```
# core-java-project-template

A Template that can be used to start a Core Java Project.
In this demo you will find a simple REST Server, based on Javalin.
Tests are written for the services itself as well as for the REST Service.

## What is offered by this template?

* TDD with Junit5
* MutationCoverage with PiTest
* Compile via Dockerimage
* Deployment via Dockerimage
* Development Dockerimage with JDK and Maven
* Production Dockerimage with JDK
* Issuetracker via Github Issues
* Projectplanning via Github Projects
* Create SBOM (cyclonedx)
* Dependency Version Management via versions plugin
* Integration Tests for the REST Server
*

## Vulnerability - Hunting

Even in small projects it is importand to scann for vulnerabilities.
But mostly there is no budget for personal projects.
What should you do? Well, you can combine different free offerings
to see who is reporting faster in wich case. Most vendors are implementing it as Github-PR.
So, see who is fast and what you will get.
I will list a few provider so that you have a solid base to start with.

* Snyk: https://snyk.io/
* OXSecurity: https://app.ox.security/
* FaradaySec: https://faradaysec.com/
```

```

## Todos

* Wie mache ich ein release? -jreleaser?
* Compile in Docker
* Run in Docker - Webservices..
* PiTest in Docker mit Source Snapshot

## Requirements:

- podman: Instead of Docker Desktop I´m using podman.

## How to start

* search and replace inside pom.xml - "https://github.com/svenruppert/core-java-project-template" with your coordinates.
* define what is your JDK you want to use and change it - default is the latest Temurin LTS
  * inside the Docker image definitions
  * inside your pom.xml
* change the properties **pitest-prod-classes** and **pitest-test-classes**
* change the properties for the deployment repositories
* change the repositories, you are resolving from. Default is maven central
* if you have a main class, change the property **app.main.class** or comment it out
* create the docker images under _tools/docker
  * develop/build.sh
  * runtime/build.sh
  * application/build.sh - first time after you created your shaded application.jar

## Docker Images for Develop and Runtime

### Developer Images

Here we are creating an image with JDK and maven (
or gradle if you are using it).

### Best practices

From time to time update the core Images with the latest updates on OS system base.
For this tag the image with the update date, so tht everybody know how old the updated
image is.

```