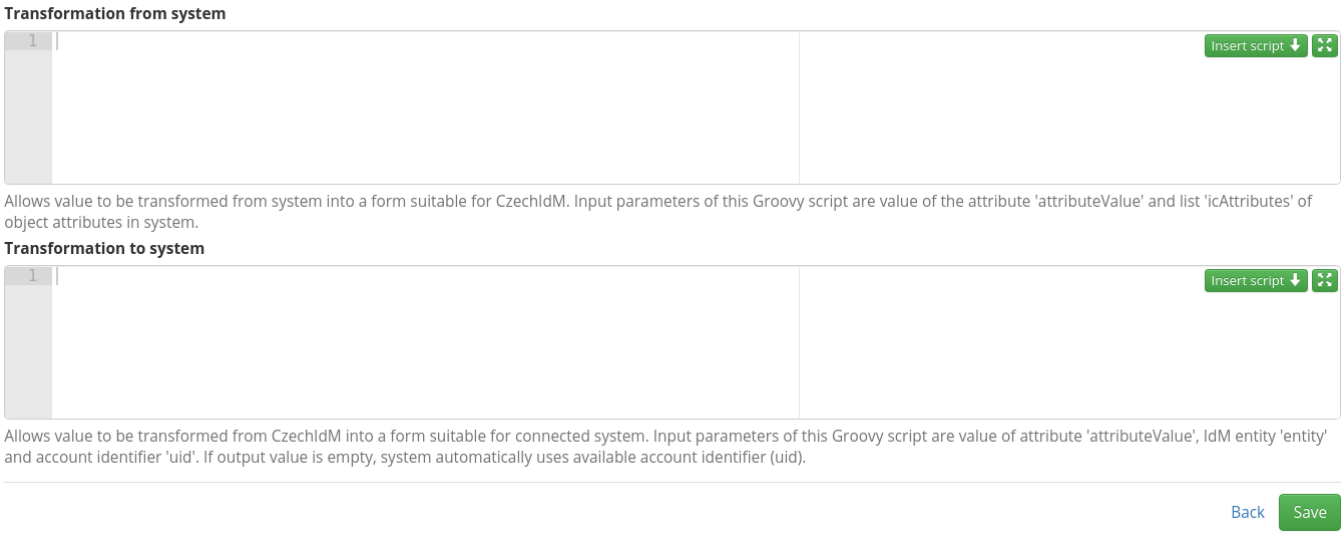
Transformation scripts - library and usage

Assuming you'd connected a managed system or source system to CzechIdM, prepared synchronization or provisioning with some attribute mappings (e.g. identity), you may then have several attributes that differ in format between the source and the managed system. For instance, you want to fill an Active Directory's attribute *diplayName* whose format is <firstName + LastName>. For scripts, we currently use Groovy language.

displayName example

From the HR system to CzechIdM you provide **firstName** and **lastName** as separate attributes as is common. Now you want to make a transformation script to fill **displayName** from 2 other attributes.

1. First of all, you must have the attribute displayName defined in provisioning mapping. In the detail of the attribute mapping you do not fill the **IdM Key**, since the transformation from multiple attributes will be used instead
2. At the bottom of the detail page, you can see two boxes where you can write your transformation scripts. The first box is used for a transformation **from the system**, i.e. to transform attribute values form the connected system to CzechIdM. That is not our case now, we only want a provisioning transformation. The second box serves for a transformation from CzechIdM **to the connected system**. In it, you will place your script.



To fill the attribute from 2 entity (identity in our case) attributes, just use this piece of code:

```
entity.firstName + " " + entity.lastName
```

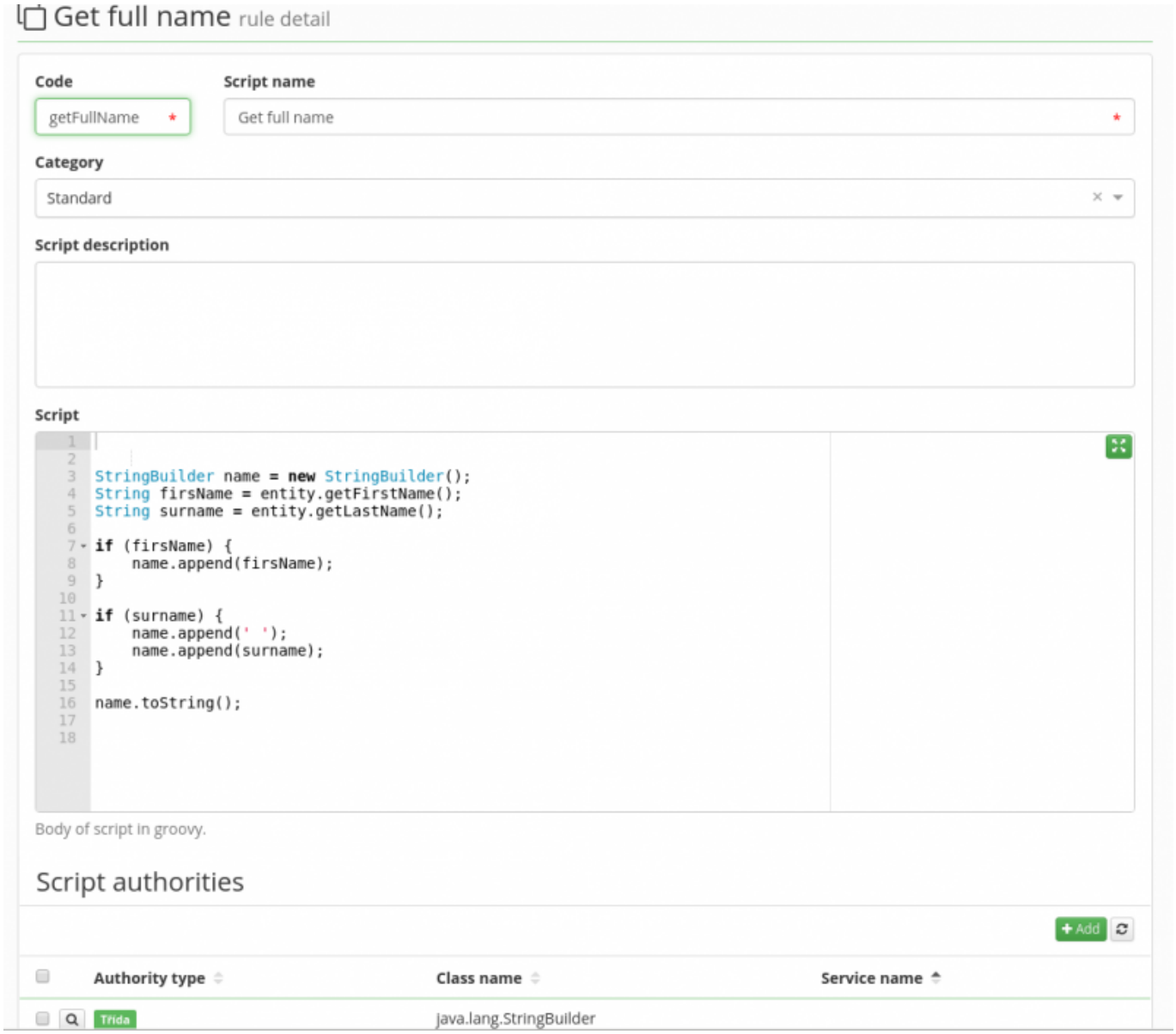
Thus you can define basic transformation scripts that does not need any additional privilege to run.

The scripts definition library

There is another convenient way to use transformation scripts. You can define your script in CzechIdM via menu **Settings → Script definition**. There you can see the list of all available scripts (not only transformation ones). If you click on the **Add** button, you can now add your own script definition to the CzechIdM script library.

Add a script to the library

If you click on **Add** button, you can add your own script definition to the CzechIdM script library.



Fill the following fields:

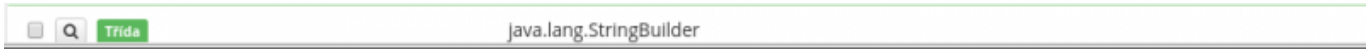
- **Code** - the name of the script usually used when calling from other scripts
- **Name** - the name visible in lists in GUI
- **Category** - You can choose from the select box the category by intended use. The category has

both informative and restrictive character. If you select the category of the script as "Transformation to", it will be available as a choice in the attribute mapping detail form in the *Transformation to* box. If you use "Standard", it will be available in all boxes, but you must be careful when using the script and optionally add some parameters.

- **Script description** - optional - visible on some forms
- **Script** - the definition of the script itself
- **Script authorities** - see the section below

Script authorities

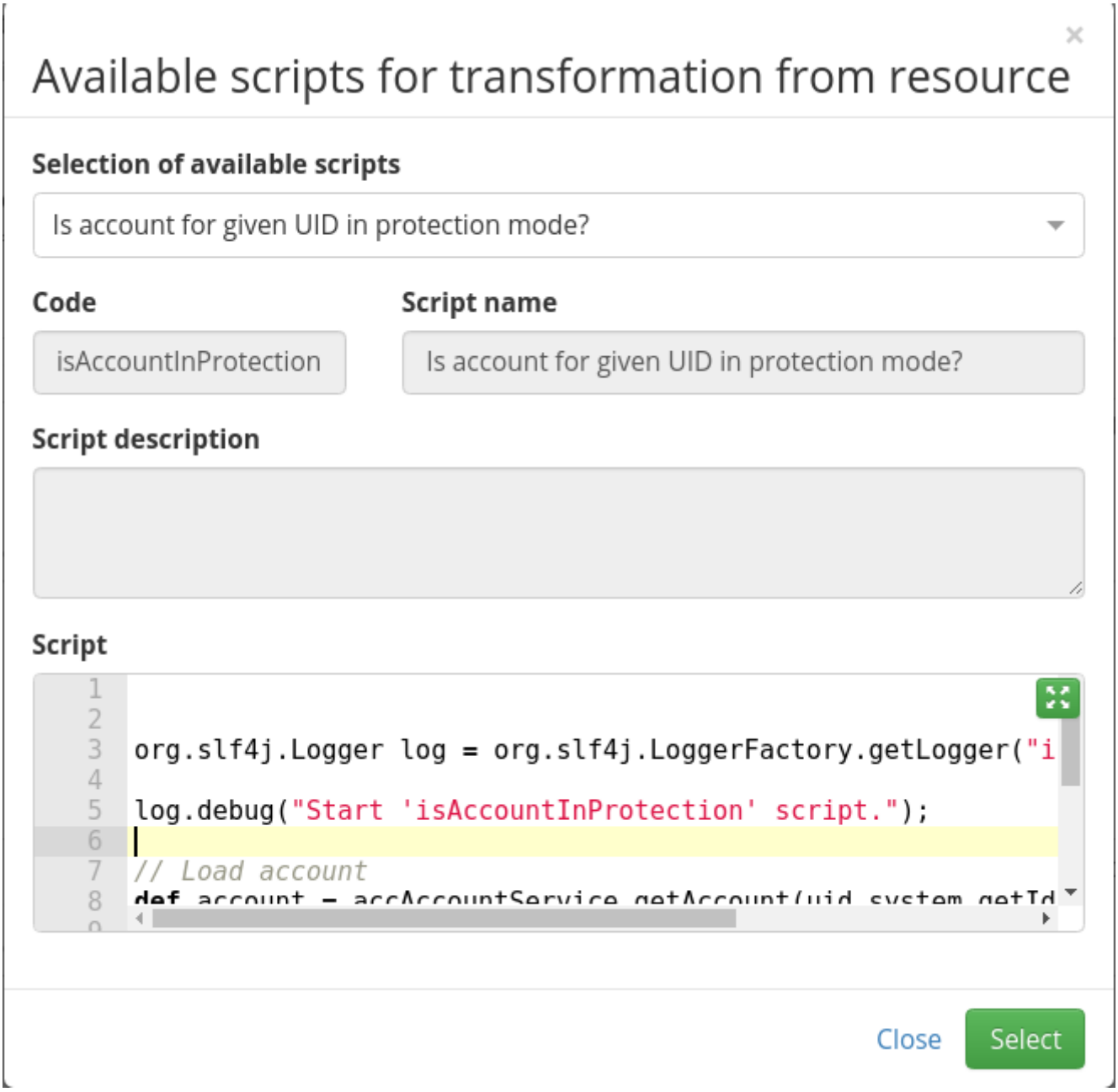
In this table, you specify what authorizations on data you give to the script. There are some basic rules, that every script has by default - e.g. get attributes and their values from entity. Other rules must be added if you want to access other entities such as Roles or Organizations from the script that is used in Identities synchronization. Another common use case is that you want to use some non basic java Classes or Methods. Again, you have to give your script specific authorization to use that Class.



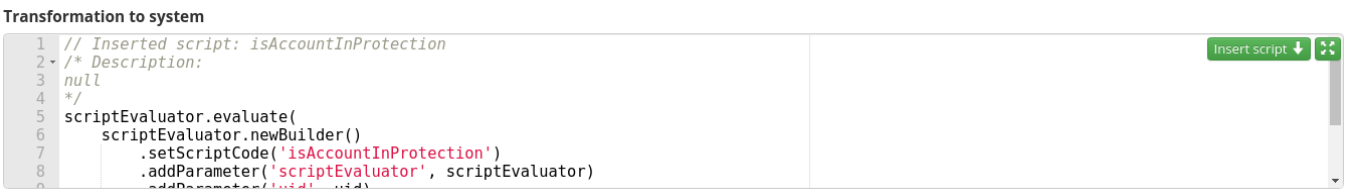
A library script use

The script is then available in synchronization and provisioning mapping in attribute detail. To use it, click on the green button **insert script** in the right upper corner of transformation box.

Then you can see a new window listing available scripts



When you select the desired one, click on the **Select** button. The script is then referred/called from the attribute mapping transformation box



If the selected script is from the **Standard** category, some parameters will not be generated automatically when inserting the script, so you must add them manually.

E.g. if you want to insert the script "getTrimmedString" as the **transformation to** script, the transformation box should contain the following code:

```
scriptEvaluator.evaluate(
```

```
    scriptEvaluator.newBuilder()
        .setScriptCode('getTrimmedString')
        .addParameter('scriptEvaluator', scriptEvaluator)
        .addParameter('uid', uid)
        .addParameter('attributeValue', attributeValue)
        .addParameter('entity', entity)
        .addParameter('system', system)
    .build());
```

If you want to insert the script "getTrimmedString" as the **transformation from** script, the transformation box should contain the following code:

```
scriptEvaluator.evaluate(
    scriptEvaluator.newBuilder()
        .setScriptCode('getTrimmedString')
        .addParameter('scriptEvaluator', scriptEvaluator)
        .addParameter('attributeValue', attributeValue)
        .addParameter('icAttributes', icAttributes)
        .addParameter('system', system)
    .build());
```

Script as a file

In fact, all default scripts that are available in GUI after CzechIdM installation, were loaded into application during its previous start. They meet XML format and file incorporates the script body (groovy), script privileges and its purpose. So if you want to track changes on your scripts e.g. with git, this is the best way.

We strongly recommend that you use scripts in files, since they can be easily upgraded with an application upgrade

Tips and tricks

- In case of provisioning the format of the attribute in transfomation script is derived from the schema of the target system. **Example**: the attribute is set as multivalued on schema then transformation script works with attributeValue as an Array even though you fill it from entity or EAV which is single valued.

Read more

- If you want to create a new script, please follow Transformation scripts - how to write a script
- Groovy Scripts Tips & Tricks