoryAdmin attribute)`, 10
 - `inlines (core.admin.ActivityDetailsAdmin attribute)`, 11
 - `inlines (core.admin.EffortStepAdmin attribute)`, 12
 - `inlines (core.admin.SectionAdmin attribute)`, 16
 - `inlines (core.admin.VariableRateAdmin attribute)`, 18
 - `instance() (core.models.HistoricalCase property)`, 37
 - `instance() (core.models.HistoricalPayment property)`, 38
 - `instance() (core.models.HistoricalRatePerDate property)`, 40
 - `instance_type (core.models.HistoricalCase attribute)`, 37
 - `instance_type (core.models.HistoricalPayment attribute)`, 38
 - `instance_type (core.models.HistoricalRatePerDate attribute)`, 40
 - `INTERNAL (core.models.PaymentSchedule attribute)`, 45
 - `InternalPaymentRecipient (class in core.models)`, 40
 - `InternalPaymentRecipient.DoesNotExist`, 41
 - `InternalPaymentRecipient.MultipleObjectsReturned`, 41
 - `InternalPaymentRecipientAdmin (class in core.admin)`, 13
 - `InternalPaymentRecipientSerializer (class in core.serializers)`, 59
 - `InternalPaymentRecipientViewSet (class in core.views)`, 67
 - `is_payable_manually() (core.models.Payment property)`, 43
 - `is_payment_and_recipient_allowed() (core.models.PaymentSchedule static method)`, 46
 - `is_ready_to_generate_payments() (core.models.PaymentSchedule method)`, 46
 - `is_valid_activity_start_date() (core.models.Activity method)`, 27
 - `is_workflow_engine_or_admin() (core.models.User method)`, 56
 - `IsEditingPastPaymentsAllowed (class in core.views)`, 67
 - `IsUserAllowed (class in core.permissions)`, 58
- ## K
- `kle_number (core.models.SectionInfo attribute)`, 53
- ## L
- `law_text_name (core.models.Section attribute)`, 52
 - `leader (core.models.Team attribute)`, 55
 - `leader_id (core.models.Team attribute)`, 55
 - `list_display (core.admin.AccountAliasAdmin attribute)`, 9
 - `list_display (core.admin.AccountAliasMappingAdmin attribute)`, 10
 - `list_display (core.admin.ActivityCategoryAdmin attribute)`, 11

list_display (*core.admin.ActivityDetailsAdmin attribute*), 11
 list_display (*core.admin.ApprovalLevelAdmin attribute*), 11
 list_display (*core.admin.EffortAdmin attribute*), 12
 list_display (*core.admin.EffortStepAdmin attribute*), 12
 list_display (*core.admin.InternalPaymentRecipientAdmin attribute*), 13
 list_display (*core.admin.MunicipalityAdmin attribute*), 13
 list_display (*core.admin.PaymentAdmin attribute*), 14
 list_display (*core.admin.PaymentDateExclusionAdmin attribute*), 14
 list_display (*core.admin.PaymentScheduleAdmin attribute*), 15
 list_display (*core.admin.PriceAdmin attribute*), 15
 list_display (*core.admin.RateAdmin attribute*), 15
 list_display (*core.admin.SchoolDistrictAdmin attribute*), 16
 list_display (*core.admin.SectionAdmin attribute*), 16
 list_display (*core.admin.SectionInfoAdmin attribute*), 17
 list_display (*core.admin.ServiceProviderAdmin attribute*), 17
 list_display (*core.admin.TargetGroupAdmin attribute*), 18
 list_filter (*core.admin.PaymentAdmin attribute*), 14
 list_filter (*core.admin.PaymentDateExclusionAdmin attribute*), 14
 list_filter (*core.admin.PaymentScheduleAdmin attribute*), 15
 list_main_activity_for() (*core.admin.SectionAdmin method*), 16
 list_sections() (*core.admin.EffortStepAdmin method*), 12
 list_supplementary_activity_for() (*core.admin.SectionAdmin method*), 16
 ListAppropriationSerializer (*class in core.serializers*), 59
 log_methods (*core.mixins.AuditMixin attribute*), 23
 logentry_set (*core.models.User attribute*), 56
 logger (*core.mixins.AuditMixin attribute*), 23

M
 main() (*in module manage*), 74
 main_account_number (*core.models.AccountAliasMapping attribute*), 25
 main_activities (*core.models.ActivityDetails attribute*), 29
 main_activities (*core.models.Section attribute*), 52
 main_activity() (*core.models.Appropriation property*), 31
 main_activity_for (*core.models.ActivityDetails attribute*), 29
 main_activity_main_account_number (*core.models.SectionInfo attribute*), 53
 main_case (*core.models.RelatedPerson attribute*), 50
 main_case_id (*core.models.RelatedPerson attribute*), 51
 main_rate (*core.models.HistoricalRatePerDate attribute*), 40
 main_rate (*core.models.RatePerDate attribute*), 50
 main_rate_id (*core.models.HistoricalRatePerDate attribute*), 40
 main_rate_id (*core.models.RatePerDate attribute*), 50
 MainActivityDetailsInline (*class in core.admin*), 13
 manage module, 74
 managed_teams (*core.models.User attribute*), 56
 max_profile() (*core.models.User class method*), 57
 max_tolerance_in_dkk (*core.models.ActivityDetails attribute*), 29
 max_tolerance_in_percent (*core.models.ActivityDetails attribute*), 30
 media() (*core.admin.AccountAliasAdmin property*), 9
 media() (*core.admin.AccountAliasMappingAdmin property*), 10
 media() (*core.admin.AccountAliasMappingCSVFileUploadForm property*), 10
 media() (*core.admin.ActivityAdmin property*), 10
 media() (*core.admin.ActivityCategoryAdmin property*), 11
 media() (*core.admin.ActivityDetailsAdmin property*), 11
 media() (*core.admin.AppropriationAdmin property*), 11
 media() (*core.admin.ApprovalLevelAdmin property*), 11
 media() (*core.admin.CaseAdmin property*), 11
 media() (*core.admin.ClassificationAdmin property*), 11
 media() (*core.admin.ClassificationInline property*), 12
 media() (*core.admin.CustomUserAdmin property*), 12
 media() (*core.admin.EffortAdmin property*), 12
 media() (*core.admin.EffortStepAdmin property*), 12
 media() (*core.admin.HistoricalRatePerDateInline property*), 13
 media() (*core.admin.InternalPaymentRecipientAdmin property*), 13
 media() (*core.admin.MainActivityDetailsInline property*), 13
 media() (*core.admin.MunicipalityAdmin property*), 13

- `media()` (*core.admin.PaymentAdmin property*), 14
 - `media()` (*core.admin.PaymentDateExclusionAdmin property*), 14
 - `media()` (*core.admin.PaymentMethodDetails property*), 14
 - `media()` (*core.admin.PaymentScheduleAdmin property*), 15
 - `media()` (*core.admin.PriceAdmin property*), 15
 - `media()` (*core.admin.PriceForm property*), 15
 - `media()` (*core.admin.RateAdmin property*), 15
 - `media()` (*core.admin.RateForm property*), 15
 - `media()` (*core.admin.RatePerDateInline property*), 16
 - `media()` (*core.admin.RelatedPersonAdmin property*), 16
 - `media()` (*core.admin.SchoolDistrictAdmin property*), 16
 - `media()` (*core.admin.SectionAdmin property*), 16
 - `media()` (*core.admin.SectionEffortStepProxyInline property*), 16
 - `media()` (*core.admin.SectionInfoActivityCategoryInline property*), 17
 - `media()` (*core.admin.SectionInfoAdmin property*), 17
 - `media()` (*core.admin.SectionInfoInline property*), 17
 - `media()` (*core.admin.ServiceProviderAdmin property*), 17
 - `media()` (*core.admin.SupplementaryActivityDetailsInline property*), 17
 - `media()` (*core.admin.SupplementaryActivityInline property*), 17
 - `media()` (*core.admin.TargetGroupAdmin property*), 18
 - `media()` (*core.admin.TargetGroupForm property*), 18
 - `media()` (*core.admin.TeamAdmin property*), 18
 - `media()` (*core.admin.VariableRateAdmin property*), 18
 - `media()` (*core.admin.VariableRateAdminForm property*), 19
 - `model` (*core.admin.HistoricalRatePerDateInline attribute*), 13
 - `model` (*core.admin.MainActivityDetailsInline attribute*), 13
 - `model` (*core.admin.RatePerDateInline attribute*), 16
 - `model` (*core.admin.SectionEffortStepProxyInline attribute*), 16
 - `model` (*core.admin.SectionInfoActivityCategoryInline attribute*), 17
 - `model` (*core.admin.SectionInfoInline attribute*), 17
 - `model` (*core.admin.SupplementaryActivityDetailsInline attribute*), 17
 - `model` (*core.admin.SupplementaryActivityInline attribute*), 17
 - `modified` (*core.mixins.AuditModelMixin attribute*), 23
 - `modified` (*core.models.HistoricalCase attribute*), 37
 - `modified_by` (*core.models.Activity attribute*), 27
 - `modifies` (*core.models.Activity attribute*), 27
 - `modifies_id` (*core.models.Activity attribute*), 27
 - `module`
 - `bevillingsplatform`, 74
 - `bevillingsplatform.initialize`, 72
 - `bevillingsplatform.settings`, 73
 - `bevillingsplatform.urls`, 73
 - `bevillingsplatform.wsgi`, 74
 - `core`, 72
 - `core.admin`, 9
 - `core.apps`, 19
 - `core.authentication`, 19
 - `core.filters`, 19
 - `core.managers`, 22
 - `core.mixins`, 23
 - `core.models`, 24
 - `core.permissions`, 58
 - `core.serializers`, 58
 - `core.signals`, 61
 - `core.utils`, 62
 - `core.views`, 64
 - `manage`, 74
 - `MONTHLY` (*core.models.PaymentSchedule attribute*), 45
 - `monthly_payment_plan()` (*core.models.Activity property*), 27
 - `Municipality` (*class in core.models*), 41
 - `Municipality.DoesNotExist`, 41
 - `Municipality.MultipleObjectsReturned`, 41
 - `MunicipalityAdmin` (*class in core.admin*), 13
 - `municipalityId2name()` (*built-in function*), 163
 - `MunicipalitySerializer` (*class in core.serializers*), 60
 - `MunicipalityViewSet` (*class in core.views*), 67
- ## N
- `name` (*core.apps.CoreConfig attribute*), 19
 - `name` (*core.models.ActivityCategory attribute*), 28
 - `name` (*core.models.ActivityDetails attribute*), 30
 - `name` (*core.models.ApprovalLevel attribute*), 32
 - `name` (*core.models.Case attribute*), 33
 - `name` (*core.models.Effort attribute*), 35
 - `name` (*core.models.EffortStep attribute*), 35
 - `name` (*core.models.InternalPaymentRecipient attribute*), 41
 - `name` (*core.models.Municipality attribute*), 41
 - `name` (*core.models.Rate attribute*), 49
 - `name` (*core.models.RelatedPerson attribute*), 51
 - `name` (*core.models.SchoolDistrict attribute*), 51
 - `name` (*core.models.ServiceProvider attribute*), 54
 - `name` (*core.models.TargetGroup attribute*), 54
 - `name` (*core.models.Team attribute*), 55
 - `name` (*core.views.ActivityDetailsViewSet attribute*), 64
 - `name` (*core.views.ActivityViewSet attribute*), 64
 - `name` (*core.views.AppropriationViewSet attribute*), 65
 - `name` (*core.views.ApprovalLevelViewSet attribute*), 65

- name (*core.views.CaseViewSet* attribute), 66
 name (*core.views.EffortStepViewSet* attribute), 66
 name (*core.views.EffortViewSet* attribute), 66
 name (*core.views.InternalPaymentRecipientViewSet* attribute), 67
 name (*core.views.MunicipalityViewSet* attribute), 67
 name (*core.views.PaymentMethodDetailsViewSet* attribute), 68
 name (*core.views.PaymentScheduleViewSet* attribute), 68
 name (*core.views.PaymentViewSet* attribute), 68
 name (*core.views.PriceViewSet* attribute), 69
 name (*core.views.RateViewSet* attribute), 69
 name (*core.views.RelatedPersonViewSet* attribute), 69
 name (*core.views.SchoolDistrictViewSet* attribute), 70
 name (*core.views.SectionInfoViewSet* attribute), 70
 name (*core.views.SectionViewSet* attribute), 70
 name (*core.views.ServiceProviderViewSet* attribute), 71
 name (*core.views.TargetGroupViewSet* attribute), 71
 name (*core.views.TeamViewSet* attribute), 71
 name (*core.views.UserViewSet* attribute), 72
 needs_recalculation (*core.models.Rate* attribute), 49
 NewPaymentPermission (class in *core.permissions*), 58
 next_payment() (*core.models.PaymentSchedule* property), 46
 next_record() (*core.models.HistoricalCase* property), 37
 next_record() (*core.models.HistoricalPayment* property), 38
 next_record() (*core.models.HistoricalRatePerDate* property), 40
 note (*core.models.Activity* attribute), 27
 note (*core.models.Appropriation* attribute), 31
 note (*core.models.Case* attribute), 33
 note (*core.models.Payment* attribute), 43
 number (*core.models.EffortStep* attribute), 35
- ## O
- objects (*core.models.AccountAlias* attribute), 24
 objects (*core.models.AccountAliasMapping* attribute), 25
 objects (*core.models.Activity* attribute), 27
 objects (*core.models.ActivityCategory* attribute), 28
 objects (*core.models.ActivityDetails* attribute), 30
 objects (*core.models.Appropriation* attribute), 31
 objects (*core.models.ApprovalLevel* attribute), 32
 objects (*core.models.Case* attribute), 33
 objects (*core.models.Effort* attribute), 35
 objects (*core.models.EffortStep* attribute), 35
 objects (*core.models.HistoricalCase* attribute), 37
 objects (*core.models.HistoricalPayment* attribute), 38
 objects (*core.models.HistoricalRatePerDate* attribute), 40
 objects (*core.models.InternalPaymentRecipient* attribute), 41
 objects (*core.models.Municipality* attribute), 41
 objects (*core.models.Payment* attribute), 43
 objects (*core.models.PaymentDateExclusion* attribute), 44
 objects (*core.models.PaymentMethodDetails* attribute), 44
 objects (*core.models.PaymentSchedule* attribute), 47
 objects (*core.models.RatePerDate* attribute), 50
 objects (*core.models.RelatedPerson* attribute), 51
 objects (*core.models.SchoolDistrict* attribute), 51
 objects (*core.models.Section* attribute), 52
 objects (*core.models.SectionInfo* attribute), 53
 objects (*core.models.ServiceProvider* attribute), 54
 objects (*core.models.TargetGroup* attribute), 55
 objects (*core.models.Team* attribute), 55
 objects (*core.models.VariableRate* attribute), 57
 ONE_TIME_PAYMENT (*core.models.PaymentSchedule* attribute), 45
 ongoing() (*core.managers.ActivityQuerySet* method), 22
 ongoing() (*core.managers.AppropriationQuerySet* method), 22
 ongoing() (*core.managers.CaseQuerySet* method), 22
 ordering (*core.admin.MainActivityDetailsInline* attribute), 13
 ordering (*core.admin.SupplementaryActivityDetailsInline* attribute), 17
- ## P
- pagination_class (*core.views.PaymentViewSet* attribute), 68
 paid (*core.models.HistoricalPayment* attribute), 38
 paid (*core.models.Payment* attribute), 43
 paid_amount (*core.models.HistoricalPayment* attribute), 39
 paid_amount (*core.models.Payment* attribute), 43
 paid_date (*core.models.HistoricalPayment* attribute), 39
 paid_date (*core.models.Payment* attribute), 43
 paid_date_or_date_case (*core.managers.PaymentQuerySet* attribute), 23
 paid_date_or_date_gte() (*core.managers.PaymentQuerySet* method), 23
 paid_date_or_date_lte() (*core.managers.PaymentQuerySet* method), 23
 paragraph (*core.models.Section* attribute), 52
 parse_account_alias_mapping_data_from_csv_path() (in module *core.utils*), 63
 parse_account_alias_mapping_data_from_csv_string() (in module *core.utils*), 63

paying_municipality (*core.models.Case attribute*), 33
 paying_municipality_id (*core.models.Case attribute*), 34
 Payment (*class in core.models*), 41
 Payment.DoesNotExist, 42
 Payment.MultipleObjectsReturned, 42
 payment_amount (*core.models.PaymentSchedule attribute*), 47
 payment_cost_choices (*core.models.PaymentSchedule attribute*), 47
 payment_cost_type (*core.models.PaymentSchedule attribute*), 47
 payment_date (*core.models.PaymentSchedule attribute*), 47
 payment_day_of_month (*core.models.PaymentSchedule attribute*), 47
 payment_frequency (*core.models.PaymentSchedule attribute*), 47
 payment_frequency_choices (*core.models.PaymentSchedule attribute*), 47
 payment_id (*core.models.PaymentSchedule attribute*), 47
 payment_id() (*core.admin.PaymentAdmin method*), 14
 payment_method (*core.models.Payment attribute*), 43
 payment_method (*core.models.PaymentSchedule attribute*), 47
 payment_method_details (*core.models.PaymentSchedule attribute*), 47
 payment_method_details_id (*core.models.PaymentSchedule attribute*), 47
 payment_plan (*core.models.Activity attribute*), 27
 payment_rate (*core.models.PaymentSchedule attribute*), 47
 payment_rate_id (*core.models.PaymentSchedule attribute*), 47
 payment_schedule (*core.models.Payment attribute*), 43
 payment_schedule (*core.models.Price attribute*), 48
 payment_schedule_id (*core.models.Payment attribute*), 43
 payment_schedule_id (*core.models.Price attribute*), 48
 payment_schedule_str() (*core.admin.PaymentAdmin method*), 14
 payment_type (*core.models.PaymentSchedule attribute*), 47
 payment_type_choices (*core.models.PaymentSchedule attribute*), 47
 payment_units (*core.models.PaymentSchedule attribute*), 47
 PaymentAdmin (*class in core.admin*), 13
 PaymentDateExclusion (*class in core.models*), 43
 PaymentDateExclusion.DoesNotExist, 44
 PaymentDateExclusion.MultipleObjectsReturned, 44
 PaymentDateExclusionAdmin (*class in core.admin*), 14
 PaymentFilter (*class in core.filters*), 21
 PaymentMethodDetails (*class in core.admin*), 14
 PaymentMethodDetails (*class in core.models*), 44
 PaymentMethodDetails.DoesNotExist, 44
 PaymentMethodDetails.MultipleObjectsReturned, 44
 PaymentMethodDetailsSerializer (*class in core.serializers*), 60
 PaymentMethodDetailsViewSet (*class in core.views*), 67
 PaymentQuerySet (*class in core.managers*), 22
 payments (*core.models.PaymentSchedule attribute*), 47
 PaymentSchedule (*class in core.models*), 44
 PaymentSchedule.DoesNotExist, 45
 PaymentSchedule.MultipleObjectsReturned, 45
 paymentschedule_set (*core.models.PaymentMethodDetails attribute*), 44
 paymentschedule_set (*core.models.Rate attribute*), 49
 PaymentScheduleAdmin (*class in core.admin*), 14
 PaymentScheduleFilter (*class in core.filters*), 21
 PaymentScheduleSerializer (*class in core.serializers*), 60
 PaymentScheduleViewSet (*class in core.views*), 68
 PaymentSerializer (*class in core.serializers*), 60
 PaymentViewSet (*class in core.views*), 68
 pays_for (*core.models.Municipality attribute*), 41
 per_payment_amount() (*core.models.PaymentSchedule property*), 47
 PER_UNIT_PRICE (*core.models.PaymentSchedule attribute*), 45
 perform_create() (*core.mixins.AuditModelViewSetMixin method*), 24
 perform_create() (*core.views.CaseViewSet method*), 66
 perform_update() (*core.mixins.AuditModelViewSetMixin method*), 24
 permission_classes (*core.views.AuditViewSet attribute*), 65
 permission_classes (*core.views.PaymentViewSet attribute*), 65

attribute), 68
 permission_classes (*core.views.ReadOnlyViewSet attribute*), 69
 PERSON (*core.models.PaymentSchedule attribute*), 45
 prev_record() (*core.models.HistoricalCase property*), 37
 prev_record() (*core.models.HistoricalPayment property*), 39
 prev_record() (*core.models.HistoricalRatePerDate property*), 40
 Price (*class in core.models*), 48
 price (*core.models.VariableRate attribute*), 57
 Price.DoesNotExist, 48
 Price.MultipleObjectsReturned, 48
 price_per_unit (*core.models.PaymentSchedule attribute*), 47
 PriceAdmin (*class in core.admin*), 15
 PriceForm (*class in core.admin*), 15
 PriceSerializer (*class in core.serializers*), 60
 PriceViewSet (*class in core.views*), 68
 profile (*core.models.User attribute*), 57
 profile_choices (*core.models.User attribute*), 57

Q

queryset (*core.views.ActivityCategoryViewSet attribute*), 64
 queryset (*core.views.ActivityDetailsViewSet attribute*), 64
 queryset (*core.views.ApprovalLevelViewSet attribute*), 65
 queryset (*core.views.CaseViewSet attribute*), 66
 queryset (*core.views.EffortStepViewSet attribute*), 66
 queryset (*core.views.EffortViewSet attribute*), 67
 queryset (*core.views.InternalPaymentRecipientViewSet attribute*), 67
 queryset (*core.views.MunicipalityViewSet attribute*), 67
 queryset (*core.views.PaymentMethodDetailsViewSet attribute*), 68
 queryset (*core.views.PaymentViewSet attribute*), 68
 queryset (*core.views.PriceViewSet attribute*), 69
 queryset (*core.views.RateViewSet attribute*), 69
 queryset (*core.views.RelatedPersonViewSet attribute*), 69
 queryset (*core.views.SchoolDistrictViewSet attribute*), 70
 queryset (*core.views.SectionInfoViewSet attribute*), 70
 queryset (*core.views.SectionViewSet attribute*), 70
 queryset (*core.views.ServiceProviderViewSet attribute*), 71
 queryset (*core.views.TargetGroupViewSet attribute*), 71
 queryset (*core.views.TeamViewSet attribute*), 71
 queryset (*core.views.UserViewSet attribute*), 72

R

Rate (*class in core.models*), 48
 rate (*core.models.HistoricalRatePerDate attribute*), 40
 rate (*core.models.RatePerDate attribute*), 50
 rate (*core.models.VariableRate attribute*), 57
 Rate.DoesNotExist, 48
 Rate.MultipleObjectsReturned, 49
 rate_amount() (*core.models.VariableRate property*), 57
 rate_and_price_changes (*core.models.User attribute*), 57
 rate_or_price_amount() (*core.models.PaymentSchedule property*), 47
 RateAdmin (*class in core.admin*), 15
 RateForm (*class in core.admin*), 15
 RatePerDate (*class in core.models*), 49
 RatePerDate.DoesNotExist, 49
 RatePerDate.MultipleObjectsReturned, 49
 RatePerDateInline (*class in core.admin*), 15
 RatePerDateSerializer (*class in core.serializers*), 60
 rates_per_date (*core.models.VariableRate attribute*), 57
 RateSerializer (*class in core.serializers*), 60
 RateViewSet (*class in core.views*), 69
 READONLY (*core.models.User attribute*), 56
 readonly_fields (*core.admin.ActivityAdmin attribute*), 10
 readonly_fields (*core.admin.PaymentAdmin attribute*), 14
 readonly_fields (*core.admin.PaymentScheduleAdmin attribute*), 15
 readonly_fields (*core.admin.PriceAdmin attribute*), 15
 readonly_fields (*core.admin.RatePerDateInline attribute*), 16
 ReadOnlyViewSet (*class in core.views*), 69
 ready() (*core.apps.CoreConfig method*), 19
 recalculate_prices() (*core.models.PaymentSchedule method*), 47
 recipient_choices (*core.models.PaymentSchedule attribute*), 47
 recipient_id (*core.models.Payment attribute*), 43
 recipient_id (*core.models.PaymentSchedule attribute*), 48
 recipient_name (*core.models.Payment attribute*), 43
 recipient_name (*core.models.PaymentSchedule attribute*), 48
 recipient_type (*core.models.Payment attribute*), 43
 recipient_type (*core.models.PaymentSchedule attribute*), 48

- `related_case` (*core.models.RelatedPerson* attribute), 51
 - `related_filters` (*core.filters.ActivityFilter* attribute), 19
 - `related_filters` (*core.filters.AllowedForStepsFilter* attribute), 20
 - `related_filters` (*core.filters.AppropriationFilter* attribute), 20
 - `related_filters` (*core.filters.CaseFilter* attribute), 20
 - `related_filters` (*core.filters.CaseForAppropriationFilter* attribute), 20
 - `related_filters` (*core.filters.CaseForPaymentFilter* attribute), 20
 - `related_filters` (*core.filters.PaymentFilter* attribute), 21
 - `related_filters` (*core.filters.PaymentScheduleFilter* attribute), 21
 - `related_filters` (*core.filters.UserForCaseFilter* attribute), 21
 - `related_persons` (*core.models.Case* attribute), 34
 - `RelatedPerson` (class in *core.models*), 50
 - `RelatedPerson.DoesNotExist`, 50
 - `RelatedPerson.MultipleObjectsReturned`, 50
 - `RelatedPersonAdmin` (class in *core.admin*), 16
 - `RelatedPersonSerializer` (class in *core.serializers*), 60
 - `RelatedPersonViewSet` (class in *core.views*), 69
 - `relation_type` (*core.models.RelatedPerson* attribute), 51
 - `required_fields_for_case` (*core.models.TargetGroup* attribute), 55
 - `required_fields_for_case_choices()` (*core.admin.TargetGroupForm* method), 18
 - `residence_municipality` (*core.models.Case* attribute), 34
 - `residence_municipality_id` (*core.models.Case* attribute), 34
 - `resident_clients` (*core.models.Municipality* attribute), 41
 - `revert_url()` (*core.models.HistoricalCase* method), 37
 - `revert_url()` (*core.models.HistoricalPayment* method), 39
 - `revert_url()` (*core.models.HistoricalRatePerDate* method), 40
 - `RUNNING_PAYMENT` (*core.models.PaymentSchedule* attribute), 45
 - `save()` (*core.models.Payment* method), 43
 - `save()` (*core.models.PaymentSchedule* method), 48
 - `save()` (*core.serializers.PaymentSerializer* method), 60
 - `save_model()` (*core.admin.VariableRateAdmin* method), 18
 - `save_payment_schedule_on_save_price()` (in module *core.signals*), 61
 - `save_without_historical_record()` (*core.models.Case* method), 34
 - `save_without_historical_record()` (*core.models.Payment* method), 43
 - `save_without_historical_record()` (*core.models.RatePerDate* method), 50
 - `saved_account_alias` (*core.models.HistoricalPayment* attribute), 39
 - `saved_account_alias` (*core.models.Payment* attribute), 43
 - `saved_account_string` (*core.models.Payment* attribute), 43
 - `sbsys_id` (*core.models.Appropriation* attribute), 31
 - `sbsys_id` (*core.models.Case* attribute), 34
 - `sbsys_template_id` (*core.models.SectionInfo* attribute), 53
 - `scaling_step` (*core.models.Case* attribute), 34
 - `scaling_step` (*core.models.HistoricalCase* attribute), 37
 - `SchoolDistrict` (class in *core.models*), 51
 - `SchoolDistrict.DoesNotExist`, 51
 - `SchoolDistrict.MultipleObjectsReturned`, 51
 - `SchoolDistrictAdmin` (class in *core.admin*), 16
 - `SchoolDistrictSerializer` (class in *core.serializers*), 60
 - `SchoolDistrictViewSet` (class in *core.views*), 70
 - `search_fields` (*core.admin.ActivityDetailsAdmin* attribute), 11
 - `search_fields` (*core.admin.EffortStepAdmin* attribute), 12
 - `search_fields` (*core.admin.MunicipalityAdmin* attribute), 13
 - `search_fields` (*core.admin.PaymentAdmin* attribute), 14
 - `search_fields` (*core.admin.PaymentDateExclusionAdmin* attribute), 14
 - `search_fields` (*core.admin.PaymentScheduleAdmin* attribute), 15
 - `search_fields` (*core.admin.SectionAdmin* attribute), 16
 - `search_fields` (*core.admin.ServiceProviderAdmin* attribute), 17
 - `Section` (class in *core.models*), 51
 - `section` (*core.models.Appropriation* attribute), 31
 - `section` (*core.models.SectionInfo* attribute), 53
- S**
- `saml_before_login()` (in module *core.utils*), 63
 - `saml_create_user()` (in module *core.utils*), 63
 - `save()` (*core.models.Activity* method), 27

Section.DoesNotExist, 52
 Section.MultipleObjectsReturned, 52
 section_id (*core.models.Appropriation* attribute), 31
 section_id (*core.models.SectionInfo* attribute), 53
 section_info (*core.models.AccountAlias* attribute), 24
 section_info() (*core.models.Appropriation* property), 31
 section_info_id (*core.models.AccountAlias* attribute), 24
 section_infos (*core.models.ActivityCategory* attribute), 28
 SectionAdmin (class in *core.admin*), 16
 SectionEffortStepProxyInline (class in *core.admin*), 16
 sectionId2name() (built-in function), 163
 SectionInfo (class in *core.models*), 52
 SectionInfo.DoesNotExist, 53
 SectionInfo.MultipleObjectsReturned, 53
 sectioninfo_set (*core.models.ActivityDetails* attribute), 30
 sectioninfo_set (*core.models.Section* attribute), 52
 SectionInfoActivityCategoryInline (class in *core.admin*), 16
 SectionInfoAdmin (class in *core.admin*), 17
 SectionInfoInline (class in *core.admin*), 17
 SectionInfoSerializer (class in *core.serializers*), 61
 SectionInfoViewSet (class in *core.views*), 70
 sections (*core.models.EffortStep* attribute), 35
 sections (*core.models.TargetGroup* attribute), 55
 SectionSerializer (class in *core.serializers*), 61
 SectionViewSet (class in *core.views*), 70
 send_activity_created_email() (in module *core.utils*), 63
 send_activity_created_email_on_paymentschedule (*core.models.PaymentSchedule* attribute), 71
 (in module *core.signals*), 61
 send_activity_deleted_email() (in module *core.utils*), 63
 send_activity_email() (in module *core.utils*), 63
 send_activity_expired_email() (in module *core.utils*), 63
 send_activity_payment_email_on_delete() (in module *core.signals*), 61
 send_activity_payment_email_on_save() (in module *core.signals*), 61
 send_activity_updated_email() (in module *core.utils*), 63
 send_appropriation() (in module *core.utils*), 63
 serializer_action_classes (*core.views.AppropriationViewSet* attribute), 65
 serializer_class (*core.views.ActivityCategoryViewSet* attribute), 64
 serializer_class (*core.views.ActivityDetailsViewSet* attribute), 64
 serializer_class (*core.views.ActivityViewSet* attribute), 64
 serializer_class (*core.views.ApprovalLevelViewSet* attribute), 65
 serializer_class (*core.views.CaseViewSet* attribute), 66
 serializer_class (*core.views.EffortStepViewSet* attribute), 66
 serializer_class (*core.views.EffortViewSet* attribute), 67
 serializer_class (*core.views.InternalPaymentRecipientViewSet* attribute), 67
 serializer_class (*core.views.MunicipalityViewSet* attribute), 67
 serializer_class (*core.views.PaymentMethodDetailsViewSet* attribute), 68
 serializer_class (*core.views.PaymentScheduleViewSet* attribute), 68
 serializer_class (*core.views.PaymentViewSet* attribute), 68
 serializer_class (*core.views.PriceViewSet* attribute), 69
 serializer_class (*core.views.RateViewSet* attribute), 69
 serializer_class (*core.views.RelatedPersonViewSet* attribute), 69
 serializer_class (*core.views.SchoolDistrictViewSet* attribute), 70
 serializer_class (*core.views.SectionInfoViewSet* attribute), 70
 serializer_class (*core.views.SectionViewSet* attribute), 70
 serializer_class (*core.views.ServiceProviderViewSet* attribute), 71
 serializer_class (*core.views.TargetGroupViewSet* attribute), 71
 serializer_class (*core.views.TeamViewSet* attribute), 71
 serializer_class (*core.views.UserViewSet* attribute), 72
 service_provider (*core.models.Activity* attribute), 27
 service_provider_id (*core.models.Activity* attribute), 27
 service_providers (*core.models.ActivityDetails* attribute), 30
 serviceplatformen_to_related_person() (*core.models.RelatedPerson* static method), 51
 ServiceProvider (class in *core.models*), 53
 ServiceProvider.DoesNotExist, 54
 ServiceProvider.MultipleObjectsReturned, 54

ServiceProviderAdmin (class in core.admin), 17
 ServiceProviderSerializer (class in core.serializers), 61
 ServiceProviderViewSet (class in core.views), 70
 set_needs_recalculation_on_save_rate() (in module core.signals), 61
 set_payment_id_on_paymentschedule_save() (in module core.signals), 61
 set_rate_amount() (core.models.VariableRate method), 57
 set_saved_account_alias_on_payment_save() (in module core.signals), 62
 set_saved_account_string_on_payment_save() (in module core.signals), 62
 setup_eager_loading() (core.serializers.ActivitySerializer static method), 58
 setup_eager_loading() (core.serializers.BaseAppropriationSerializer static method), 59
 setup_eager_loading() (core.serializers.CaseSerializer static method), 59
 setup_eager_loading() (core.serializers.PaymentScheduleSerializer static method), 60
 spinOff() (built-in function), 164
 spinOn() (built-in function), 164
 start_date (core.models.Activity attribute), 27
 start_date (core.models.HistoricalRatePerDate attribute), 40
 start_date (core.models.RatePerDate attribute), 50
 status (core.models.Activity attribute), 28
 status() (core.models.Appropriation property), 31
 strict_amount_sum() (core.managers.PaymentQuerySet method), 23
 suffix (core.views.ActivityDetailsViewSet attribute), 64
 suffix (core.views.ActivityViewSet attribute), 64
 suffix (core.views.AppropriationViewSet attribute), 65
 suffix (core.views.ApprovalLevelViewSet attribute), 65
 suffix (core.views.CaseViewSet attribute), 66
 suffix (core.views.EffortStepViewSet attribute), 66
 suffix (core.views.EffortViewSet attribute), 67
 suffix (core.views.InternalPaymentRecipientViewSet attribute), 67
 suffix (core.views.MunicipalityViewSet attribute), 67
 suffix (core.views.PaymentMethodDetailsViewSet attribute), 68
 suffix (core.views.PaymentScheduleViewSet attribute), 68
 suffix (core.views.PaymentViewSet attribute), 68
 suffix (core.views.PriceViewSet attribute), 69
 suffix (core.views.RateViewSet attribute), 69
 suffix (core.views.RelatedPersonViewSet attribute), 69
 suffix (core.views.SchoolDistrictViewSet attribute), 70
 suffix (core.views.SectionInfoViewSet attribute), 70
 suffix (core.views.SectionViewSet attribute), 70
 suffix (core.views.ServiceProviderViewSet attribute), 71
 suffix (core.views.TargetGroupViewSet attribute), 71
 suffix (core.views.TeamViewSet attribute), 71
 suffix (core.views.UserViewSet attribute), 72
 supplementary_activities (core.models.ActivityDetails attribute), 30
 supplementary_activities (core.models.Section attribute), 52
 supplementary_activities() (core.models.Appropriation property), 31
 supplementary_activity_for (core.models.ActivityDetails attribute), 30
 supplementary_activity_main_account_number (core.models.SectionInfo attribute), 53
 SupplementaryActivityDetailsInline (class in core.admin), 17
 SupplementaryActivityInline (class in core.admin), 17
 supplied_activities (core.models.ServiceProvider attribute), 54
 synchronize_payments() (core.models.PaymentSchedule method), 48

T

target_group (core.models.Case attribute), 34
 target_group_id (core.models.Case attribute), 34
 TargetGroup (class in core.models), 54
 TargetGroup.DoesNotExist, 54
 TargetGroup.MultipleObjectsReturned, 54
 TargetGroupAdmin (class in core.admin), 18
 TargetGroupForm (class in core.admin), 18
 TargetGroupSerializer (class in core.serializers), 61
 TargetGroupViewSet (class in core.views), 71
 tax_card (core.models.PaymentMethodDetails attribute), 44
 tax_card_choices (core.models.PaymentMethodDetails attribute), 44
 Team (class in core.models), 55
 team (core.models.User attribute), 57
 Team.DoesNotExist, 55
 Team.MultipleObjectsReturned, 55
 team_id (core.models.User attribute), 57
 TeamAdmin (class in core.admin), 18
 teamId2name() (built-in function), 164
 TeamSerializer (class in core.serializers), 61
 TeamViewSet (class in core.views), 71

testSpin() (*built-in function*), 164
 text (*core.models.Section attribute*), 52
 to_representation() (*core.serializers.TargetGroupSerializer method*), 61
 total_cost() (*core.models.Activity property*), 28
 total_cost_expected() (*core.models.Appropriation property*), 31
 total_cost_full_year() (*core.models.Activity property*), 28
 total_cost_granted() (*core.models.Appropriation property*), 31
 total_cost_this_year() (*core.models.Activity property*), 28
 total_expected_full_year() (*core.models.Appropriation property*), 31
 total_expected_this_year() (*core.models.Activity property*), 28
 total_expected_this_year() (*core.models.Appropriation property*), 31
 total_granted_full_year() (*core.models.Appropriation property*), 31
 total_granted_this_year() (*core.models.Activity property*), 28
 total_granted_this_year() (*core.models.Appropriation property*), 31
 triggers_payment_email() (*core.models.Activity property*), 28

U

update() (*core.serializers.PriceSerializer method*), 60
 upload_csv() (*core.admin.AccountAliasMappingAdmin method*), 10
 urls() (*core.admin.AccountAliasMappingAdmin property*), 10
 User (*class in core.models*), 55
 User.DoesNotExist, 56
 User.MultipleObjectsReturned, 56
 user_created (*core.mixins.AuditModelMixin attribute*), 23
 user_created (*core.models.HistoricalCase attribute*), 37
 user_modified (*core.mixins.AuditModelMixin attribute*), 23
 user_modified (*core.models.HistoricalCase attribute*), 37
 user_permissions (*core.models.User attribute*), 57
 UserForCaseFilter (*class in core.filters*), 21
 userId2name() (*built-in function*), 163
 users (*core.models.Team attribute*), 55
 UserSerializer (*class in core.serializers*), 61
 UserViewSet (*class in core.views*), 71

V

validate() (*core.serializers.ActivitySerializer method*), 58
 validate() (*core.serializers.CaseSerializer method*), 59
 validate() (*core.serializers.PaymentScheduleSerializer method*), 60
 validate() (*core.serializers.PaymentSerializer method*), 60
 validate_cvr() (*in module core.utils*), 63
 validate_expected() (*core.models.Activity method*), 28
 VariableRate (*class in core.models*), 57
 VariableRate.DoesNotExist, 57
 VariableRate.MultipleObjectsReturned, 57
 variablerate_ptr (*core.models.Price attribute*), 48
 variablerate_ptr (*core.models.Rate attribute*), 49
 variablerate_ptr_id (*core.models.Price attribute*), 48
 variablerate_ptr_id (*core.models.Rate attribute*), 49
 VariableRateAdmin (*class in core.admin*), 18
 VariableRateAdminForm (*class in core.admin*), 18
 vat_factor (*core.models.ServiceProvider attribute*), 54
 vat_factor() (*core.models.Activity property*), 28
 verbose_name (*core.admin.HistoricalRatePerDateInline attribute*), 13
 verbose_name (*core.admin.MainActivityDetailsInline attribute*), 13
 verbose_name (*core.admin.SectionEffortStepProxyInline attribute*), 16
 verbose_name (*core.admin.SectionInfoInline attribute*), 17
 verbose_name (*core.admin.SupplementaryActivityDetailsInline attribute*), 17
 verbose_name (*core.admin.SupplementaryActivityInline attribute*), 18
 verbose_name_plural (*core.admin.HistoricalRatePerDateInline attribute*), 13
 verbose_name_plural (*core.admin.MainActivityDetailsInline attribute*), 13
 verbose_name_plural (*core.admin.SectionEffortStepProxyInline attribute*), 16
 verbose_name_plural (*core.admin.SectionInfoInline attribute*), 17
 verbose_name_plural (*core.admin.SupplementaryActivityDetailsInline attribute*), 17

`verbose_name_plural`
(*core.admin.SupplementaryActivityInline*
attribute), [18](#)

W

`weekday()` (*core.admin.PaymentDateExclusionAdmin*
method), [14](#)
`WEEKLY` (*core.models.PaymentSchedule* *attribute*), [45](#)
`WORKFLOW_ENGINE` (*core.models.User* *attribute*), [56](#)
`write_prism_file_v0()` (*in module core.utils*), [63](#)