User control

Overview

Welcome to PlantUML!

You can start with a simple UML Diagram like:

```
Bob->Alice: Hello
```

Or

```
class Example
```

You will find more information about PlantUML syntax on https://plantuml.com

(Details by typing `license` keyword)

```
PlantUML 1.2022.8beta8
[From string (line 3) ]

@startuml
title User Control Context
!include https://raw.githubusercontent.com/adrianvlupu/C4-PlantUML/latest/C4_Context.puml
 Cannot open URL
```

Level 1: System Context diagram

This is a basic Rest API test development to demonstrate knowledge. This application allows you to add, update, delete and consult user information. The API is developed in Java with Spring Boot, using JPA, and is relationally persisted with Postgres SQL. The tests have been carried out using Postman and the user requests are attached. Docker images have been added for the application and for the database. Added entity mapping process to be able to update one or several fields at once. When adding a user, a check is made to only allow people over 18 years of age to register and their CPF must be valid. Unit tests performed with JUnit/Mockito are added.

Scope: A single API software system.

Api

`\Api`

Welcome to PlantUML!

You can start with a simple UML Diagram like:

```
Bob->Alice: Hello
```

Or

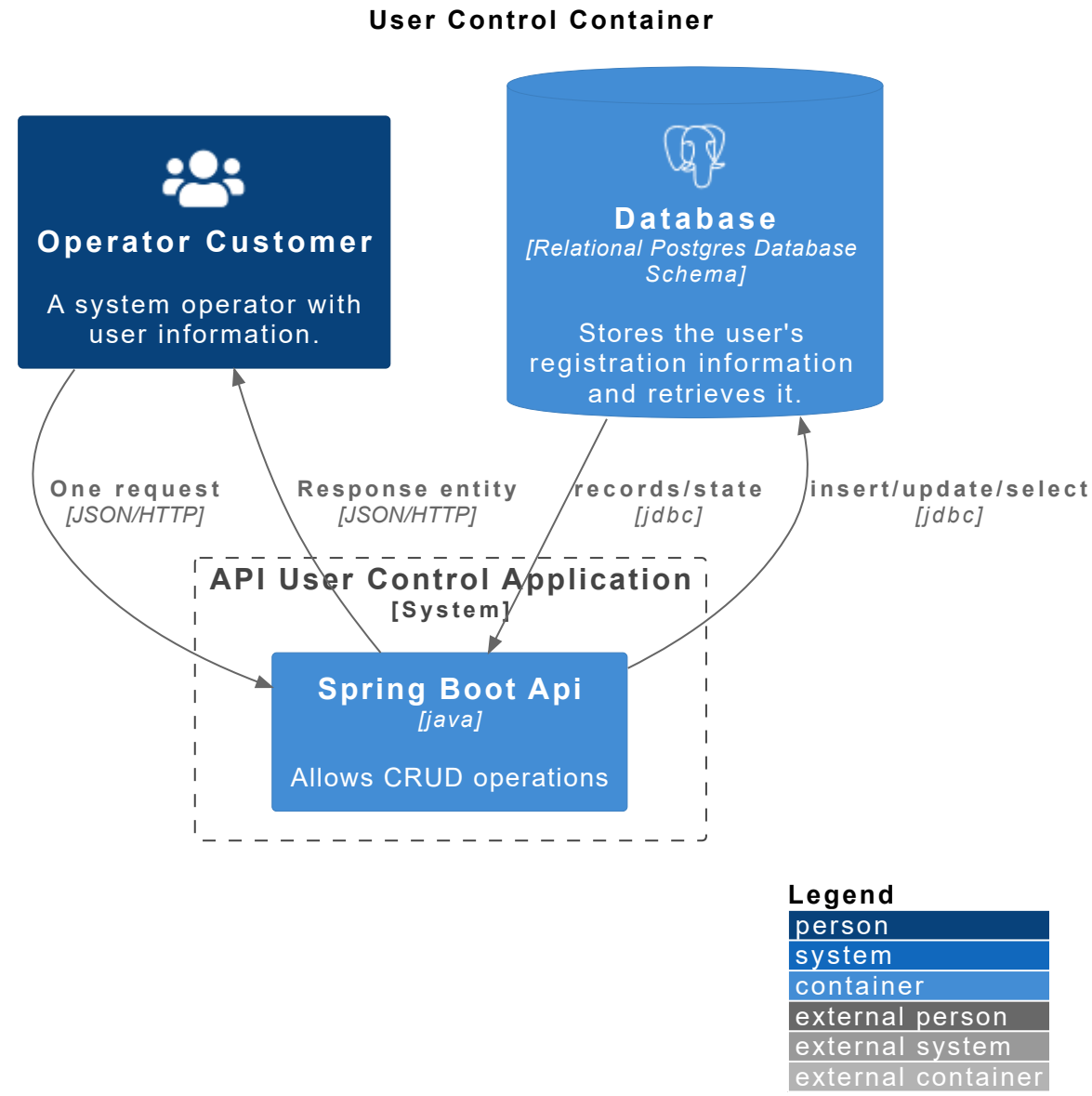
```
class Example
```

You will find more information about PlantUML syntax on https://plantuml.com

(Details by typing `license` keyword)

```
PlantUML 1.2022.8beta8
[From string (line 3) ]

@startuml
title User Control component
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4_Component.puml
 Cannot open URL
```



Level 3: Component diagram

The component diagram shows how a container is made up of a number of "components", what each of those components is, their responsibilities, and the technology/implementation details.

In this diagram, you can visualize the relationship of the various components of the API and their interaction. An operator makes a request to consult, create, delete or update user information. The controller takes the request and forwards it to the service layer. The service is in charge of performing the operation in the database (persistence) or applying some business rule. Subsequently, the information of the operation and the result is returned to the controller, who in turn sends back a response.

Scope: A single container.

Level 2: Container diagram

Este diagrama permite visualizar el contenido y arquitectura de la Api en un segundo nivel

The Container diagram shows the high-level shape of the software architecture and how responsibilities are distributed across it. It also shows the best technology choices and how the containers communicate with one another. It's a simple, high-level technology focused diagram that is useful for software developers and support/operations staff alike.

Scope: A single API software system.