Wired2Perform Coding Assignment

Kavindhya Kurupitage

ToDo Application

1. Used Tools and platforms

ECLIPSE IDE

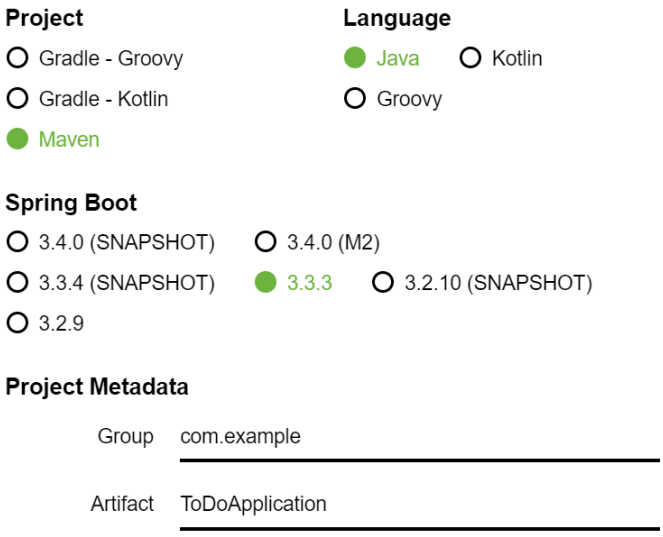
Spring Boot

JAVA Language

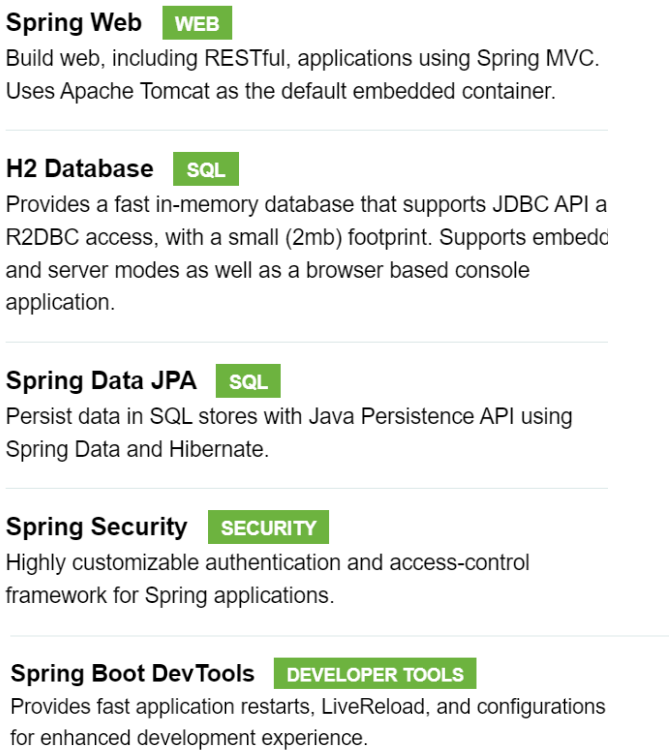
SQL

POSTMAN

2. Created a project using ***Spring Initializr.*** (selected Maven project, JAVA language and Spring Boot version 3.3.3)

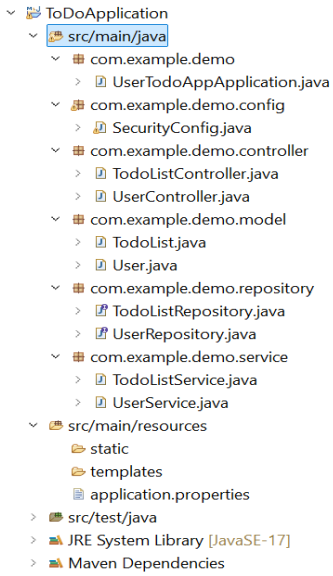


3. Used several dependencies

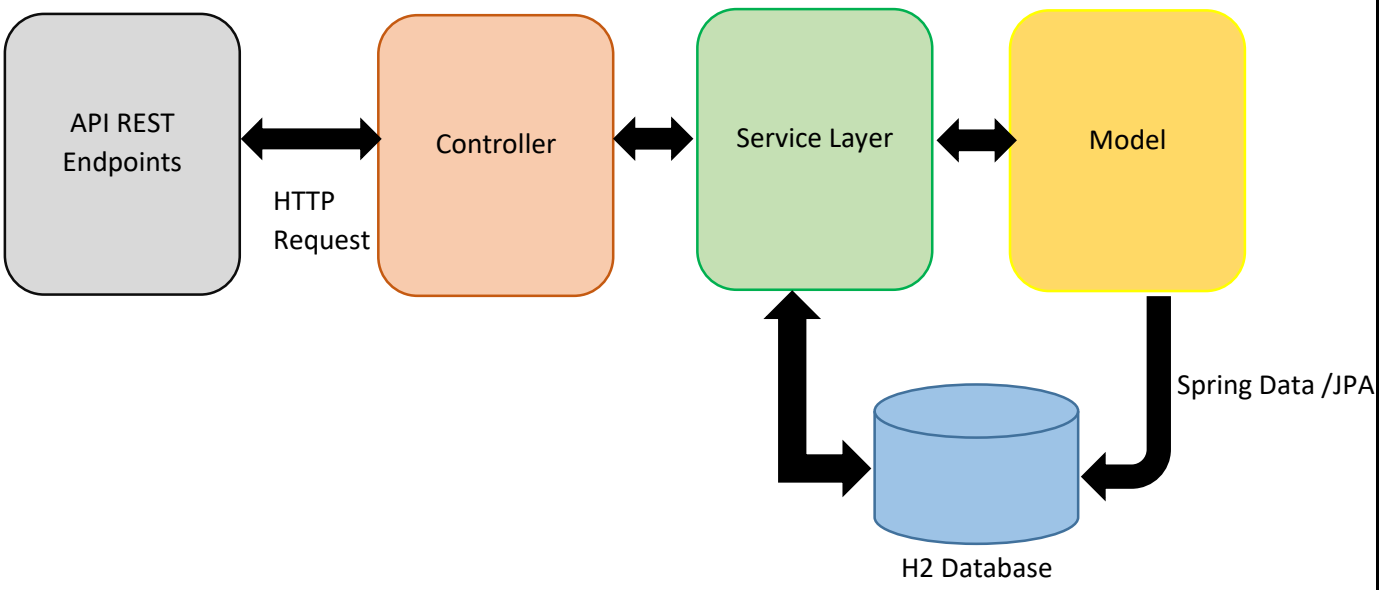


- Spring Web is used to build the REST API.
- H2 Database is used for the in-memory database.
- Spring data JPA is used to handle the database side.
- Spring Boot DevTools is used to easy development techniques.
- Used Spring Security for authentications.

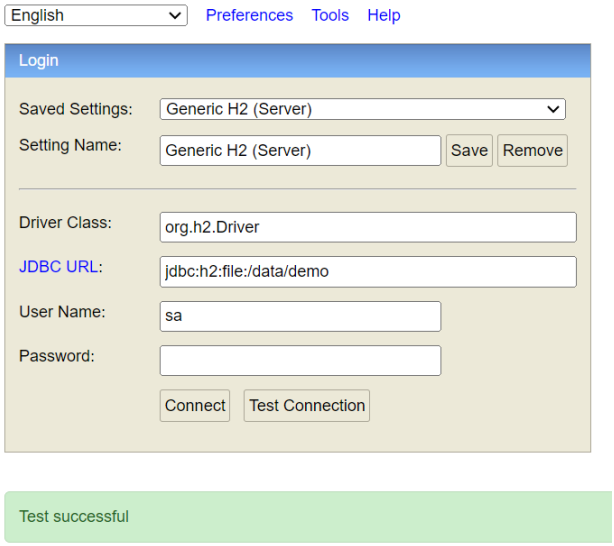
4. This is my ToDo Application folder structure



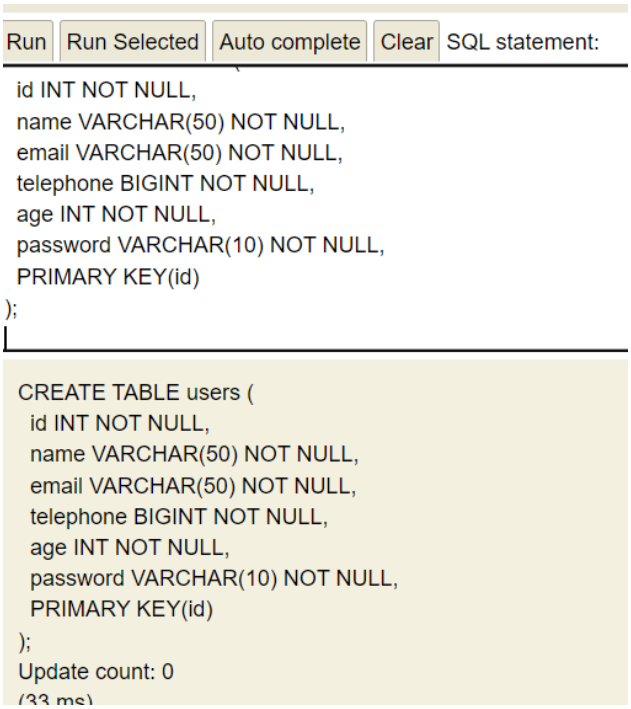
5. My ToDo Application architecture



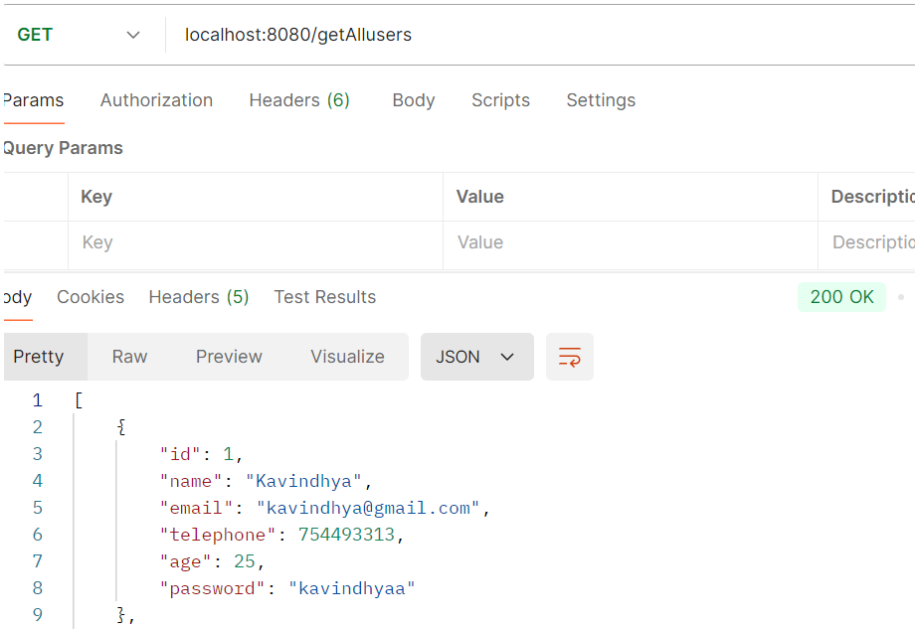
6. After running the UserTodoAppApplication.java file open the H2 database using local host ***http://localhost:8080/h2-console***.



7. Created a data table including users.Used SQL queries for it.



8. Tested using Postman



9. Tested other CRUD operation using Postman.

```java
@RestController
@RequestMapping("/api/todolists")
public class TodoListController {

    @Autowired
    private TodoListService todoListService;

    @PostMapping("/create")
    public ResponseEntity<TodoList> createTodoList(@RequestBody TodoList todoList) {
        TodoList newTodoList = todoListService.createTodoList(todoList);
        return ResponseEntity.ok(newTodoList);
    }

    @GetMapping("/user/{userId}")
    public ResponseEntity<List<TodoList>> getTodoLists(@PathVariable Long userId) {
        List<TodoList> todoLists = todoListService.getTodoListsByUser(userId);
        return ResponseEntity.ok(todoLists);
    }

    @PutMapping("/update")
    public ResponseEntity<TodoList> updateTodoList(@RequestBody TodoList todoList) {
        TodoList updatedTodoList = todoListService.updateTodoList(todoList);
        return ResponseEntity.ok(updatedTodoList);
    }

    @DeleteMapping("/delete/{id}")
    public ResponseEntity<Void> deleteTodoList(@PathVariable Long id) {
        todoListService.deleteTodoList(id);
        return ResponseEntity.ok().build();
    }
```