

Toggleing

Agenda

As an SDC user, I would like to have the ability to enable and disable certain features during development.

How it works

First, we need to add a new feature label into ToggleableFeature.class

```
@Label("My feature")  
MY_FEATURE;
```

For every section of code we would like to control using the toggling of this specific feature – we should add a condition –

```
if(ToggleableFeature.MY_FEATURE.isActive()) {  
    // do something  
}
```

The toggling manager checks if this feature is enabled, and if it does – it runs the code.

(note – the default state if a feature is enable = false)

Current features

Right now, we have toggle for

org.openecomp.sdc.be.dao.impl.heal.HealJanusGraphDao#healGraphVertex

For healing feature (using ToggleableFeature.*HEALING*)

How to use

There are two ways to change a feature state – togglz UI and REST calls

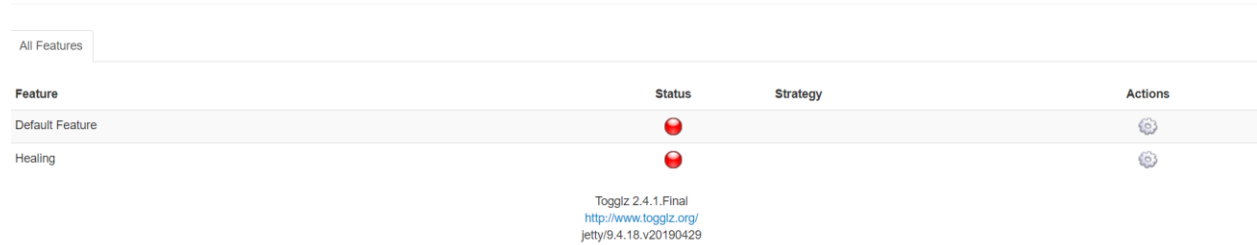
UI

Go into `<your_env_ip>:8080/catalog/togglz`





Once loaded, you will see all of the features that defined under `ToggleableFeature.java`

Togglz

/



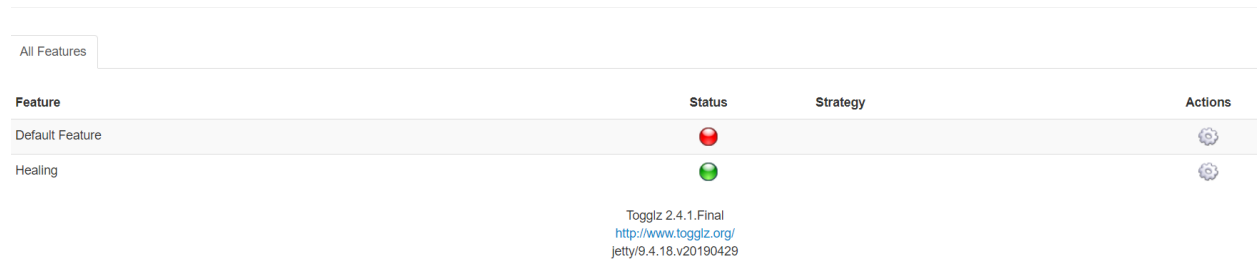
The screenshot shows the Togglz web interface. At the top, there is a navigation bar with a tab labeled "All Features". Below this is a table with the following columns: "Feature", "Status", "Strategy", and "Actions". The table contains two rows: "Default Feature" and "Healing". Both features have a red status indicator. The "Actions" column for each feature contains a gear icon. Below the table, there is a footer with the text: "Togglz 2.4.1.Final", "http://www.togglz.org/", and "jetty/9.4.18.v20190429".

| Feature | Status | Strategy | Actions |
|-----------------|---|----------|---|
| Default Feature |  | |  |
| Healing |  | |  |





To change a feature's status – press on "Actions" and choose "enable" option.
After that – the feature is enabled.

Togglz

/



The screenshot shows the Togglz web interface after the "Healing" feature has been enabled. The table structure is the same as in the previous screenshot, but the status indicator for "Healing" is now green. The "Default Feature" remains disabled (red status indicator). The "Actions" column for each feature still contains a gear icon. The footer text remains the same: "Togglz 2.4.1.Final", "http://www.togglz.org/", and "jetty/9.4.18.v20190429".

| Feature | Status | Strategy | Actions |
|-----------------|---|----------|---|
| Default Feature |  | |  |
| Healing |  | |  |

REST

The following REST calls access toggle features and states

GET /v1/catalog/toggle

Gets all toggle features and its states

```
{
    MY_FEATURE_1: false,
    MY_FEATURE_2: true,
    ...
}
```

GET /v1/catalog/toggle/{featureName}/state

Get state of toggle feature, which its name is featureName

PUT /v1/catalog/toggle/state/{state}

Update states of all toggle feature to a given state

For example –

Calling to /v1/catalog/toggle/state/true, will enable all features, and when we get all features states, we will see

```
{
    MY_FEATURE_1: true,
    MY_FEATURE_2: true
}
```

PUT /v1/catalog/toggle/{featureName}/state/{state}

Change feature state, for the feature which its name is featureName.

For example –

If we had two features, which its state was

```
{
    MY_FEATURE_1: true,
    MY_FEATURE_2: true
}
```

Calling to this REST call for MY_FEATURE_1 will do the following –

/v1/catalog/toggle/MY_FEATURE_1/state/false →

```
{
    MY_FEATURE_1: false,
    MY_FEATURE_2: true
}
```

And the condition `ToggleableFeature.MY_FEATURE_1.isActive()` will be false.

Note – when the BE docker is getting restarted – all of the features states are set to “false”, meaning – everytime that the docker retsarts, you will need to set the wanted features to “true”