

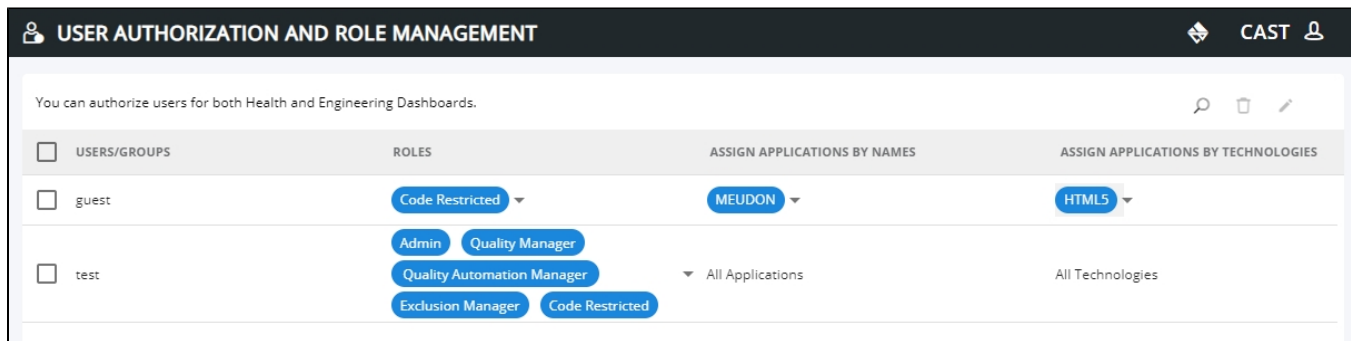
Changing the name of the cast_dashboards schema

- [Introduction](#)
- [Step 1 - Update application.properties](#)
- [Step 2 - Update schema.sql](#)
- [Step 3 - Restart the web application](#)

Introduction

A **graphical user interface** was implemented in 2.1 for managing the assignment of **role and data authorizations** to users and groups of users. This interface replaces the existing mechanism provided by the **roles.xml** and the **authorizations.xml** files. You can find out more in:

- [User roles](#) and [User roles - 2.x and above](#)
- [Data authorization](#) and [Data authorization - 2.x and above](#)



This management interface relies on a **schema hosted on a CAST Storage Service/PostgreSQL instance** to store the roles/data authorizations. This schema is called **cast_dashboards** by default and will be created on first startup of the web application. This schema is defined using two files:

```
2.x ZIPs
<unpacked_zip>\configurations\application.properties
<unpacked_zip>\configurations\schema.sql

2.x WARs
CATALINA_HOME\webapps\<deployed_war>\WEB-INF\classes\application.properties
CATALINA_HOME\webapps\<deployed_war>\WEB-INF\classes\schema.sql
```

If you want to **change the name of the schema**, for example if you want to host one roles/authorizations schema for each deployed dashboard on the same CAST Storage Service/PostgreSQL instance (due to requiring different roles/authorizations in each dashboard) you can do so as explained below.

Step 1 - Update application.properties

In the deployed dashboard, locate the following file:

```
2.x WARs:
CATALINA_HOME\webapps\<deployed_war>\WEB-INF\classes\application.properties

2.x ZIPs:
<unpacked_zip>\configurations\application.properties
```

Locate the following section:

```
#datasource configuration for user management
spring.datasource.url=jdbc:postgresql://localhost:2282/postgres?currentSchema=cast_dashboards
spring.datasource.platform=postgres
spring.datasource.username=operator
spring.datasource.password=CastAIP
spring.datasource.initialization-mode=always
spring.datasource.driver-class-name=org.postgresql.Driver
spring.liquibase.change-log=classpath:db/changelog/db.changelog-master.xml
spring.liquibase.default-schema=cast_dashboards
spring.liquibase.enabled=true
```

Change the following lines to modify the schema name:

- `spring.datasource.url=jdbc:postgresql://localhost:2282/postgres?currentSchema=cast_dashboards`
- `spring.liquibase.default-schema=cast_dashboards`

For example to call your schema "**user_management**":

```
#datasource configuration for user management
spring.datasource.url=jdbc:postgresql://localhost:2282/postgres?currentSchema=user_management
spring.datasource.platform=postgres
spring.datasource.username=operator
spring.datasource.password=CastAIP
spring.datasource.initialization-mode=always
spring.datasource.driver-class-name=org.postgresql.Driver
spring.liquibase.change-log=classpath:db/changelog/db.changelog-master.xml
spring.liquibase.default-schema=user_management
spring.liquibase.enabled=true
```

Save the file before proceeding.

Step 2 - Update schema.sql

In the deployed dashboard, locate the following file:

```
2.x WARs:
CATALINA_HOME\webapps\
```

Locate the following line:

```
CREATE SCHEMA IF NOT EXISTS cast_dashboards;
```

Change the name of the schema to match what you have entered in the **application.properties** file, for example to call your schema "**user_management**":

```
CREATE SCHEMA IF NOT EXISTS user_management;
```

Save the file before proceeding.

Step 3 - Restart the web application

Restart the web application so that changes are taken into account. The new schema will be created on the target CAST Storage Service/PostgreSQL instance.