The Application is built based on 3 layers (API, Service, Data Access) and enables a user to send (GET, POST) HTTP requests to PostgreSQL Database.

1.

Expected input:

Request URL: localhost:8080/api/person

Method: GET

Expected output: JSON object, which contains the data currently stored in the DB

2. Expected input:

Request URL: localhost:8080/api/person

Method: POST

Expected output: if a person by the given name exists in the DB, throw an Exception with the message: “Already exists”, otherwise save JSON object, which contains the data to the DB

3. Expected input:

Request URL: localhost:8080/api/person/:id

Method: GET

Expected output: return a JSON object based on ID, otherwise throw an Exception with the message:"Student with id " + personId + " has not been found".

Classes:

@DemoApplication

@Person

@PersonConfig

@PersonController

@PersonService

@PersonRepository

Functionality:

@DemoApplication

Initializes the application

@Person

Defines the object that is to be stored in the database.

Class Body:

Parameters: id, firstName, lastName

Constructor, getters, setters

@PersonConfig

PersonConfig Adds two instances of the Person class to the DB

Class Body:

Method @commandLineRunner saves two variables to the DB

@PersonController

PersonController represents the API layer of the application, which sends CRUD requests to the DB

Class Body:

Specifies RESTful requests a user can make to the DB. (CRUD operations)

@PersonService

PersonService Class represents the interactions of the service layer of the application with the Data Access Layer

Class Body:

Defines CRUD operations on a service layer through methods @getALL, @addPerson, @getPerson

@PersonRepository

PersonRepository Describes the functionality of the Data Access layer, or how the Application interacts with the Postgres Database

Class Body:

Handles requests from the user on the Data Access layer (direct communication with the DB).