IT325 Web Services Final Project
Personnel Deployment Tracker Api

– Defense Industry –

Student:
Balcem Hassine
BA/IT

Supervisor:
Dr.Montassar Ben Messaoud

January,2024

Table of Contents

Table of Contents

1. Motivation
   - Problem Statement
   - Solution
2. Personnel Development Tracker API ?
   - Features of Personnel Development Tracker API
3. Implementations
   - Technologies used
   - Database Design
   - API Routes
   - Security
4. Conclusion
   - SWOT Analysis:
   - Possible Enhancement

Motivation

Problem Statement

Problem Statement:

In the absence of a centralized solution, military units face challenges in resource allocation, logistics planning, and emergency response coordination.

Motivation

Solution

- By providing a centralized system, this API enables commanders to effortlessly monitor the real-time location, status updates, and relevant details of deployed military personnel.
- The solution facilitates efficient logistics planning, improves personnel accountability and ensures commanders have the necessary information for emergency response coordination.

Table of Contents

1. Motivation
   - Problem Statement
   - Solution
2. Personnel Development Tracker API ?
   - Features of Personnel Development Tracker API
3. Implementations
   - Technologies used
   - Database Design
   - API Routes
   - Security
4. Conclusion
   - SWOT Analysis:
   - Possible Enhancement

Personnel Development Tracker API ?

PersonalDeploymentTracker is an API that provides:

- Tracking of Deployments.
- Conveniently list the enlisted personnel to efficiently schedule future deployments.
- Manage leave requests of the enlisted personnel.
- Manage The skill list of enlisted.

Table of Contents

1. Motivation
   - Problem Statement
   - Solution
2. Personnel Development Tracker API ?
   - Features of Personnel Development Tracker API
3. Implementations
   - Technologies used
   - Database Design
   - API Routes
   - Security
4. Conclusion
   - SWOT Analysis:
   - Possible Enhancement

Implementations

Technologies used

IDE: IntelliJ

- Intelligent Code Assistance
- Seamless Integration
- Rich Plugin Ecosystem

Implementations

Technologies used

Spring Boot:

- Microservices Architecture Support.
- Auto-Configuration and Opinionated Defaults.
- Embedded Database Support.

Implementations
Technologies used

H2 Database:

- Lightweight In-Memory Database.
- JDBC Compatibility
- Web-Based Console.

Implementations
Technologies used

Insomnia:

- Streamlines the Testing phase.
- Ensures reliability and effectiveness.
- Allows for thourough testing and validation of API validations.

Implementations

Technologies used

Swagger :

- Toolset for API Development.
- Generates interactive API documentation.
- Based on the OpenAPI specifications.

Implementations

Database Design

The ERD diagram for the elements in the database:

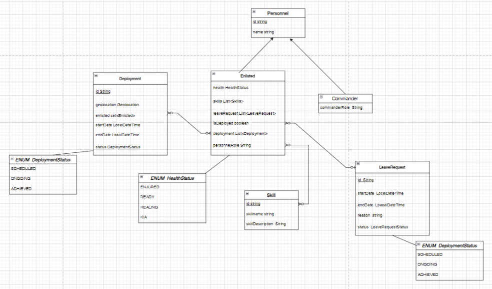
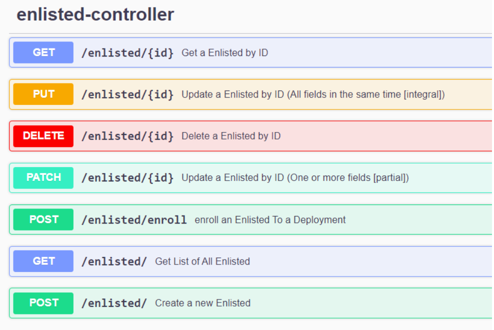


Figure: Entity Relationship Diagram .

Implementations

API Routes

Enlisted Routes :



enlisted-controller

GET /enlisted/{id} Get a Enlisted by ID

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

DELETE /enlisted/{id} Delete a Enlisted by ID

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

POST /enlisted/enroll enroll an Enlisted To a Deployment

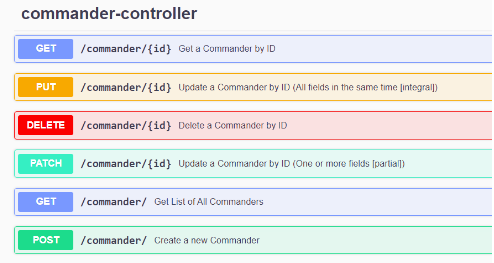
GET /enlisted/ Get List of All Enlisted

POST /enlisted/ Create a new Enlisted

Figure: Enlisted Routes.

Implementations
API Routes

Commander Routes :



commander-controller

GET /commander/{id} Get a Commander by ID

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

DELETE /commander/{id} Delete a Commander by ID

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

GET /commander/ Get List of All Commanders

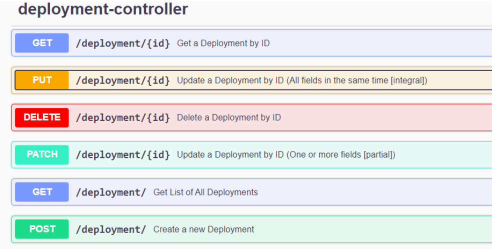
POST /commander/ Create a new Commander

Figure: Commander Routes.

Implementations

API Routes

Deployment Routes :



deployment-controller

GET /deployment/{id} Get a Deployment by ID

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

DELETE /deployment/{id} Delete a Deployment by ID

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

GET /deployment/ Get List of All Deployments

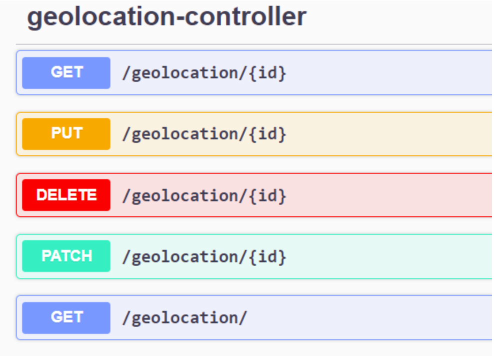
POST /deployment/ Create a new Deployment

Figure: Deployment Routes.

Implementations

API Routes

Geolocation Routes :



geolocation-controller

GET /geolocation/{id}

PUT /geolocation/{id}

DELETE /geolocation/{id}

PATCH /geolocation/{id}

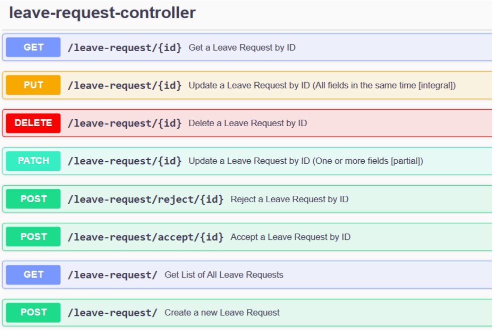
GET /geolocation/

Figure: Geolocation Routes.

Implementations

API Routes

Leave-Request Routes :



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])

DELETE /leave-request/{id} Delete a Leave Request by ID

PATCH /leave-request/{id} Update a Leave Request by ID (One or more fields [partial])

POST /leave-request/reject/{id} Reject a Leave Request by ID

POST /leave-request/accept/{id} Accept a Leave Request by ID

GET /leave-request/ Get List of All Leave Requests

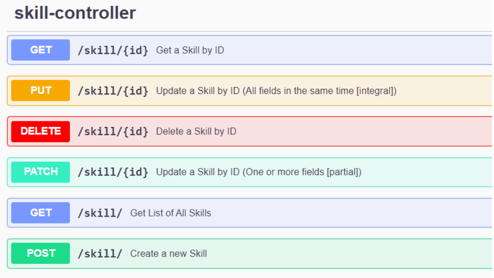
POST /leave-request/ Create a new Leave Request

Figure: Leave-Request Routes.

Implementations

API Routes

Skills Routes :



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

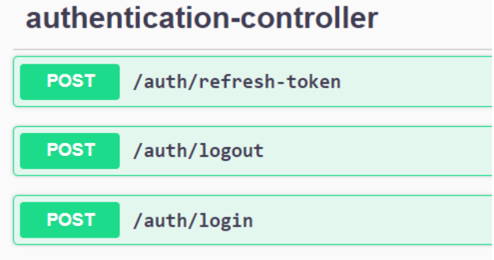
POST /skill/ Create a new Skill

Figure: Skills Routes.

Implementations

API Routes

Authentication Routes :



authentication-controller

POST /auth/refresh-token

POST /auth/logout

POST /auth/login

Figure: Authentication Routes.

Implementations

Security

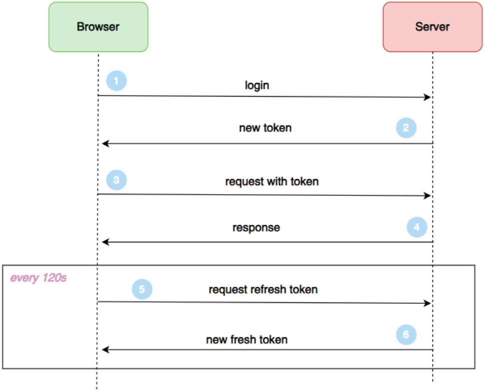


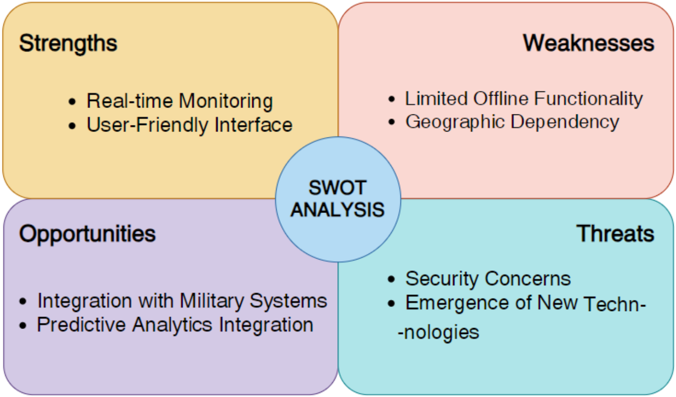
Figure: Authentication Flow.

Table of Contents

1. Motivation
   - Problem Statement
   - Solution
2. Personnel Development Tracker API ?
   - Features of Personnel Development Tracker API
3. Implementations
   - Technologies used
   - Database Design
   - API Routes
   - Security
4. Conclusion
   - SWOT Analysis:
   - Possible Enhancement

Conclusion

SWOT Analysis:



SWOT ANALYSIS

Strengths

- Real-time Monitoring
- User-Friendly Interface

Weaknesses

- Limited Offline Functionality
- Geographic Dependency

Opportunities

- Integration with Military Systems
- Predictive Analytics Integration

Threats

- Security Concerns
- Emergence of New Techn- -nologies

Figure: SWOT Analysis.

Conclusion

Possible Enhancement

- Implementation of advanced security protocols.
- Integration of other APIs (geolocation, Weather,...)
- Improving the project's code by adding more complex and well-designed functions.

Conclusion

In Summary:

- Discussed the need for a centralized system to help the Military Units.
- Presented an API that helps Military units effortlessly monitor the real-time location, status updates, and relevant details of deployed military personnel.
- Revealing the technology behind it.
- Highlighted the strengths, weaknesses, opportunities, and threats of the API and the possible enhancement.

Thanks for the attention !