University Of Tunis
Tunis Business School
IT325 Web Services Final Project

Personnel Deployment Tracker

Defense Industry

- Rest Api -

Student:

Balcem Hassine

Supervisor:
Dr. Montassar Ben Messaoud

Tunis,
January 2024.

Table of Contents

Abstract

In navigating the multifaceted terrain of Tunisia’s defense industry, the intricacies of personnel management emerge as a linchpin for the overarching goals of operational efficiency and bolstered national security. This in-depth exploration delves into the paradigm-shifting realm of a transformative personnel tracking solution meticulously designed to transcend conventional approaches entrenched in the Tunisian defense framework.

In response to the ever-evolving dynamics of the industry, this innovative system stands poised to reshape the very fabric of how personnel are tracked and managed. Beyond the conventional bounds, it becomes a strategic ally in the optimization of resource allocation, a veritable compass in the labyrinth of workforce management intricacies. Its forward-looking design integrates cutting-edge technologies, effectively positioning it as a vanguard in the digital transformation journey undertaken by Tunisia’s defense sector.

Within the intricate architecture of this groundbreaking solution lies not just an instrument of efficiency but a cornerstone aligning with Tunisia’s strategic imperatives. It represents a pivotal step towards a more agile, responsive, and operationally ready defense infrastructure. By harnessing the power of technology, this innovative approach seamlessly aligns with and propels Tunisia toward the forefront of modern defense practices, promising a future marked by heightened transparency, strategic agility, and a robust defense apparatus ready to face the challenges of the contemporary geopolitical landscape.[1]

Chapter 1

Introduction

The Personnel Deployment Tracker API represents a robust solution developed using Java Spring Boot, designed to streamline and enhance the tracking of military personnel deployment. Leveraging the power of Spring Boot, this API ensures efficient and scalable performance, making it well-suited for the dynamic demands of military operations. Allowing for seamless storage and retrieval of critical data associated with deployed personnel.

Documentation is a key aspect of any successful project, and this API is no exception. The comprehensive documentation is created using Swagger tool, providing clear insights into the available endpoints, request parameters, and expected responses. This user-friendly documentation not only aids in project development but also serves as an invaluable resource for developers and stakeholders alike.

Git has been employed as version control. This allows for effective collaboration among team members and the ability to track changes throughout the development life-cycle.

Security is of paramount importance, and the Personnel Deployment Tracker API incorporates a small authentication inspired by the JWT authentication as a robust security measure. This implementation ensures that only authorized personnel have access to sensitive deployment information, providing a secure environment for military units to manage and monitor their personnel effectively.

The project encompasses all CRUD (Create, Read, Update, Delete) operations, offering a comprehensive suite of functionalities. From adding and updating personnel deployment information to retrieving critical details and deleting obsolete records, the API provides a versatile toolkit for military commanders to manage their personnel efficiently.

Chapter 2

Problem formulation and solving

2.1 Problem Statement:

Military units face the complex challenge of efficiently managing and monitoring the deployment status of personnel across diverse operational scenarios. The absence of a centralized system often results in information silos, making it difficult for commanders to obtain real-time updates on the location, status, and activities of deployed military personnel. This lack of visibility hampers effective coordination and communication, hindering commanders from making timely and informed decisions. Additionally, without a streamlined solution, there is a risk of reduced personnel accountability, inefficient resource allocation, and challenges in responding promptly to emergencies. The need for a comprehensive tool to address these issues is evident, calling for a solution that simplifies the tracking and management of deployed personnel in a centralized and accessible manner.

2.2 Solution:

The PersonalDeploymentTracker API addresses the multifaceted challenge of efficiently managing and monitoring personnel deployment status in diverse operational scenarios. It provides a solution by offering tracking capabilities for enrolled personnel, including real-time location, key dates such as start and end dates, and a convenient listing of enlisted personnel with their respective skills and health status. The API enables efficient scheduling of future deployments by displaying available periods based on accepted leaves and other schedules, alongside health and deployment status. Additionally, it manages leave requests by allowing enlisted personnel to issue requests and commanders to accept or reject them. The API also handles the skill list of enlisted personnel, contributing to comprehensive deployment management. This solution is crucial for overcoming the existing challenges in military operations, ensuring centralized and accessible information to enhance coordination, communication, and decision-making among military units.

Chapter 3

Database Design

3.1 Entity Relationship Diagram

Below here is the ERD diagram for the elements in the database:

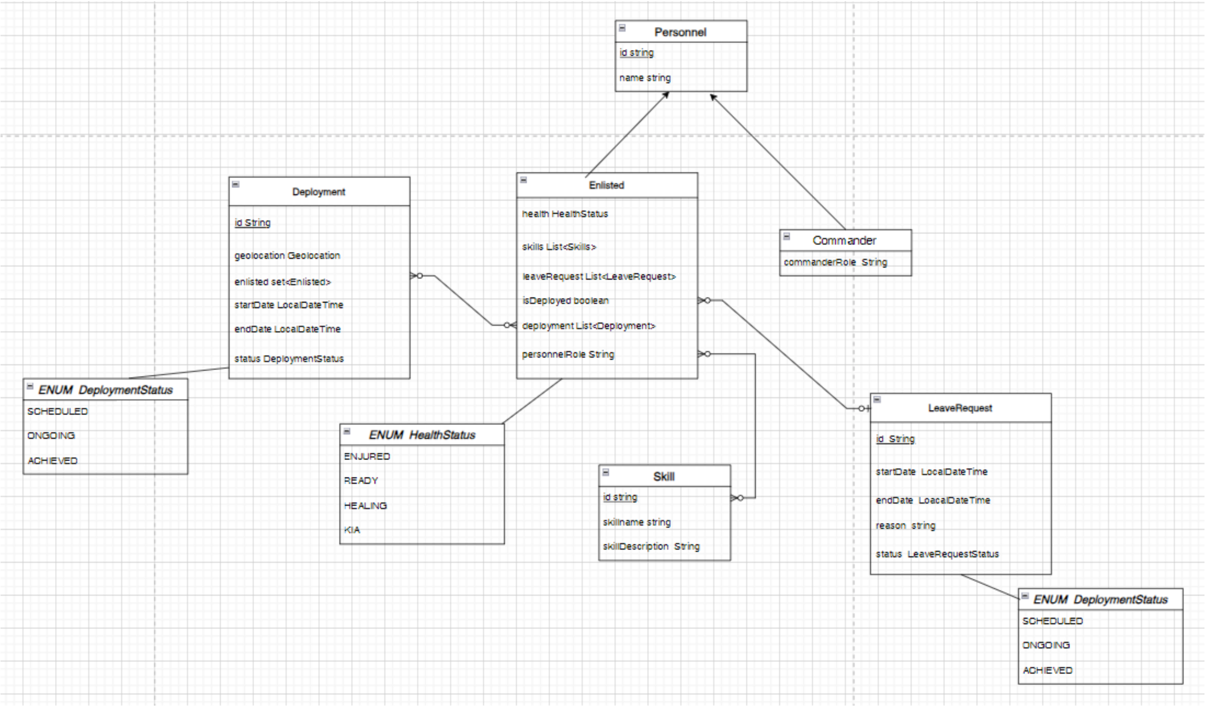


Figure 3.1: Entity Relationship Diagram

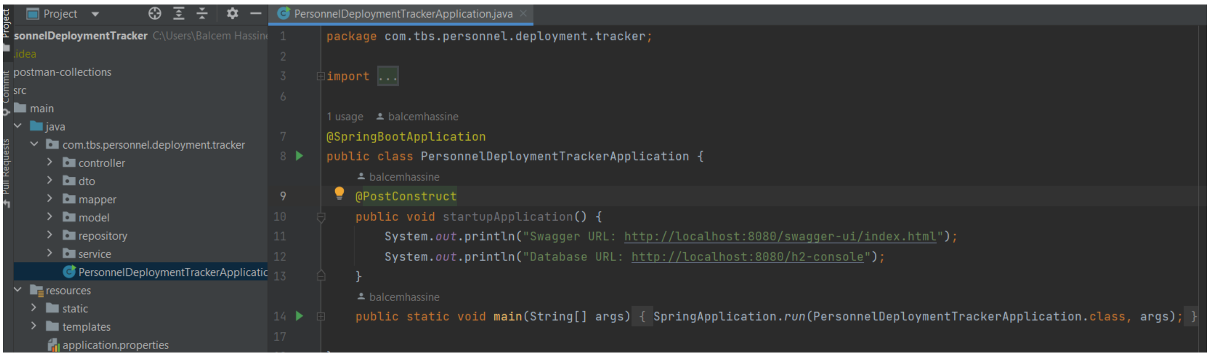
Chapter 4

Technologies used and the work carried out

4.1 Java Spring Boot:

The Java Spring Framework, commonly known as Spring, is a widely adopted open-source enterprise-level framework designed for developing robust, standalone applications on the Java Virtual Machine (JVM). Renowned for its versatility and scalability, Spring simplifies application development by providing a modular architecture and an extensive set of libraries. With principles like inversion of control and aspect-oriented programming, Spring empowers developers to create scalable and maintainable production-grade applications efficiently. Its widespread use underscores its pivotal role in Java application development. [2]

I have used the Java Spring Boot to develop my app.



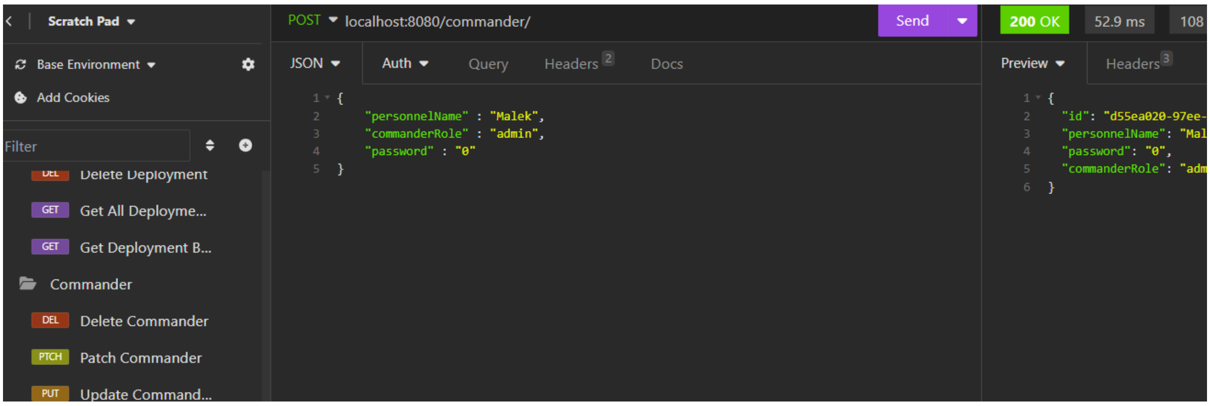
```java
package com.tbs.personnel.deployment.tracker;

import ...

@SpringBootApplication
public class PersonnelDeploymentTrackerApplication {

    @PostConstruct
    public void startupApplication() {
        System.out.println("Swagger URL: http://localhost:8080/swagger-ui/index.html");
        System.out.println("Database URL: http://localhost:8080/h2-console");
    }

    public static void main(String[] args) { SpringApplication.run(PersonnelDeploymentTrackerApplication.class, args); }
```

Figure 4.1: Java Spring Boot main class

4.2 Insomnia:

Insomnia is an open source desktop application that takes the pain out of interacting with and designing, debugging, and testing APIs. Insomnia combines an easy-to-use interface with advanced functionality like authentication helpers, code generation, and environment variables.[3]

I have used insomnia to make API calls.



POST localhost:8080/commander/

200 OK 52.9 ms 108

```
{
      "personnelName" : "Malek",
      "commanderRole" : "admin",
      "password" : "0"
}
```

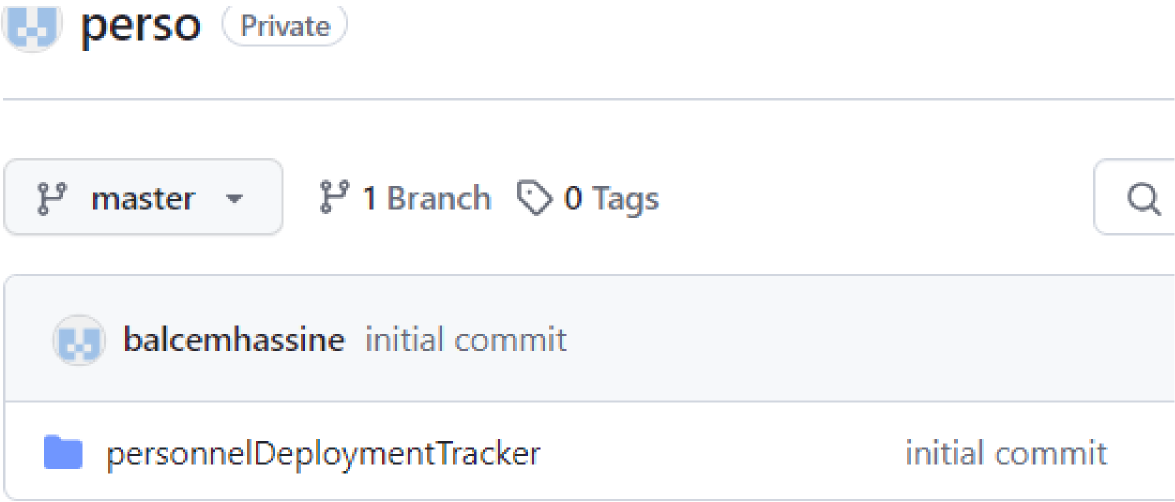
```
{
  "id": "d55ea020-97ee-
  "personnelName": "Mal
  "password": "0",
  "commanderRole": "adm
}
```

Figure 4.2: insomnia

4.3 GitHub:

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. [4]

I have used Git-Version Control to save changes and coordinate between the different repositories that I had.



balcemhassine initial commit

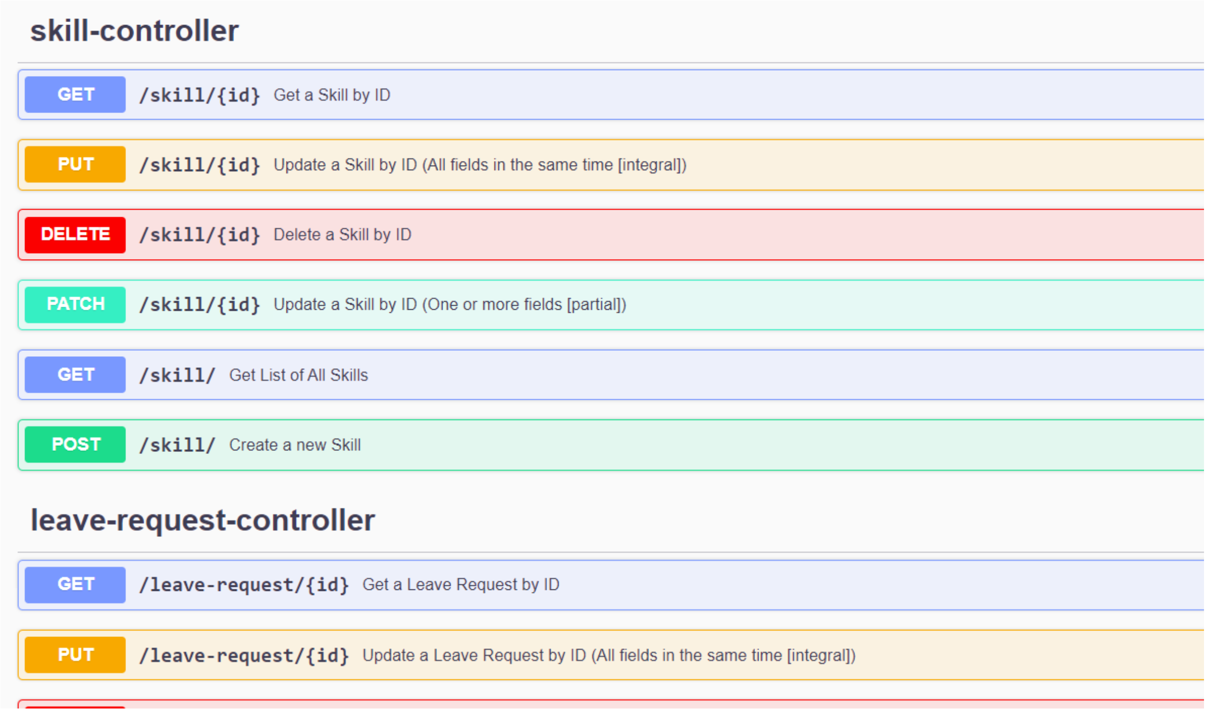
personnelDeploymentTracker initial commit

Figure 4.3: git

4.4 Swagger:

Swagger is an open-source framework for designing, building, and documenting RESTful APIs. It provides a set of tools for creating and managing API documentation, testing, and client code generation. Swagger was created by Tony Tam and his team at Reverb Technologies in 2011. The framework was later donated to the OpenAPI Initiative, which is now maintained by the Linux Foundation. Swagger is widely used to design and document APIs in various programming languages, and allows for a more efficient and streamlined development process by providing an easy-to-use interface for creating and managing API documentation.[5]

all the HTTP methods are documented using swagger.



skill-controller

GET /skill/{id} Get a Skill by ID

PUT /skill/{id} Update a Skill by ID (All fields in the same time [integral])

DELETE /skill/{id} Delete a Skill by ID

PATCH /skill/{id} Update a Skill by ID (One or more fields [partial])

GET /skill/ Get List of All Skills

POST /skill/ Create a new Skill

leave-request-controller

GET /leave-request/{id} Get a Leave Request by ID

PUT /leave-request/{id} Update a Leave Request by ID (All fields in the same time [integral])

Figure 4.4: Swagger

Chapter 5

Security and User Authentication

5.1 Log in

A user when registering should input the necessary information (ID, and password) to the request's body. The user will get a response containing a token that expires after a particular amount of time and should be used to authenticate the user.

5.2 User Authentication

5.2.1 Access Token:

The purpose of this token is to let a user get into the platform and do their transaction with their account from the resource server. This token has a short lifespan of 2 minutes for security purposes and for giving purpose to the next token.

5.2.2 Refresh Token:

The purpose of this token is to refresh the user's session. That way any malicious actor that might get the access token cannot use it whenever the user refreshes the session with the refresh token. when the user does that, they will get a new access and refresh token.

5.3 Authentication flow:

The authentication workflow requires a token and a specific workflow to go with it.
Below here is a figure and an explanation of the workflow.

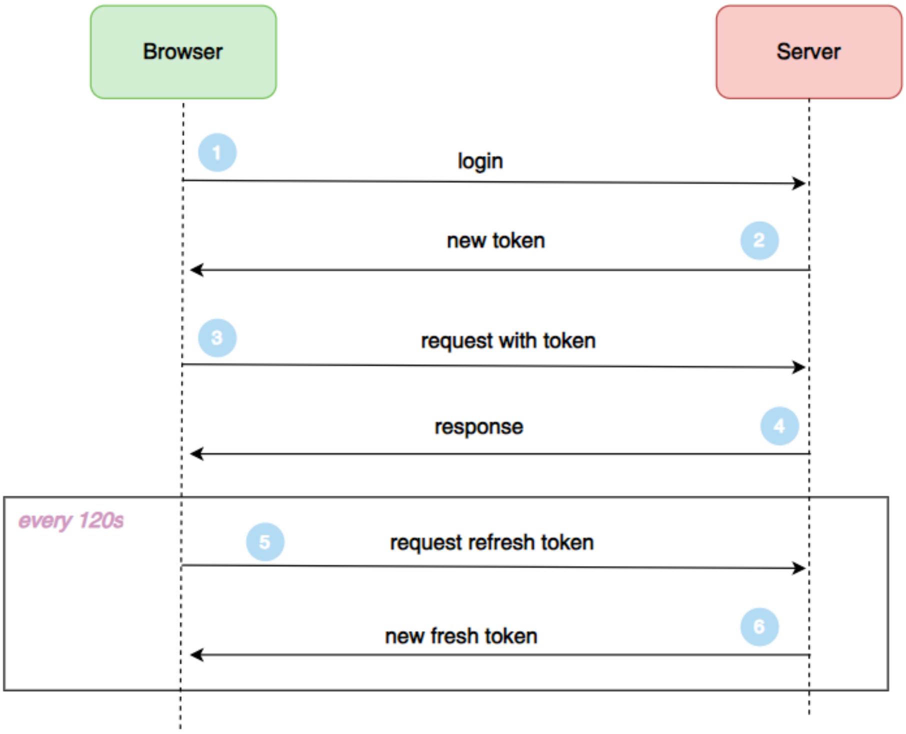


Figure 5.1: Authentication Flow

1. The user sends a request with the login that has the ID and password.

2. The server will process the request and send a response that contains the access and refresh tokens if the user's credentials are valid and a new session is created.

3. The user uses the access token to send requests for data from the resource server.

4. The server processes requests and sends protected resources.

5.The token expires after 120s, so every 120s the browser sends a request to the server for a refresh token using the access token.

6. The server will process the request and send a response that contains the new fresh token.

Chapter 6

Conclusion

In conclusion, the Personnel Deployment Tracker API stands as a pivotal tool addressing the intricate challenges faced by military units in monitoring and managing the deployment status of personnel. By offering a centralized and streamlined solution, this API empowers commanders with real-time information on the location, status, and relevant details of deployed military personnel. The project's significance lies in its ability to simplify coordination and communication, ensuring that commanders can make timely and informed decisions. This not only enhances personnel accountability but also contributes to efficient logistics planning and facilitates quick responses during emergencies. As a straightforward yet powerful tool, the Personnel Deployment Tracker API plays a vital role in improving the overall effectiveness and readiness of military units, showcasing its value as a key component in modern military operations. Looking ahead, there are numerous possible enhancements to explore, including token encoding with data, token hashing for enhanced security, integration with geoencoding APIs for precise location data, the implementation of geofencing features, and the potential inclusion of real-time GPS position IDT devices, further enhancing the API's capabilities and adaptability to evolving military needs.

References

[1] https://carnegie-mec.org/2020/04/29/evolution-of-tunisia-s-military-and-role-of-foreign-security-sector-assistance-pub-81602

[2] ”java spring boot” https://www.ibm.com/topics/java-spring-boot

[3] ”Insomnia” https://docs.insomnia.rest/insomnia/get-started

[4] “Git.” https://git-scm.com/.

[5] ”Swagger” https://en.wikipedia.org/wiki/Swagger(software)