Contents

[1. Tech Stack 1](#_Toc46505171)

[2. MongoDB Setup 1](#_Toc46505172)

[2.1 PRD\_LDS 1](#_Toc46505173)

[2.2 Data-Catalog 1](#_Toc46505174)

[3 Dream Factory Setup 2](#_Toc46505175)

[3.1 Create API Service 2](#_Toc46505176)

[3.2 Create Roles 3](#_Toc46505177)

[3.3 Add Scripts 3](#_Toc46505178)

[3.4 Add users and set roles 4](#_Toc46505179)

[3.5 Create App 5](#_Toc46505180)

[4 Data-Catalog Portal 6](#_Toc46505181)

[4.1 Pull Code from Github 6](#_Toc46505182)

[4.2 Add Email service code 6](#_Toc46505183)

[4.3 Add Environment Variables 6](#_Toc46505184)

[4.5 Deploy 7](#_Toc46505185)

[**4.5.1 Build File** 7](#_Toc46505186)

# Tech Stack

PRD\_LDS

Data-catalog

ReactJS

Dream Factory

Front End

Back End

Data Base - MongoDB

# MongoDB Setup

## PRD\_LDS

* Create PRD\_LDS if it does not exist
* This database contains all the datasets that is available to users
* Import/create datasets that are to be displayed on data-catalog portal.

Sample data can be imported from <https://github.com/JTC-DataEngineering/Data-Catalog/tree/master/database>

## Data-Catalog

* Create data-catalog database if doesn’t exist in mongodb
* This database contains the metadata of datasets available and collections that is required for various data-catalog portal features.
* Import/create the following collections
  + collection\_metadata
  + attribute\_metadata
  + collection\_rating
  + collection\_comment

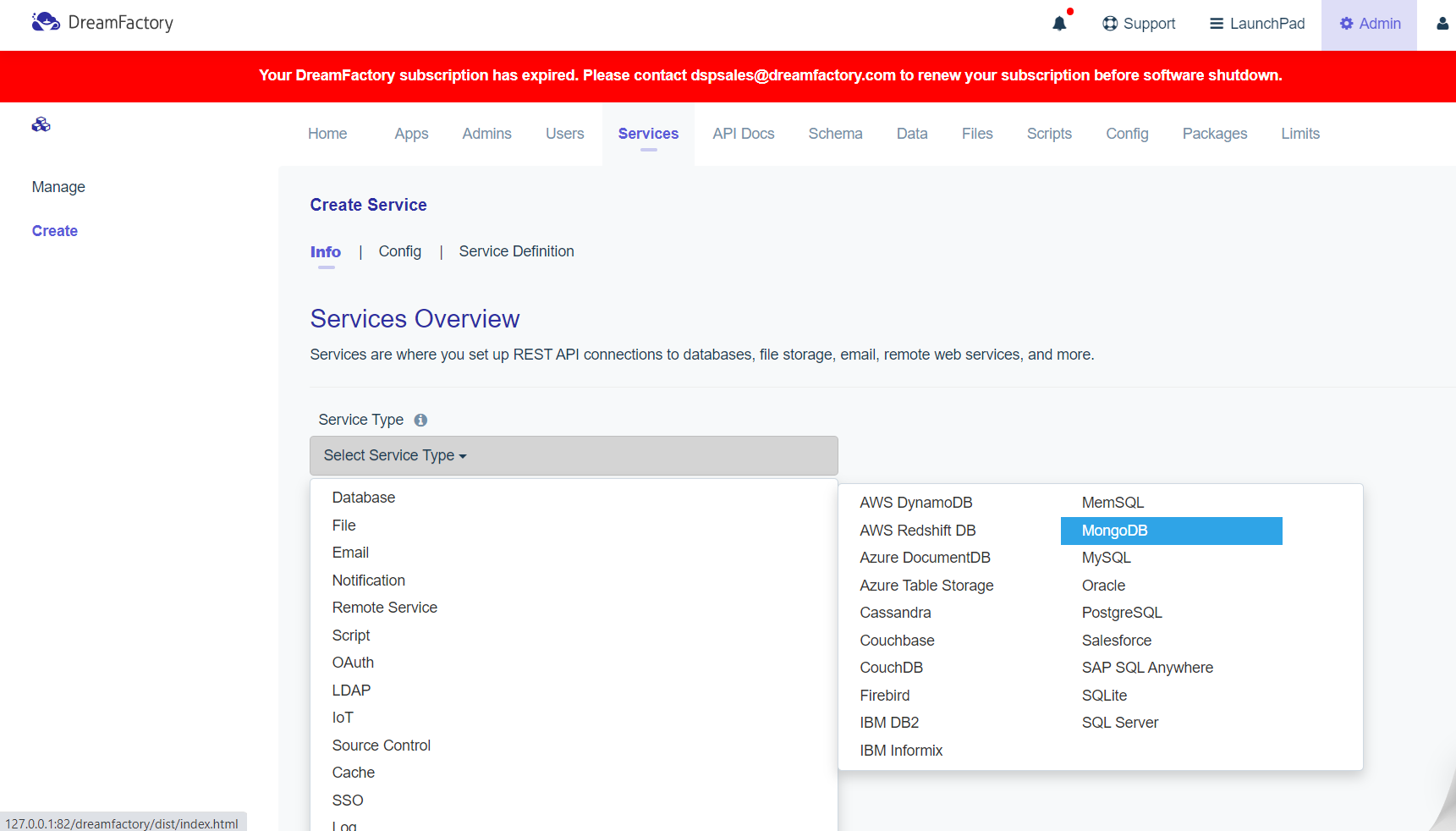
Sample data can be imported from <https://github.com/JTC-DataEngineering/Data-Catalog/tree/master/database>

# Dream Factory Setup

## 3.1 Create API Service

Navigate to Home>Services>Create.

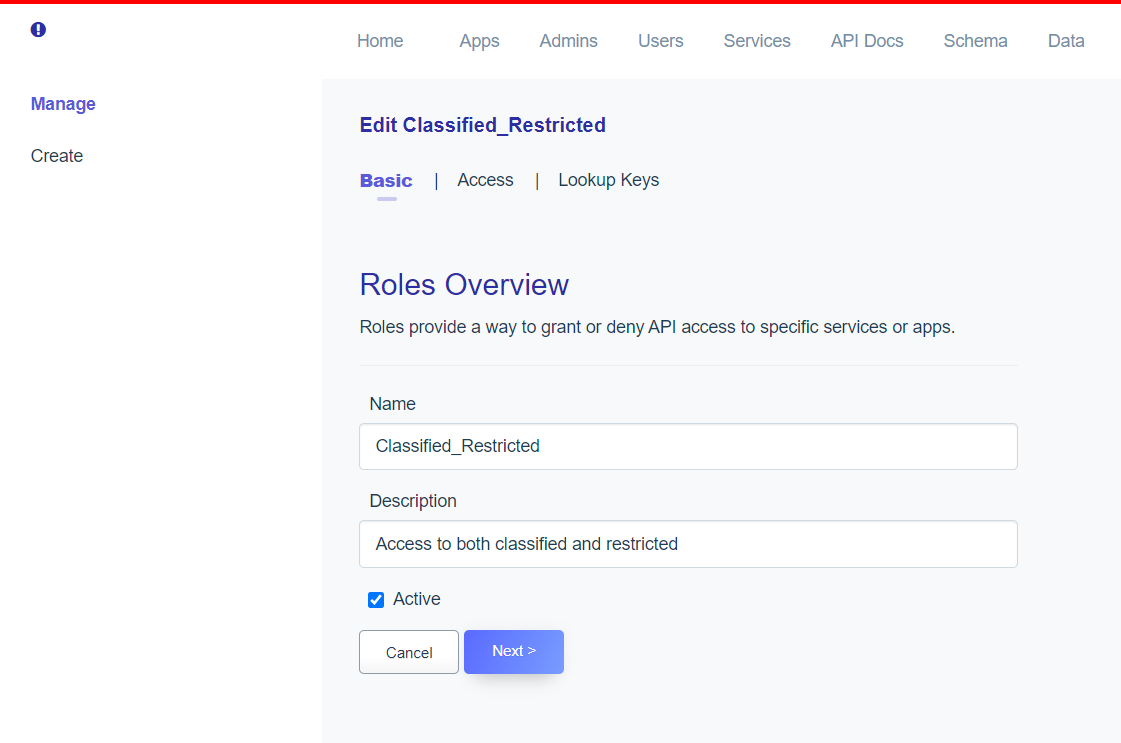
* Select service type > Database> MongoDB > Enter service Details > Config > Enter MongoDB Config
* Create service for both data-catalog and PRD\_LDS (if doesn’t exist)



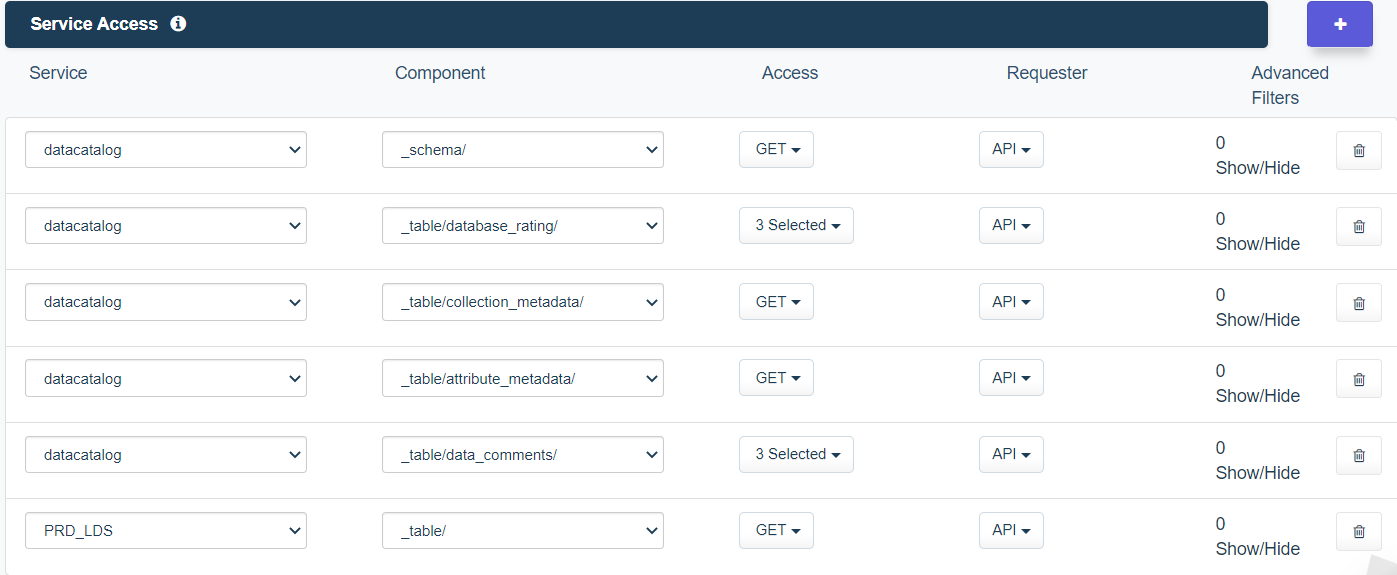
## 3.2 Create Roles

Navigate to Home>Roles>Create

* + 1. **Enter Role details**
* Enter desired role name and description for the role. Make sure active checkbox is checked.



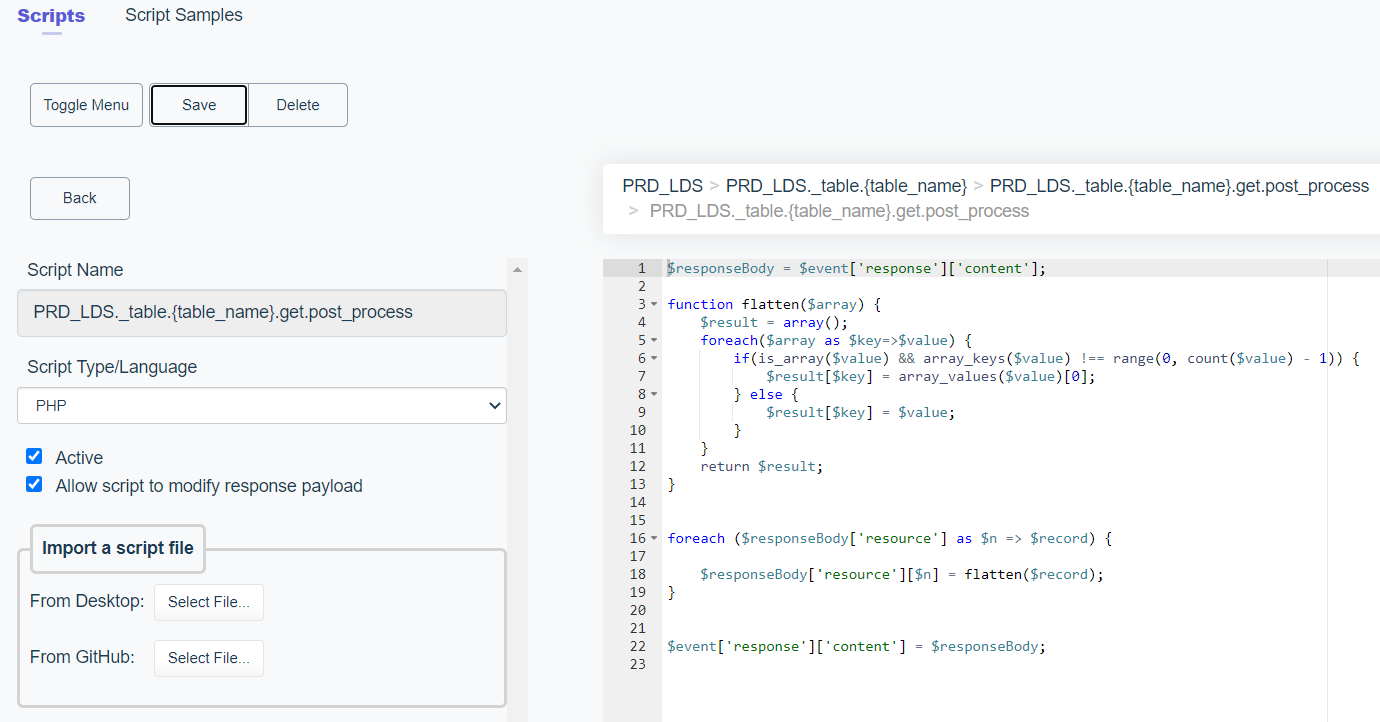
* + 1. **Add all the access rights for this role.**
* Click Access tab and click the + sign to add the access rights for this role, the GET/POST/PATCH access rights has also to be specified.



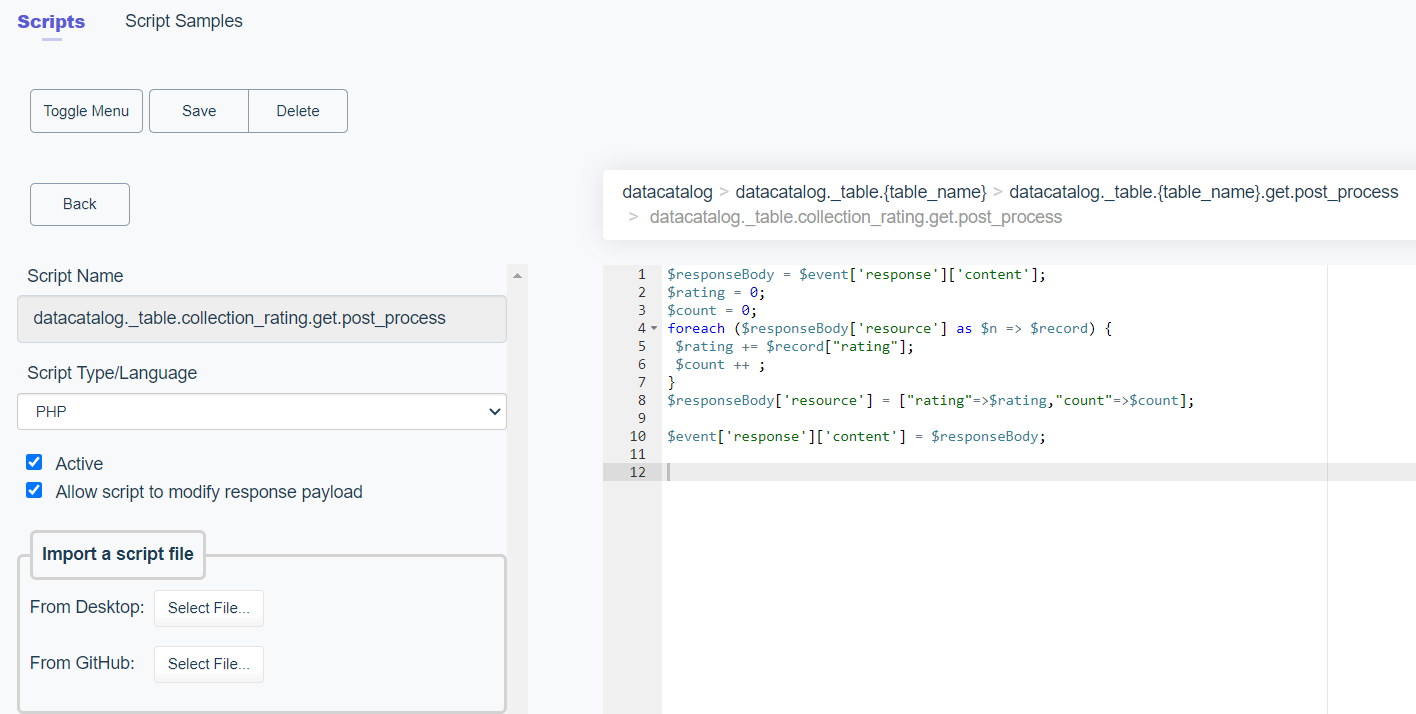
* + 1. **Repeat for all the roles required**

## Add Scripts

* + 1. **Flattening Script**
* Navigate to Home>Scripts>PRD\_LDS>datacatalog.{table\_name}.get.post\_process
* Add the following script to PRD\_LDS.\_table.{table\_name}.get.post\_proces



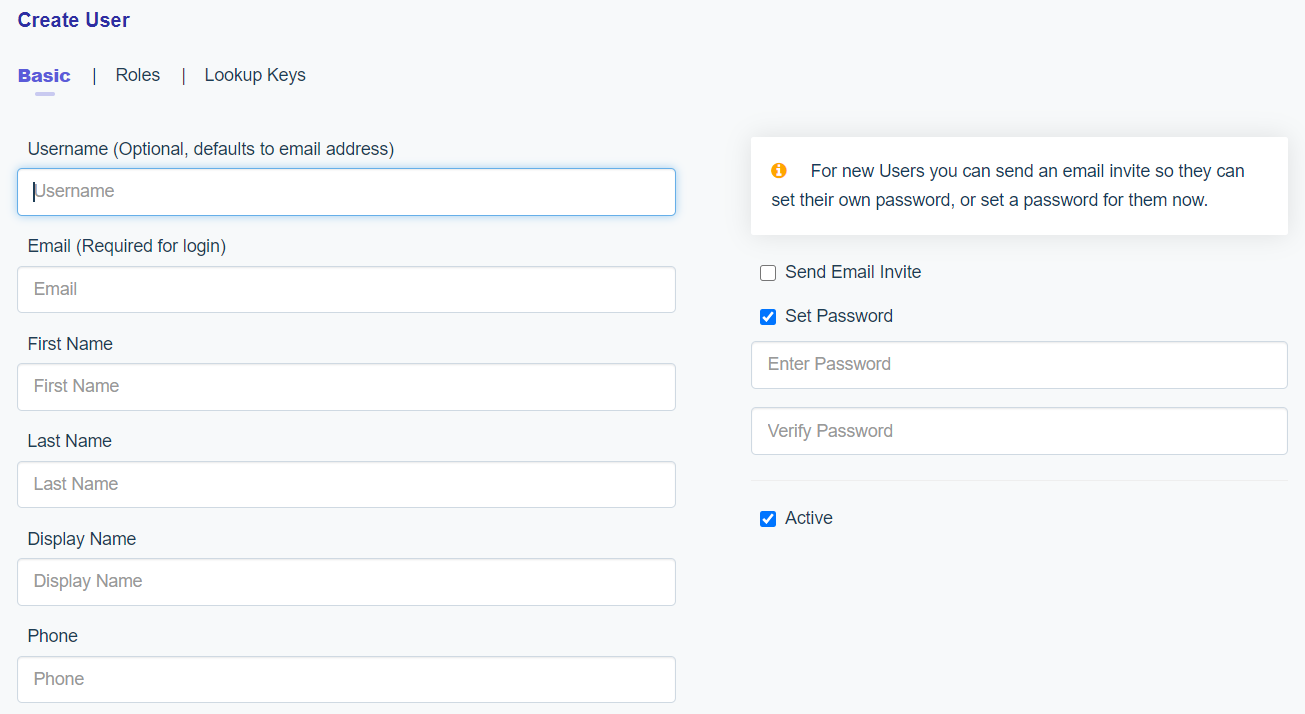
* + 1. **Rating Script**
* Navigate to Home>Scripts>datacatalog>datacatalog.{table\_name}.get.post\_process
* Add the following script to datacatalog.\_table.collection\_rating>get>post\_process



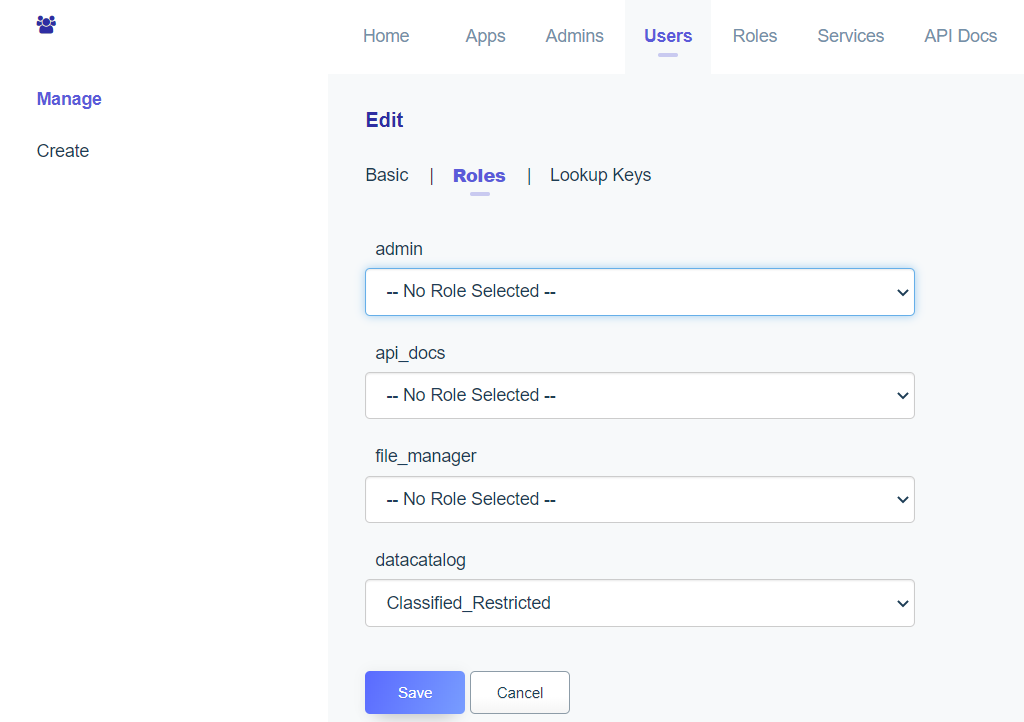
## 3.4 Add users and set roles

Navigate to Users>Create

* + 1. **User Details**
* Enter user details and set password to the default password.
* Active has to be checked



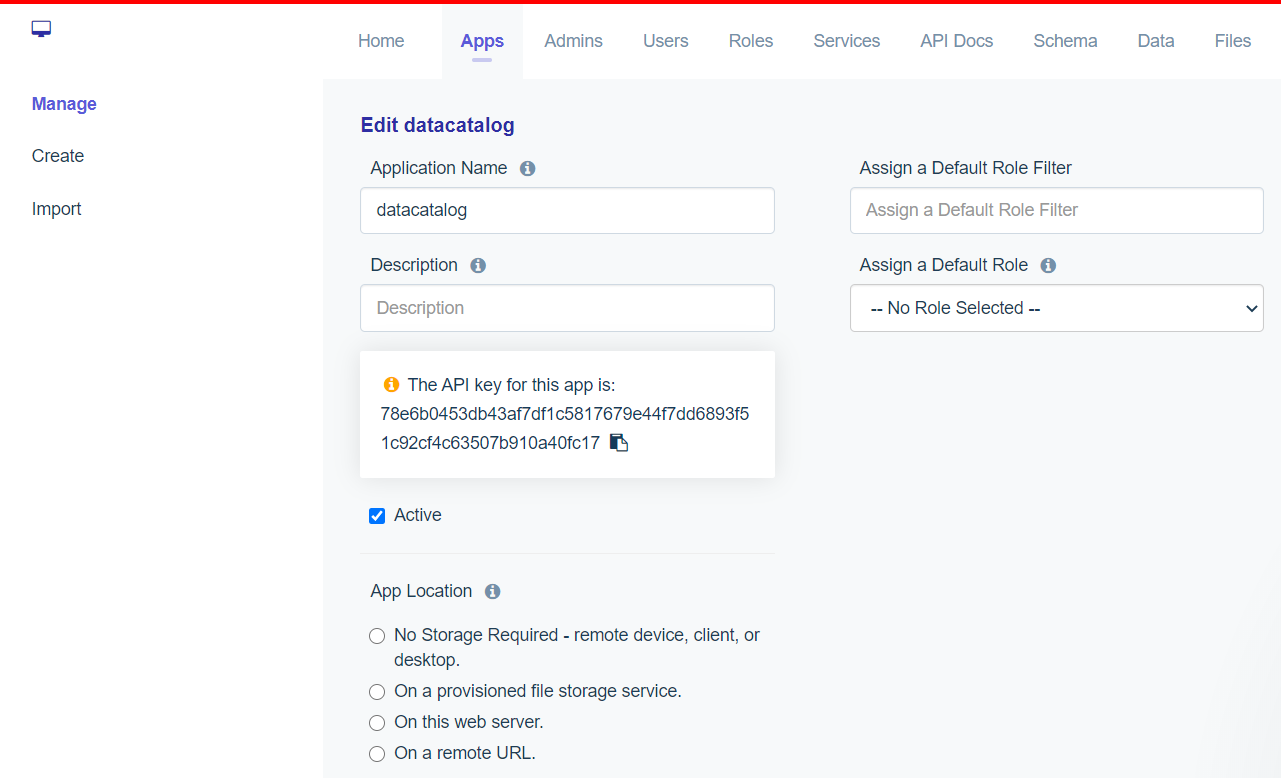
* + 1. **Roles**
* Click Roles and assign roles for the user according to access rights



## Create App

Navigate to Apps>Create

* Enter App details and check the active check box.
* An API key will be generated after app is created. This key will be used by users to access the data-catalog app.



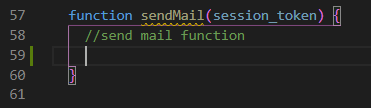
# Data-Catalog Portal

## 4.1 Pull Code from Github

* Code can be pulled from <https://github.com/JTC-DataEngineering/Data-Catalog>

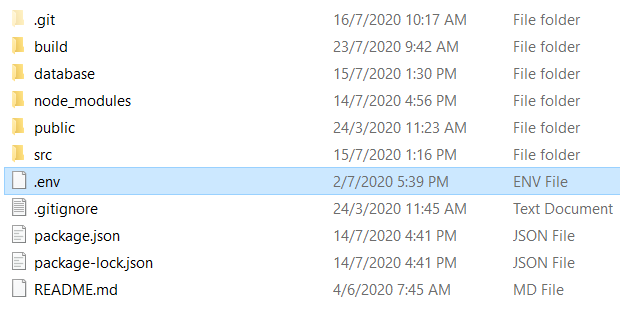
## 4.2 Add Email service code

Add Email service code in Login.js

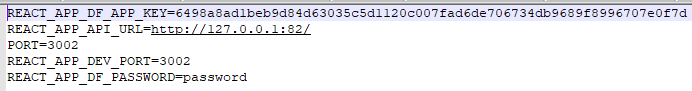


## Add Environment Variables

* + 1. **Add a .env file in the root folder**



* + 1. **Add the following variables**
* REACT\_APP\_DF\_APP\_KEY , app key generated at step 3.5
* REACT\_APP\_API\_URL, URL where the data-catalog is going to be hosted
* PORT, port where the data-catalog is going to be hosted
* REACT\_APP\_DEV\_PORT, same as above, but will be used in the app
* REACT\_APP\_DF\_PASSWORD, default password set to every user



## 4.4 Deploy

### **4.5.1 Build File**

Enter npm run build to build the app for production to the build folder.

**4.5.2 Deploy the build**

Enter Serve -s build. Web app will be hosted at the port number specified in the .env file