

Knowage for dashboards - User Manual

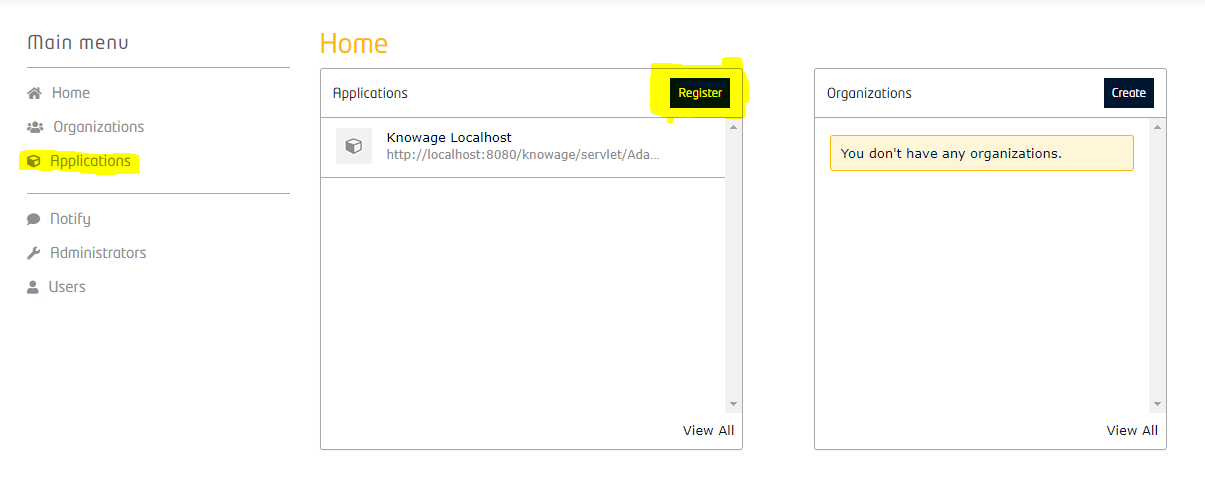
Welcome to the user manual section, here we describe a step by step guide to setup and configure the Knowage for dashboard instance.

# Contents

* ACS application registration
* Comunication check between VM and ACS
* Setup local instance
* Dataset configuration
* Dashboard execution
* Dashboards view by role

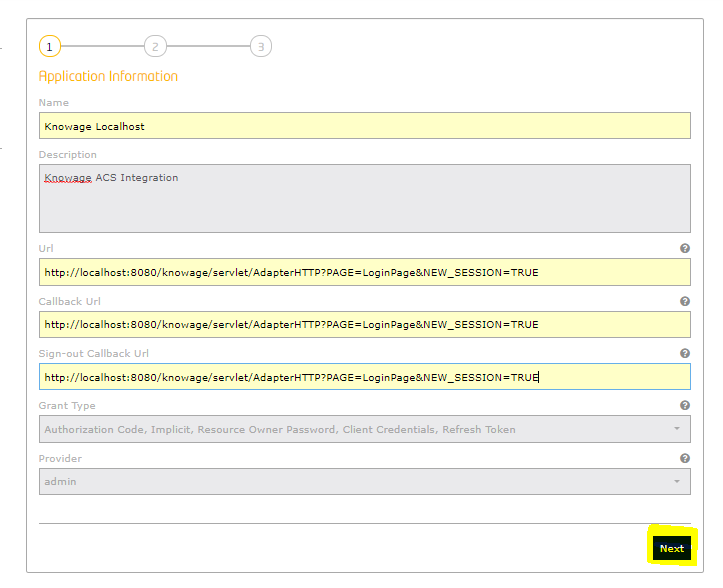
# ACS application registration

1. Sign-up on ACS (follow this guide: **Demeter user signup and activate**)
2. Sign-in to access the Home page
3. Into the Home page, simply click the **Register** button or click **applications** on main menu and then the **Register** button.

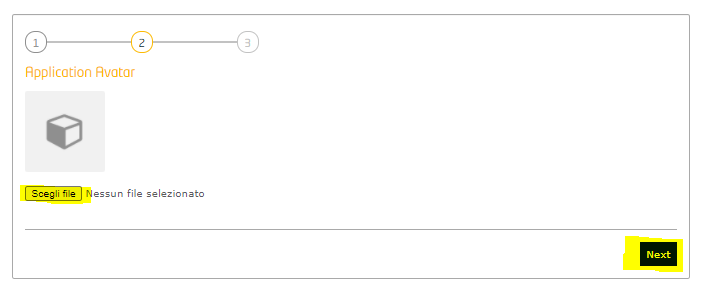


1. A three-step wizard will appear. On the first step we need to set the following fields:

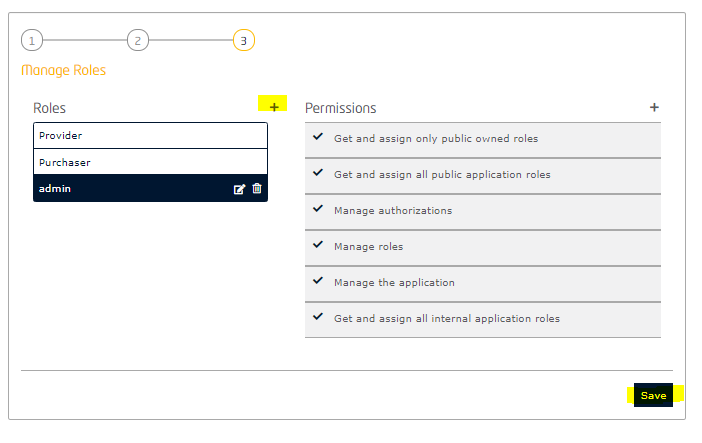
* **Name**: Name of the application
* **Description**: Some description
* **Url**: Application url, for a local installation of Knowage set it as:
  + http://localhost:8080/knowage/servlet/AdapterHTTP?PAGE=LoginPage&NEW\_SESSION=TRUE
* **Callback Url**: The user agent will be redirected to this URL when OAuth flow is finished.
  + http://localhost:8080/knowage/servlet/AdapterHTTP?PAGE=LoginPage&NEW\_SESSION=TRUE
* **Sign-out Callback Url**:
  + http://localhost:8080/knowage/servlet/AdapterHTTP?PAGE=LoginPage&NEW\_SESSION=TRUE



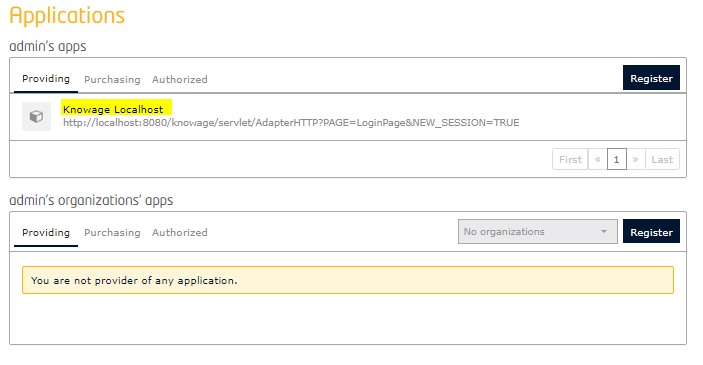
1. Once done, click the **Next** button to move on second step.
2. Here it's possible to choose an icon image. Once done click the **Next** button.

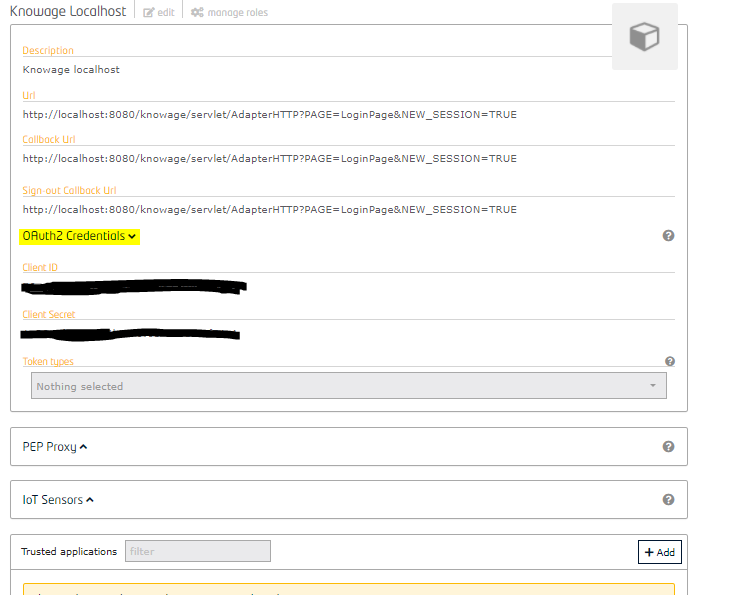


1. In the third step we can set up roles and permissions
2. Click the "**+**" icon on the roles section and add an **admin** role.

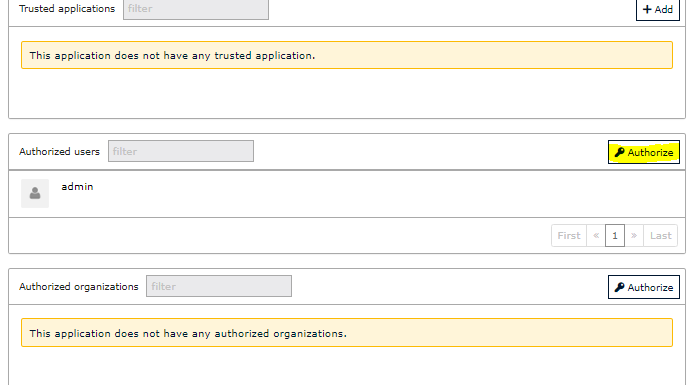


1. Click **Save** to complete.
2. Now click on the application to open the details, from there it's possible to **edit** or **manage roles**.

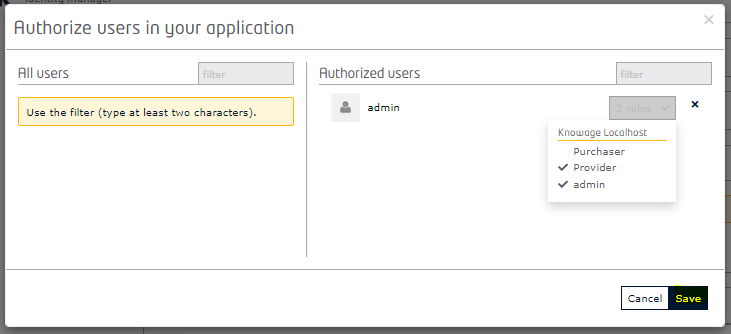




1. Click on **OAuth2 Credentials** and take note of the **Cliend ID** and **Client Secret** tokens
2. Scroll down the page and click the **Authorize** button on **Authorized users** section.



1. Here it's possible to select the ACS users to grant them access and roles to use our application.



# Comunication check between VM and ACS

Before proceeding to the next steps, it is recommended to verify the correct communication between the docker Virtual Machine (VM) and ACS by running the following command:

|  |
| --- |
| curl --location --request POST 'https://acs.bse.h2020-demeter-cloud.eu:5443/v1/auth/tokens' --header 'Content-Type: application/json' --data ' {  "name": "ACS\_MAIL",  "password": "ACS\_PASSWORD"  }' |

Put your ACS account mail on **ACS\_MAIL** and your ACS password on **ACS\_PASSWORD**.

If this test is not successful, then you must check with the system administrator that port **5443** is open and can connect to the ACS host.

# Setup local instance

1. Visit the docker hub page and check for the latest tag version of the instance:

* Knowage
* Knowagedb

1. Create a **new folder** and name it as your preference.
2. Now create a new file inside that folder and rename it as **docker-compose.yml**
3. Copy and paste the content below into the docker-compose file.

Note: remember to check and change the image tag version with the latest one:

* **image: demeterengteam/knowagedb-7.2.0: v2.0**
* **image: demeterengteam/knowage-7.2.0: idm-v2.0**

*docker-compose.yml*:

|  |
| --- |
| version: "3"  services:  knowagedb:  hostname: knowagedb  image: demeterengteam/knowagedb-7.2.0:v2.0  networks:  - hostnet  ports:  - "3306:3306"  environment:  - MYSQL\_ROOT\_PASSWORD=r00t  - MYSQL\_DATABASE=knowagedb  volumes:  - ./mariadb\_conf:/home/knowage/mariadb\_conf  knowage:  hostname: knowage  image: demeterengteam/knowage-7.2.0:idm-v2.0  networks:  - hostnet  ports:  - "8080:8080"  depends\_on:  - knowagedb  environment:  - DB\_HOST=knowagedb  - DB\_PORT=3306  - DB\_USER=root  - DB\_PASS=r00t  - DB\_DB=knowagedb  env\_file:  - ./KnowageParameters.env  networks:  hostnet: |

1. Now create a new file into the same folder and rename it as **KnowageParameters.env** (or as your preference).

* Note: If a different name is given, then remember to update also the **env\_file** reference into the **docker-compose.yml**.

1. Copy and paste the content below into the KnowageParameters file.

* Note: The acronym **"IDM"** (Identity Manager) is used to indicate the **Access Control System (ACS)**.

*KnowageParameters.env content:*

|  |
| --- |
| ACS\_INTEGRATION=true  SECURITY\_LOGOUT\_URL=https://acs.bse.h2020-demeter-cloud.eu:5443/auth/logout?\_method=DELETE  OAUTH2\_CLIENT\_ID=INSERT APP CLIENT ID FROM IDM  OAUTH2\_SECRET=INSERT APP CLIENT SECRET FROM IDM  ACS\_BASE\_URL=https://acs.bse.h2020-demeter-cloud.eu:5443  ACS\_ADMIN\_ID=ADMIN ID  ACS\_ADMIN\_EMAIL=ADMIN EMAIL  ACS\_ADMIN\_PASSWORD=ADMIN PASSWORD  REDIRECT\_URL=http://localhost:8080/knowage/servlet/AdapterHTTP?PAGE=LoginPage\&NEW\_SESSION=TRUE  PROXY\_HOST=  PROXY\_PORT= |

1. On **OAUTH2\_CLIENT\_ID** and **OAUTH2\_SECRET**, paste the tokens from your Knowage application registered on ACS.

* Note: Refer to **point 11** of **ACS application registration**

1. Set your **ACS Admin account ID**, **e-mail** and **password** on **ACS\_ADMIN\_ID**, **ACS\_ADMIN\_EMAIL** and **ACS\_ADMIN\_PASSWORD** fields.
2. Set the **REDIRECT\_URL** with the **Callback Url** provided on your Knowage application registered on ACS.
3. If you need to configure a proxy, set **PROXY\_HOST** and **PROXY\_PORT** else, leave those values empty.
4. Once everything is set, open a **command prompt** on the same folder of the **docker-compose** file.
5. If it's the first installation of the instance, or you want to update the version, then execute the command:

docker-compose up –d

else, execute the command

docker-compose start

1. Wait until the service is fully started (it should take 1 minute or less).
2. Open a browser and paste the URL for logging in Knowage:

http://localhost:8080/knowage/servlet/AdapterHTTP?PAGE=LoginPage&NEW\_SESSION=TRUE

1. Use the ACS account registered to access.
2. If it's the first time, an **Authorization** message will pop out to **Allow** the access.
3. Execute the command docker-compose stop to stop the instance. That's wont delete your data from the container.

* Note: Using again docker-compose up -d to start the instance , will reset all the data and configuration made inside Knowage.

# DSS Table

The following table shows, for each Dashboard (DSS) , the name, which pilots use it and the datasets name that feed it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | **DSS** | **DSS Name** | **Pilot** | **Datasets Name** |
| A1 | DSS1.A1 | A1 Plant Yield Estimation | 1.1, 1.2, 1.3, 1.4, 3.4 | 4A1.DSS1.FieldBordersLayer, 4A1.DSS1.PredictedFieldValueLayer |
| A2 | DSS2.A2 | A2 Plant Phenology Estimation | 3.1 | 4A2\_PhenologyEstimation |
| A3 | DSS3.A3 | A3 Plant Stress Detection | 1.4 | 4A3\_PlantStressData |
| A5 | DSS5.A5 | A5 Estimate Beehive | 5.3 | 4A5\_EstimateBeehive |
| B1 | DSS1.B1 | B1 Irrigation Managemen | 1.1, 1.2, 1.3, 3.1, 3.2 | 4B1\_IrrigationManagement |
| C1 | DSS1.C1 | C1 Nitrogen Balance Model | 1.3, 1.4, 2.3, 3.1, 3.2 | 4C1\_NitrogenBalanceData |
| C2 | DSS1.C2.D1 | C2 Nutrient and Sowing  recommendation | 1.3, 1.4, 2.3, 3.1, 3.2 | 4C2\_NutrientMonitorZones,  4C2\_NutrientMonitorWeather, 4C2\_NutrientMonitorParcel |
| D1 | DSS1.D1 | Emission | 2.1 | N/A |
| D2 | DSS1.D2 | D2 Field Operation | 2.3, 5.1, 5.2 | 4D2\_FieldOperation |
| D3 | DSS1.D3 | D3 Variable Rate | 3.4, 5.1, 5.2 | 4D3\_VariableRate |
| E1 | DSS1.E1 | E1 Pest Estimation with Sterile Fruit Flies | 1.1, 1.2, 3.3, 5.1, 5.3 | 4E1\_PestEstimation |
| E2 | DSS1.E2 | E2 Estimate Temperature-Related Pest Events | 3.1, 3.3 | 4E2\_TempPestEvents |
| F1 | DSS1.F1 | F1 Estimate Milk Production | 4.1, 4.2, 4.4, 5.4 | F1\_EstimateMilkProduction |
| F2 | DSS1.F2 | F2 Poultry Feeding | 4.4, 5.4 | 4F2\_PoultryFeeding |
| G1 | DSS1.G1.D1 | G1 Estimate Animal Welfare Condition Training | 4.2, 4.3, 4.4, 5.4 | 4G1.DSS1.AnimalWelfareTraining, 4G1.DSS1.AnimalWelfareMetrics |
| G1 | DSS1.G1.D2 | G1 Estimate Animal Welfare Condition Prediction | 4.2, 4.3, 4.4, 5.4 | 4G1.DSS2.AnimalWelfarePrediction |
| G2 | DSS1.G2 | G2 Poultry Well Being | 4.3, 5.4 | 4G2\_PoultryWellBeing |
| H1 | DSS1.H1.D1 | H1 Milk Quality Training | 4.2 | 4H1.DSS1.MilkQualityMetrics, 4H1.DSS1.MilkQualityTraining |
| H1 | DSS1.H1.D2 | H1 Milk Quality Prediction | 4.2 | 4H1.DSS2.MilkQuality Prediction |
| H2 | DSS1.H2 | H2 Transport Condition | 5.1, 5.4 | 4H2\_TransportCondition |
| I1 | DSS1.I1 | I1 Generic Farm Comparison | ALL | 4I1\_GenericFarmComparison |
| I2 | DSS1.I2 | I2 Neighbour Benchmarking | ALL | 4I2\_NeighbourBenchmarking |
| I3 | DSS1.I3 | I3 Technology Benchmarking | ALL | 4I3\_TechnologyBenchmarking |

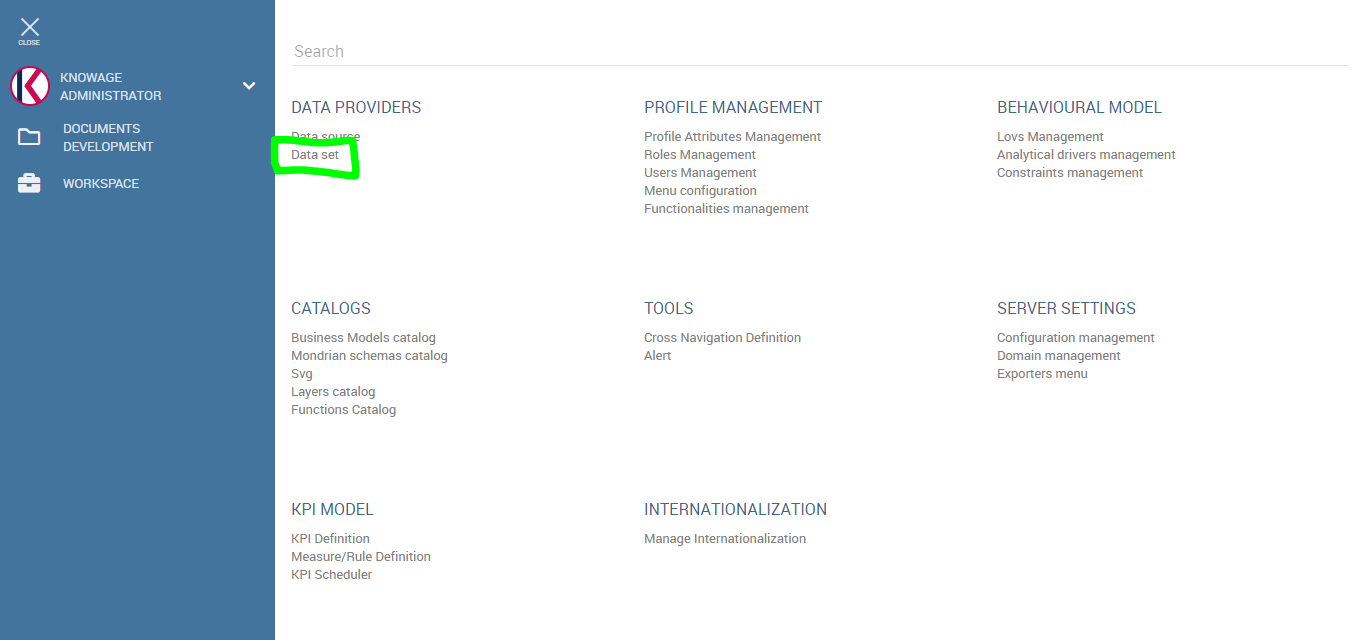
# Dataset configuration

Note: The dataset configuration is optional and can be skipped if your component is registered within the BSE.

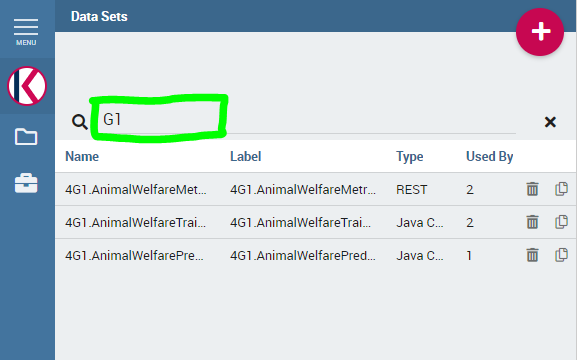
1. After login into Knowage click on **Menu** at top-left.



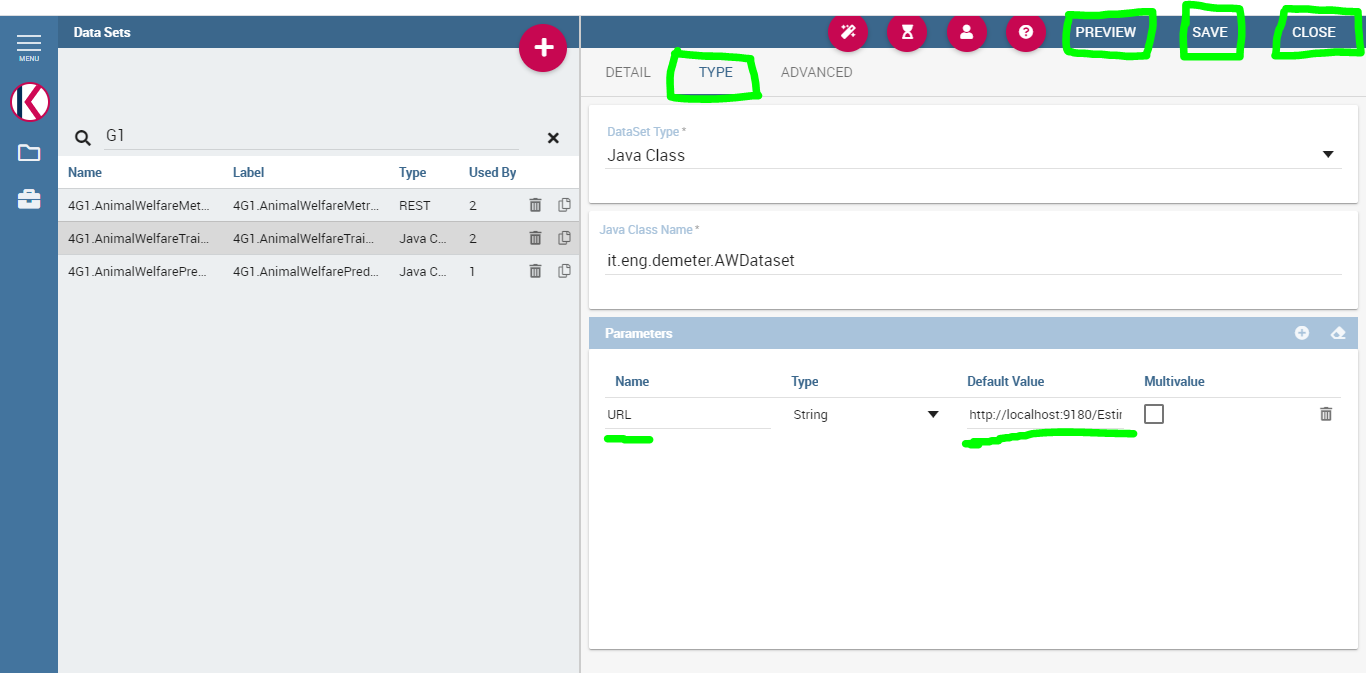
1. Then click on **Data set**.



1. A list of datasets will be loaded, input your component code into the **search** field to filter the list (i.e. "G1"). This funtionality will be very useful during the endpoint customization process.



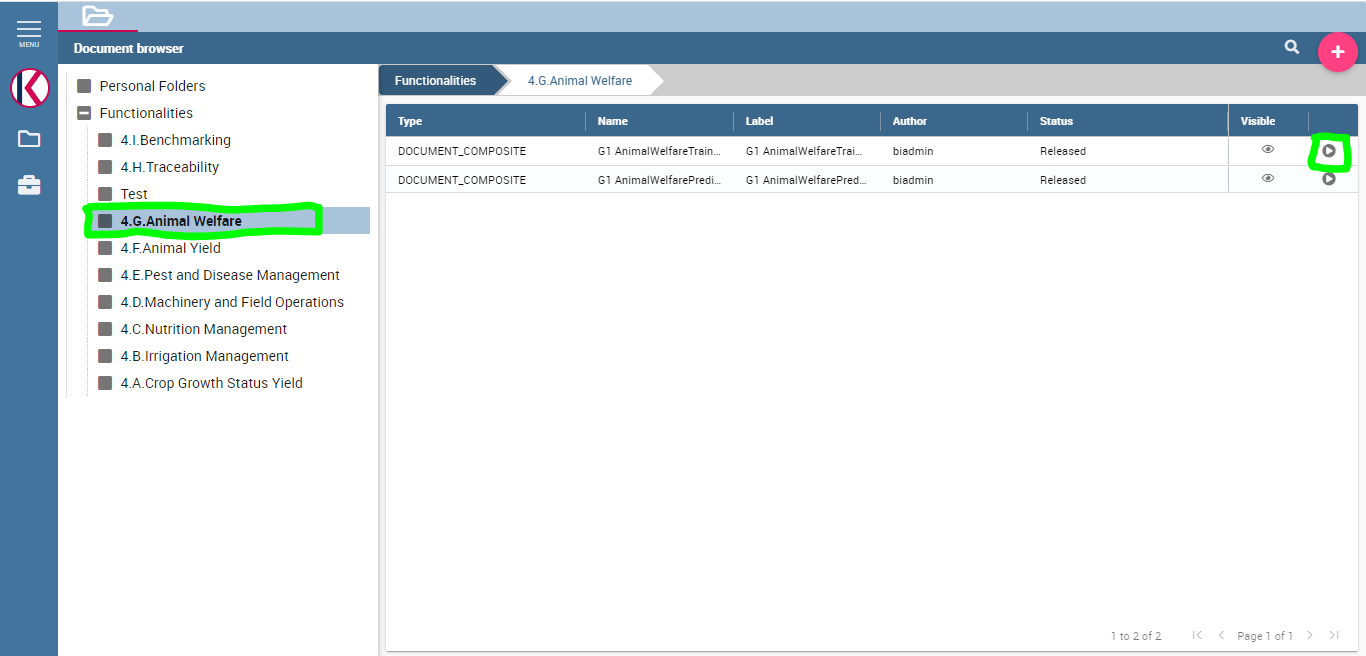
1. Click on the dataset you wish to configure and click on **Type** , here you can change the default value for the endpoint **URL parameter**
2. Click on **Preview** to check the data, then **Save**.



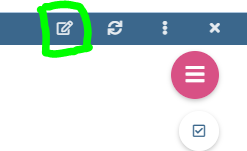
1. Now click the **folder icon** on the top-left menu to open the dashboards explorer.



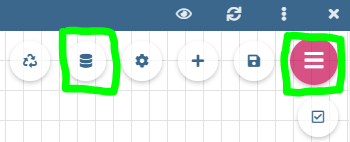
1. Select your area and open the dashboard by clicking on the **Play** button at right side.



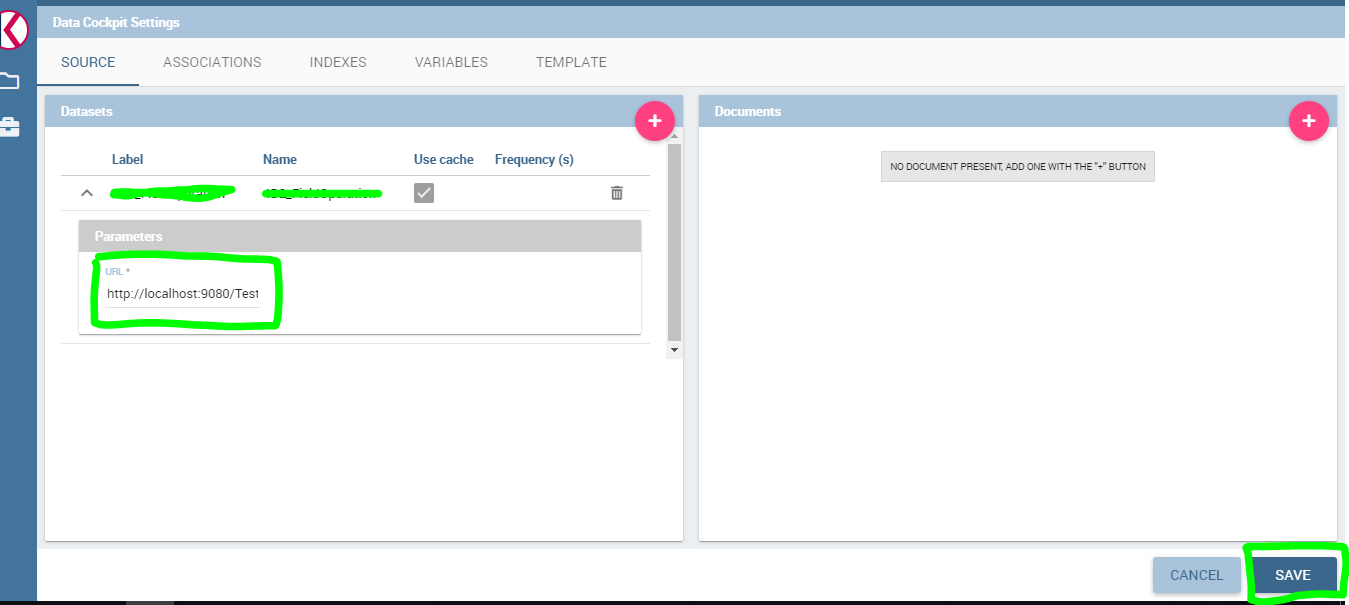
1. Now click the **Edit** button on top-right side menu.



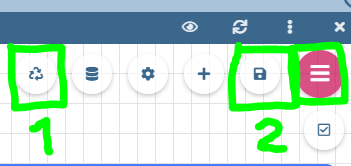
1. Click the red circle to open the **Cockpit menu** and click on **Data Configuration**



1. Here you will notice the same **endpoint URL parameter** that need to be changed with the new one, then **Save**.



1. Click again the **Cockpit menu** and click on **Clear Cache** first, then **Save** the cockpit.



1. Click on the eye to go back in **View mode** or **close** and **re-open** the dashboard.

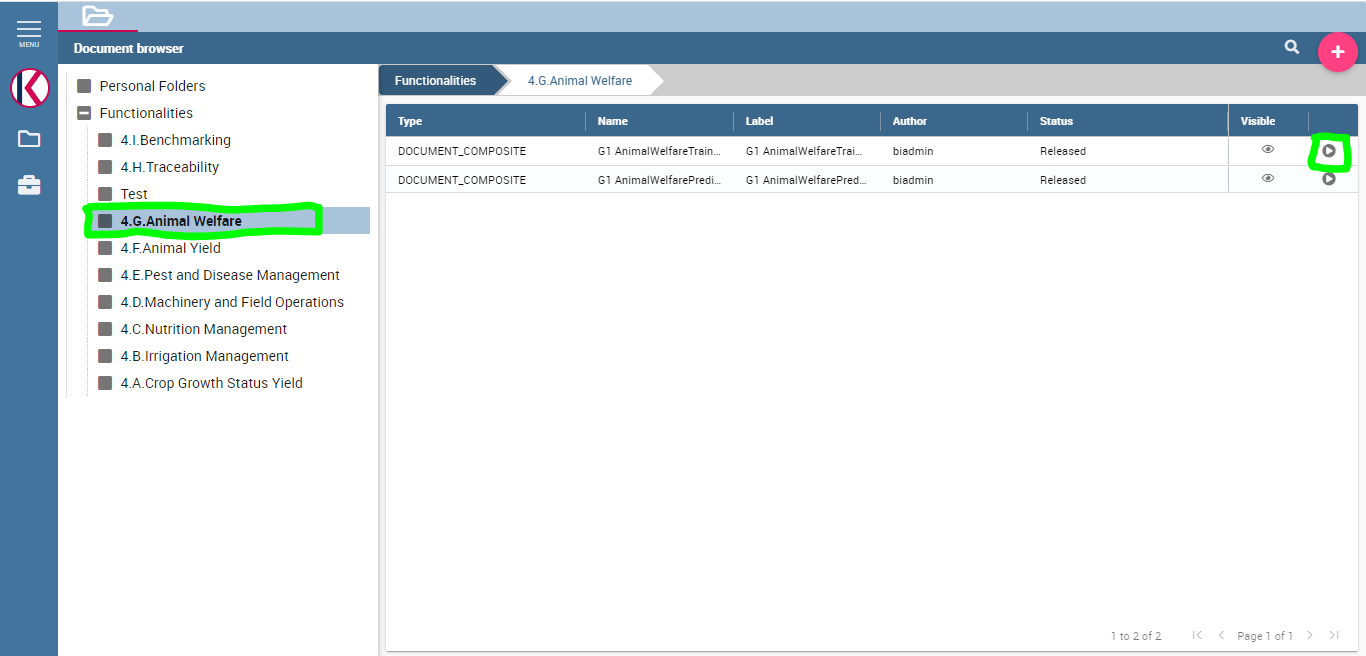


# Dashboard execution

1. Once logged in, click the **folder icon** on the left side menu.

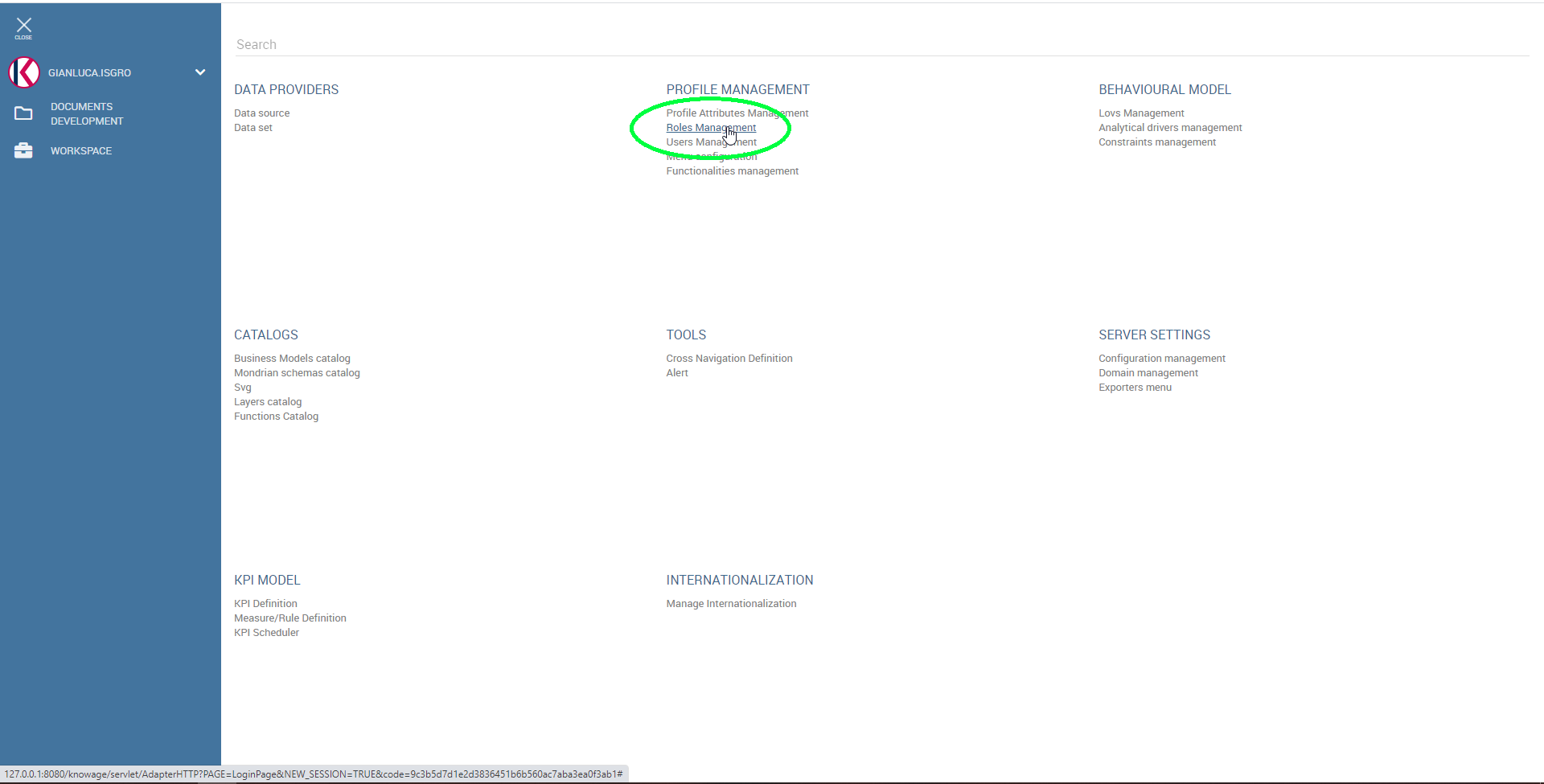


1. This will open the cockpit window, with a list (**Functionalities**) containing each **area component**.
2. Click the desired area to show up the relative list of **dashboards** developed.
3. To open a dashboard simply click on the "**play**" button at the right side of the dashboard description.

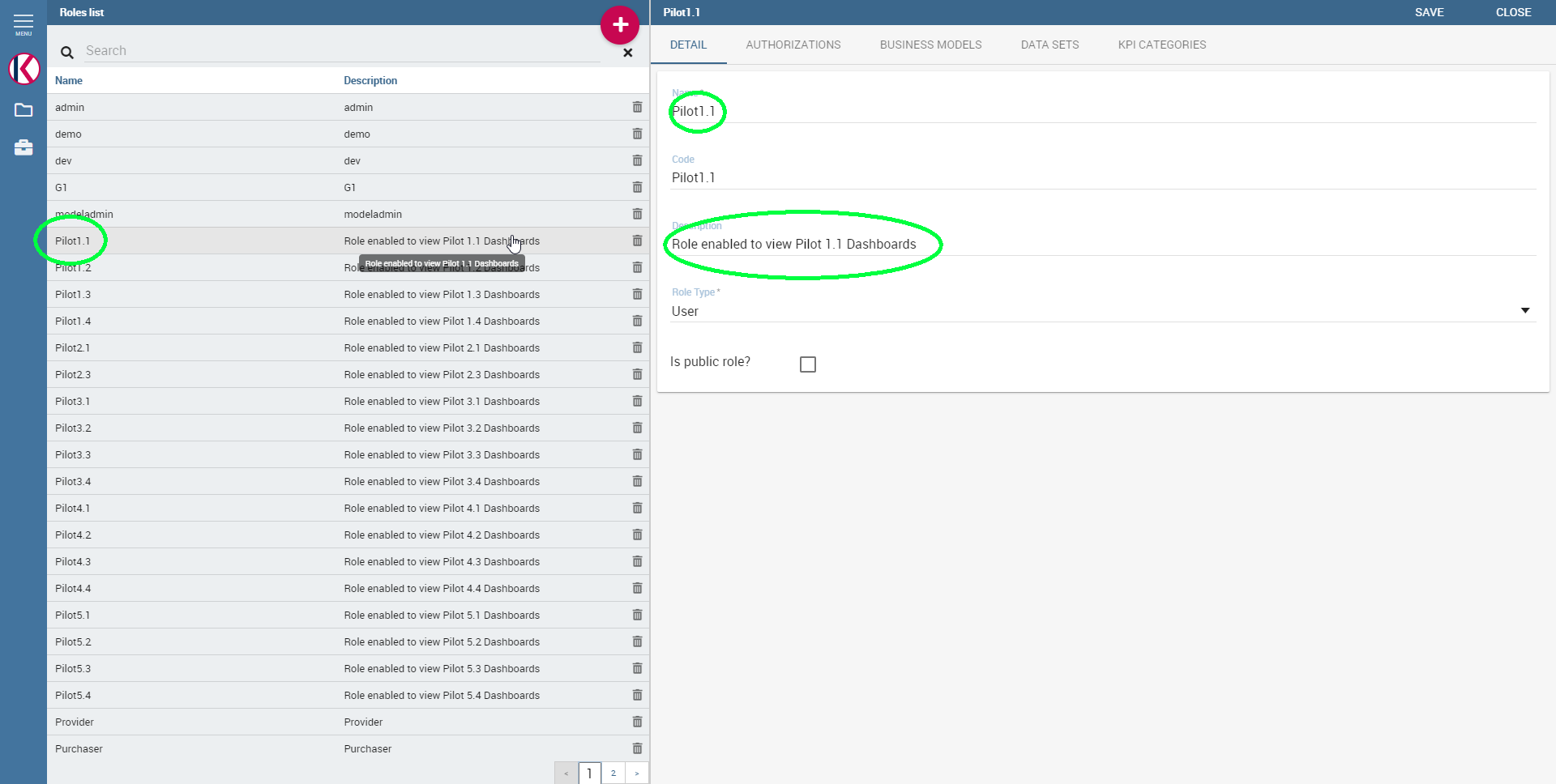


# Dashboards view by role

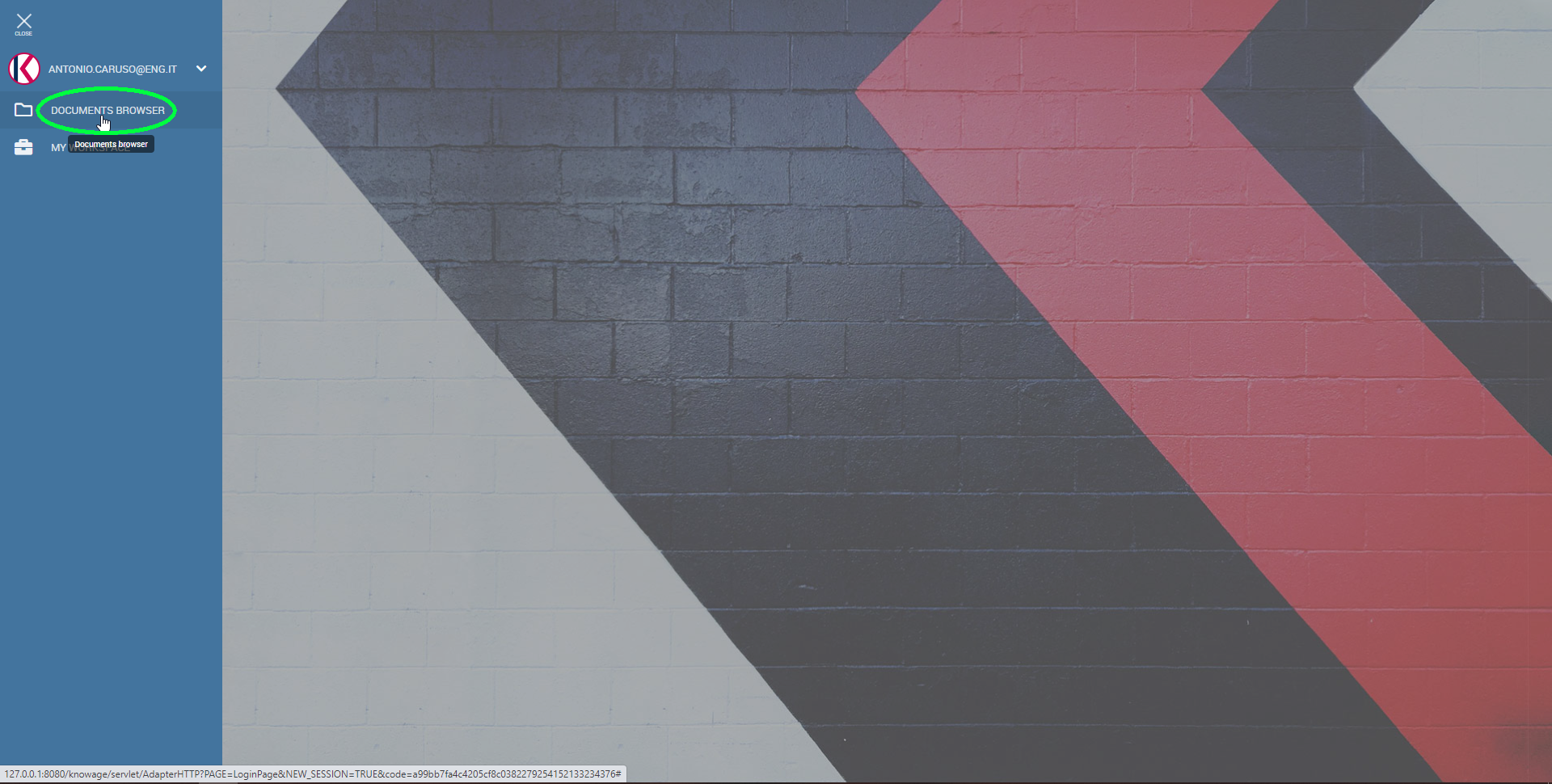
1. Management of **roles** by the **administrator**.



1. List of **user roles** by **pilot**, visible and manageable by the **administrator**.



1. **Pilot x.y** user login.



1. View dashboards list for **Pilot x.y user**.

