

# Build from source

eHour is mainly developed in Java 7 and Scala. As of eHour 1.4.2, the intention is not to add any new Scala code but quickly migrate to Java 8 for quicker compile times, better IDE support, readable syntax and stacktraces that don't an enigma to decypher.

## Architecture

eHour has a classic 3-tier architecture with a persistence layer (eHour-persistence), service layer (eHour-service) and view layer (eHour-wicketweb).

## Complete stack

- Hibernate
- Spring
- Apache Wicket
- AOP
- JQuery
- Highcharts
- Selenium
- Maven
- Java 7
- Scala

## Source

eHour's source is versioned at Github: <https://github.com/te-con/ehour>

Gitflow is loosely followed: the current released version is in master while the version being developed is in the develop branch. The nightly builds are build from the develop branch. Any features that take a considerable amount of time to develop lives in a feature/<feature name> branch. After the feature is finished that branch is merged into the develop branch.

## Modules

The code is divided in a number of modules:

Maven module name	Description
eHour-audit	Contains the aspect for the audit log. Only one annotation is in this module but the annotation is used by the service and web tier
eHour-common	Contains common functionality and all the domain objects. Domain objects are shared between the 3 tiers.
eHour-jetty	The embedded Jetty server as used by the standalone version and the selenium end-to-end tests
eHour-persistence	The data access objects for all supported databases, the Derby (embedded DB) validator and Derby native SQL queries.
eHour-persistence-mysql	The MySQL native SQL queries plus MySQL SQL scripts.
eHour-persistence-postgresql	Ditto for Postgresql
eHour-selenium	End-to-end tests with Selenium
eHour-service	The service tier with all the business logic
eHour-standalone	The assembly package and install4j scripts for building the standalone executables
eHour-war	The assembly package for building the WAR
eHour-web	Contains static HTML, CSS and Javascript
eHour-wicketweb	The view tier

The reason for splitting out eHour-web and eHour-wicketweb is the different packaging for the standalone and WAR distribution.

## Building with Maven

To build with Maven 3, use the standard "mvn clean install".

## Profiles

Apart from the default build there are two profiles:

Profile	Execution	Description
prod	mvn clean install -Pprod	Builds the ehour-<version>-war.zip. The standalone executables are built with install4j 5 however the license file cannot be shared.
e2e	mvn clean install -Pe2e	Builds and executes the end-to-end Selenium tests.

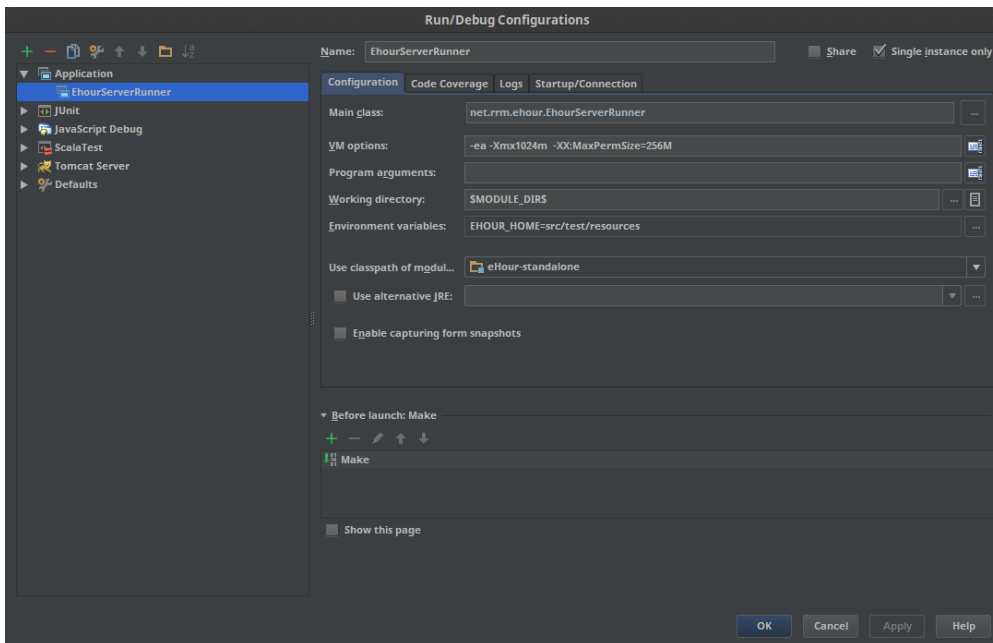
## Speeding up compile time

The Scala classes take a while to compile. To speed up compile time over multiple iterations, you might want to run a Zinc compile server. Download Zinc from <https://github.com/typesafehub/zinc> and run it in before compiling with Maven. In Maven's output you will see a message that the Zinc compile server is being used.

## Developing in IntelliJ

Clone the source and open the POM file as a project. For Scala support you want to have the Scala plugin installed.

To run eHour from IntelliJ, create a runtime configuration like:



## Developing in Eclipse

todo