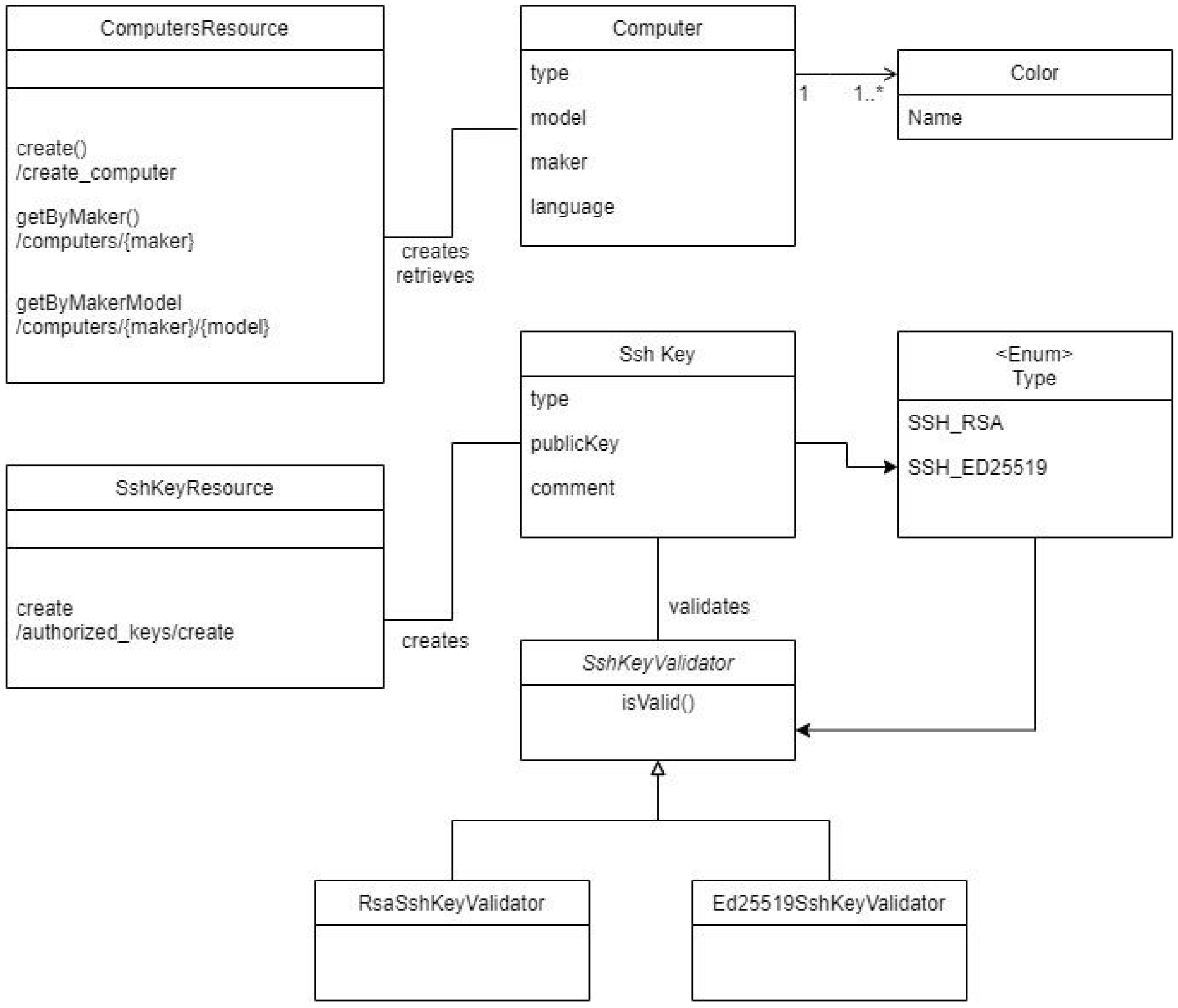
Computers and Keys Project Documentation 🖥️🔑

Overview

Computers and Keys API provides a set of endpoints to IT suppliers and computer creators. Computers API consists of create and retrieve operations to Computers while Keys API registers the user before granting the access to Computers API.

Domain Model

The following diagram highlights the main concepts:



The table below shows the classes and its responsibilities:

Classes	Responsibilities
ComputersResource	A resource responsible for creating and retrieving Computers. This class checks if the http request is valid and returns an appropriate response.
Computer	Computer is a persistent unit.
Color	Color of the computer.
SshKeyResource	A resource responsible for creating SshKey. This class checks if the http request is valid and returns an appropriate response.
SshKey	SshKey is a persistent unit.
Type	Type of the SshKey.
RsaSshKeyValidator	Validates rsa ssh public keys.
Ed25519SshKeyValidator	Validates ed25519 ssh public keys.

Technologies

The project uses:
- Jersey RESTful and so the domain model shows Resource classes which come from Java EE concepts. The resource classes provide endpoints to the users to access our API.
- Jetty server that is provided by Jersey. It is important to note that this is not a fully qualified Java EE container. With this, Entity Management is also the task of the system.
- Maven as its build automation tool.
- PostgreSQL
- Hibernate

Setup

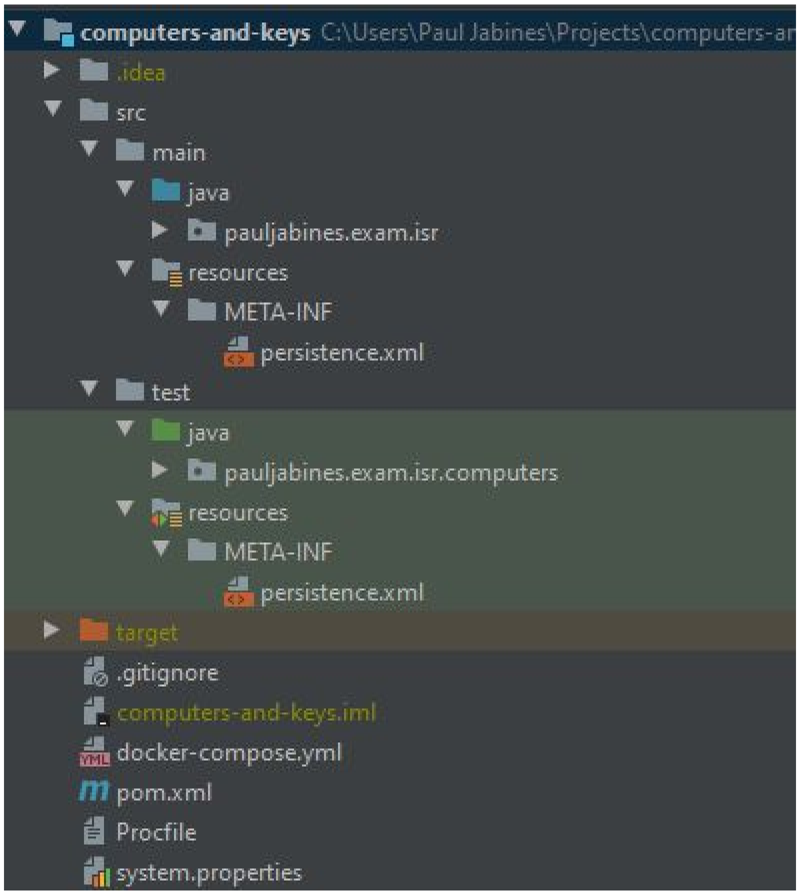
Prerequisites:

- Java 11
- Maven
- Docker desktop
- Git

Note: I am using Windows.

1. Clone the repository, https://github.com/jabinespbi/computers-and-keys, decide what protocol you use.
2. Run the docker desktop
3. Execute the docker-compose.yml which is located at the root of the project
   - docker-compose up -d
4. Cd to the project root and execute the command below:
   - mvn clean package jetty:run

Project Structure



```
computers-and-keys C:\Users\Paul Jabines\Projects\computers-ar
  .idea
  src
    main
      java
        pauljabines.exam.isr
      resources
        META-INF
          persistence.xml
    test
      java
        pauljabines.exam.isr.computers
      resources
        META-INF
          persistence.xml
  target
  .gitignore
  computers-and-keys.iml
  docker-compose.yml
  pom.xml
  Procfile
  system.properties
```

The project structure is a standard java project structure. It consists of:

- *root folder* that contains the docker-compose.yml, pom.xml
- *src/main/java* path that has the java classes
- src/main/resources path that contains resource files i.e. persistence.xml
- *the test folder* has the structure similar to main folder

Development Approach

Git

Commit often. Commit small. Whenever you have something that works especially when the test works, commit your changes.

Tests

Creating tests is a must. This to prevent bugs in the system especially when there are big changes. You can practice Test Driven Development.

Styles and Conventions

1. Class name for the integration test should be in format ClassResourceIntegrationTest.
2. Method names for the integration test classes should be in format methodName_stateUnderTest_ExpectedBehavior.

Refer to google java style guide for the other styles and conventions.