

# CLAMP Development Guide

- [Setting up your development environment](#)
- [Cloning CLAMP code](#)
- [Project structure](#)
- [Building CLAMP](#)
- [Testing CLAMP](#)

## Setting up your development environment

Clamp is built using maven and is mostly composed of java code. Please follow the instructions under the [Setting your development environment page](#) to prepare your machine for developing clamp.

## Cloning CLAMP code

CLAMP code is located in a single git repository named clamp

it is accessible via HTTPS : <https://gerrit.onap.org/r/a/clamp.git>

or SSH : <ssh://USERNAME@gerrit.onap.org:29418/clamp>

Please make sure to review the ONAP [Development Procedures and Policies](#) to understand how to interact with the gerrit/git repositories.

## Project structure

CLAMP is composed of a main pom.xml and is structured in only one maven module, there is no sub-modules. It's composed of a back-end written in java, and a Designer UI front-end written in HTML/javascript (with angularJS).

The back-end code is stored in the standard java locations `"/src/main/java"`, `"src/main/resources"`. And all the back-end tests are located in `"src/test/java"`, `"src/test/resources"`.

The front-end code is stored in a sub folder of the back-end code in `"src/main/resources/META-INF/resources"`.

### Unit test

The unit tests are composed of 2 different types:

1. *The standard unit tests*, testing obviously the class and methods
2. *The Spring integration tests* that really starts Clamp (using `@SpringBootTest`) and that can make use of everything in the code, like the database, the http connection, etc ...  
Those tests must have a specific filename pattern: `**ItCase.java`

The code coverage is done by Jacoco and the reports are configured to provide unit tests and integration coverage.

### Location details

Path	Content
<b>Docs</b>	
<code>/docs</code>	The documentation in .rst files
<b>Docker</b>	
<code>/extra/docker/clamp</code>	The docker compose file to start clamp and the Mariadb
<code>/extra/docker/mariadb</code>	The mariadb conf files used by docker compose
<code>/extra/sql</code>	The SQL file containing the CLAMP schema, Clamp Stored procedures and Camunda schema
<code>/src/main/docker</code>	The Docker file + the scripts to start clamp that are used to build the image
<b>Code</b>	
<code>/src/main/java</code>	The back-end and front-end code + resources
<code>/src/main/test</code>	The unit tests and Spring integration tests to validate CLAMP code
<code>/src/main/scripts</code>	Groovy scripts used by maven build
<code>/src/main/resources</code>	The default CLAMP configuration files + the Front end code

/src/main/resources/clds	The Clamp specific configuration files
/src/main/resources/xsl	The xsl used to convert BPMN XML to JSON (from designer UI)
/src/main/resources/META-INF/resources	The front end code
/src/main/resources/META-INF/resources/designer	The designer front code (HTML + javascript)

## Building CLAMP

To build CLAMP you need to use maven with the goal "clean install". By default it executes the standard unit tests and the Spring integration tests *but does not build the docker images*.

You can disable the integration tests by executing: "mvn clean install -DskipITs=true"

There are one profile to build the docker image named "docker", to build the docker image: "mvn clean install -P docker"

## Testing CLAMP

We created CSIT (Continuous System and Integration Testing) tests for clamp as requested for each ONAP component. This use Robot Framework. See this wiki page for more information: [Creating a CSIT Test](#)

The tests are created in the integration repository: <https://gerrit.onap.org/r/#/admin/projects/integration>

For Clamp we have created 2 test plans that test:

- The APIs to create templates, close loops,...
- The UI to login, click on menus, create templates or close loops manually, set properties,...

This is testing Clamp standalone and thus cannot validate interactions with other components (sdc, policy...). This still require some manual E2E testing.