

## Task To Do Application

The web based online Task to Do application has been built using Spring Boot Framework for backend and Angular 10 for building frontend task to do client. Authentication is implemented using basic auth and spring security.

To Do application performs the following task,

1. User can login using user credentials.
2. User can view his/her task.
3. User can add/remove/update his/her task.
4. All changes can be persistent to allow view them in next sign in by the same user.
5. Each task should display the date of last updates and description.
6. User can check/uncheck any task on their list .

### **Back End Implementation :**

Back end application and rest api has been implemented using SpringBoot and Spring Security. Basic Authentication will be used to authenticate any REST calls made from Angular to Spring Boot. The HTTP protocol is used to create a simple authentication mechanism called basic authentication. HTTP requests utilizing this protocol have an Authorization header with the word Basic, a space, and the base 64 encoded string username:password.

The application is implemented using H2 database and spring data JPA and two hardcoded users which can be used for testing are,

- Username : **“test”** and password : **“pwd123”**.
- Username : **“testleaduser2”** with password : **“87654321”**

### **Front End Client:**

Front End Client has been implemented using Angular 10. The implementation has been done by breaking complex code into reusable parts i.e. Components. Components are basically classes that interact with the .html file of the component, which gets displayed on the browser. The client is built using breaking into several components i.e. login, tasktodo, updateTaskToDo, CreateTaskToDo, logout, header. For data connection to fetches data from database used in multiple components has been achieved by using services i.e. TaskToDoService, AuthService, BasicAuthInterceptorService.

.

### **Future things to Do :**

1. Use Spring Aspect Oriented Programming for logging.
2. Use JWT authentication instead basic -auth.
3. Remove uses of hard coded test users.
4. Include junit test cases for every api method invocation.

