# CryptoCurrency

This is a project template for AEM-based applications. It is intended as a best-practice set of examples as well as a potential starting point to develop your own functionality.

### Prerequisites

What things you need to install the software and how to install them:

1. JAVA 8

2. Maven

3. Copy settings.xml provided as a part of the project to maven\_home/conf folder

4. AEM 6.4

## Modules

The main parts of the template are:

\* core: Java bundle containing all core functionality like OSGi services, listeners or schedulers, as well as component-related Java code such as servlets or request filters.

\* ui.apps: contains the /apps (and /etc) parts of the project, ie JS&CSS clientlibs, components, templates, runmode specific configs as well as Hobbes-tests

\* ui.content: contains sample content using the components from the ui.apps

\* ui.tests: Java bundle containing JUnit tests that are executed server-side. This bundle is not to be deployed onto production.

\* ui.launcher: contains glue code that deploys the ui.tests bundle (and dependent bundles) to the server and triggers the remote JUnit execution

## How to build

To build all the modules run in the project root directory the following command with Maven 3:

mvn clean install

If you have a running AEM instance you can build and package the whole project and deploy into AEM with

mvn clean install -PautoInstallPackage

Or to deploy it to a publish instance, run

mvn clean install -PautoInstallPackagePublish

Or alternatively

mvn clean install -PautoInstallPackage -Daem.port=4503

Or to deploy only the bundle to the author, run

mvn clean install -PautoInstallBundle

## Verifying the application functionality

Go to below url if you have installed AEM locally:

http://localhost:4502/content/CryptoCurrency/en/CryptoHome.html

From here the application can be tested and user can navigate to the remaining pages of the application and navigate back to the home page.

## Testing

There are three levels of testing contained in the project:

\* unit test in core: this show-cases classic unit testing of the code contained in the bundle. To test, execute:

mvn clean test

\* server-side integration tests: this allows to run unit-like tests in the AEM-environment, ie on the AEM server. To test, execute:

mvn clean verify -PintegrationTests

\* client-side Hobbes.js tests: JavaScript-based browser-side tests that verify browser-side behavior. To test:

in the browser, open the page in 'Developer mode', open the left panel and switch to the 'Tests' tab and find the generated 'MyName Tests' and run them.

## Maven settings

The project comes with the auto-public repository configured. To setup the repository in your Maven settings, refer to:

http://helpx.adobe.com/experience-manager/kb/SetUpTheAdobeMavenRepository.html