# Superloop Todo Task project Summary

TodoTaskAPI application is a spring boot application which exposes certain REST APIs. This application provides services like

1. Creating a new TODOItem
2. Editing a TODOItem
3. Viewing the TODOItem
4. IntegrationTesting the TODOItem endpoints
5. Validating a TODOitem
6. Deleting a TODOitem
7. Editing a TODOitem status

* Since spring boot provides Rapid Application development, it’s easily deployable without any requirement of web application servlet containers, being a standalone java application.
* Spring Data JPA is used to write the Data access layer.
* This project utilizes an embedded database (HSQL) to connect and query data. There is a customized create-table.sql written which when executed will create the table structure in the embedded database.

**Endpoints**

Below are the endpoints defined in the project.

/api/v1

/todo

/todos/{status}

/todo/{id}

/todos

/todoStatus

Response Codes: Success (200 OK), Bad Request (400), Unauthorized (401)

# Application Flow

TODO Task API utilizes the spring mvc as the framework which consists of the below 4 layers. However, there is no View part associated with this application since it concentrates only on the backend service endpoints.

* Controller
* Service/ServiceImpl
* Business Object/Business Object Impl
* Repository

The controller class is annotated with @RestController, which exposes the endpoints and acts as an entry point which receives requests from clients. Controller class defines the 3 endpoints namely

* A browser/client request gets mapped to the controller through respective endpoint. The controller identifies the request and forwards the same to the corresponding Service layer. The code now needs to execute some business logics for which the Business classes are written and at last to talk to the database it gets connected with the Repository layer. The model/VO stores the value objects which can then be used in various layers mentioned above

The JPATransactionManager have been used to support and manage our transactions. The required tables are created in the ‘In memory’ database (HSQL), We can see and execute the queries in tables.

The list of users for application authentication purpose has been maintained as in memory

**Testing the Endpoints**

**Note: While building the UI project, may need to install devkit**

**npm install --save-dev @angular-devkit/build-angular**

The APIs are tested using Angular Client.

**Swagger URL**

 