
ZAC

Release 0.12.2

Maykin Media, Gemeente Utrecht

Jul 06, 2020

CONTENTS:

- 1 Configuration 3**
 - 1.1 Configuring the services 3
 - 1.2 Subscribing to notifications 4
- 2 Available forms 5**
 - 2.1 Camunda forms 5
 - 2.2 Open Forms integration 5
- 3 Indices and tables 7**

Welcome to the documentation for the zaakafhandelcomponent (ZAC).

CONFIGURATION

After deploying the ZAC, you need some runtime configuration. An administrator can do this.

1.1 Configuring the services

The ZAC does not copy data, but reads it from the external APIs. These APIs need to be configured.

Most API's are added as services under **ZGW_Consumers > Services** in the admin interface.

1.1.1 API's for Zaakgericht werken

Add the following services:

- Catalogi API
- Zaken API
- Documents API
- Notifications API

You will need a client ID and secret for each of those, consult the administrator for each API for those.

1.1.2 Camunda process engine

Navigate to **Camunda configuration** in the admin and fill out the details.

1.1.3 Kadaster configuration

If you make use of BAG objects, you'll need to configure the Kadaster APIs. Navigate to **Kadasterconfiguratie** in the admin and verify/fill out the fields.

1.1.4 Kownsl

Kownsl manage advices and approvals. Add a **Service** for it:

- Type: ORC (Overige)
- Authorization type: API Key
- Header key: Authorization
- Header value: Token <insert kownsl token>

1.1.5 Open Forms

Open Forms is a form builder engine, for which the ZAC has some basic support at the moment. Add a **Service** for it:

- Type: ORC (Overige)
- Authorization type: API Key
- Header key: Authorization
- Header value: Token <insert open forms token>

1.2 Subscribing to notifications

The ZAC is aimed at performance, and for that reason, results of API calls are cached. Certain events require cache invalidation, and these events are received from the Notifications API.

After installation and configuration of the servers, run the following command in a container:

```
src/manage.py subscribe_notifications https://zac.gemeente.nl
```

This will set up the ZAC to receive notifications sent from the other APIs and act accordingly.

AVAILABLE FORMS

The ZAC presents some (dynamically generated) forms to its users.

2.1 Camunda forms

The ZAC handles two distinct flavours of form definitions: form key based, or form definition based. If a form key is present for a task, then this will be used. If it's absent, then the form definition is used.

2.1.1 Form keys

A Camunda user task can contain a form key reference. The following form keys are implemented by the ZAC:

- `zac:configureAdviceRequest`: presents a form to the end user to select documents from the case to create an advice request, and allows selection of assignees
- `zac:configureApprovalRequest`: presents a form to the end user to select documents from the case to create an approval request, and allows selection of assignees
- `zac:doRedirect`: grabs the `redirectTo` process variable, and redirects the user to this location. A `?state` parameter is added for the receiving application, which is consumed when the external application redirects the user back to the ZAC.

2.1.2 Form definition

Camunda allows a user task to contain a simple form definition with primitive fields. These fields are implemented in the ZAC, and if such a form definition is present, the ZAC renders a form for the user to fill out. Upon submission, the user task receives the field values as process variables and the task is marked as completed.

2.2 Open Forms integration

WIP

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`