

# Courses Management

Technical reference paper

2012 DEViati Team - University of Trento



# Courses Management technical reference for Developers and Sysadmins

This document is a simple reference for developers and sysadmins who want to start working with our Courses app. Here follows few things you should know:

## Development methodology

- we use Git source control management system for the development; it currently resides on Bitbucket (<https://bitbucket.org/hohenstaufen/courses/>) as a private repository, but we can make it public if needed. Bitbucket provides a ticketing system (for task management) as well as a wiki (so we have all these documentation also in the project wiki);
- we also have a Jenkins installation for the monitoring and management of continuous integration of our software; you can check it on <https://ci.ahohenstaufen.net/> (user: *supervisor* password: *checkthis*);
- all our code is tested using the JUnit testing framework; tests are automatically runned by Jenkins every push on our repository;
- we use PivotalTracker as an “high level user stories” tracker, and Bitbucket ticketing system for “technical” tasks;

## Technologies

- Courses is developed in Groovy, using the Grails framework;
- we use MySQL as DBMS;
- for the client side, we chose to use Twitter Bootstrap; for its customization, we use SASS-SCSS based stylesheets; we also made strong use of Ajax;

## Setting up the development environment

To start developing our Courses Grails app, you need to have some dependencies installed first; you will absolutely need:

- Java JDK (we recommend the Oracle version, but OpenJDK should work too)  
Reference version: 1.7.0\_04
- Groovy  
Reference version: 1.8.6
- Grails (version 2.0.3 is out since few weeks ago, we haven't made the upgrade yet)  
Reference version: 2.0.1

then we have some optional dependencies:

- JRuby (used by a Grails plugin to automatically compile our css when running the app)  
Reference version: 1.6.6
- Mysql (if you use an existing installation of Mysql, you do not need that)  
Reference version: 5.5.24

you can use your preferred IDE, but some of us have used NetBeans (7.1.2), and found that you can develop flawlessly with this one. We do **not** recommend Eclipse.

**Note:** reference version doesn't mean minimum version, we worked with different OS's and version of software (someone with a Mac and the builtin JVM, for example, some others with Arch Linux, Gentoo and Ubuntu and the software version that comes from the corresponding repositories). The only package that could cause problem (if you haven't installed the right version) is Grails itself; if you want to use a more up-to-date release, you should upgrade the Courses app (a simple **grails upgrade** should be enough, but we haven't tested it yet). Otherwise simply install the 2.0.1 version of Grails; follow this Gist (<https://gist.github.com/901884>) to install Grails on Ubuntu; if you use Arch Linux simply install Grails from the community repository, and then downgrade it using **downgrade** tool from the AUR.

## Deploying the application

### Setting up for production

Grails supports the concept of environments (reference: <http://www.grails.org/Environments>). The default environments are `test`, `development` and `production`.

All the settings are defined in Grails configuration folder (`courses/grails-app/conf/`); the most relevant files here are `DataSource.groovy` and `Config.groovy` (but also others may be of interest); in each of these files you can find different configuration sections, named by the environment they are related to. Customizing these section will reflect changes when you run the app for the environment you specify. The setup of the database connection is defined inside the file `DataSource.groovy` (in the configuration folder). If you want to customize it for production, simply change the section `production` (you can use the `development` section as an example, or check out the official documentation here: [http://grails.org/Quick+Start#Configure a Data Source \(Optional\)](http://grails.org/Quick+Start#Configure+a+Data+Source+(Optional)) , but its pretty simple, if you are familiar with JDBC). Once you have customized the settings, you are ready for packaging.

### Packaging

Packaging your Grails application is an easy task; simply run `grails [envname] war` and Grails will generate a WAR file in the `target/` folder of your application. If you want to package the application for your production environment, use `grails prod war`.

### Deployment

Once you have packaged your app, you are ready for deployment. We have tested Courses with Tomcat 7 and it works perfectly, but you can also use a different application server (reference: <http://grails.org/Deployment>).

To deploy Courses on Tomcat, simply copy the WAR file into `TOMCAT_HOME/webapps` folder and restart the container. Be aware that on default settings, Grails will attempt to write to the tomcat directory certain files, typically the `stacktrace.log` and the h2 database. Your Tomcat directory is likely installed with owner/group of root, while the Tomcat server runs as user tomcat6. This will cause your application to fail to start with write permission errors. Either change your Grails production settings so that these files are written elsewhere to directories with correct permissions OR change the owner of the Tomcat directory (for Unix: `/var/lib/tomcat6`) to tomcat6 (or whatever tomcat user you set up using the `TOMCAT6_USER` setting).

### Loading initial data

We have prepared some sample data to show you how the Courses application works. It is a simple sql dump, so you have to load it into the database; if you have a Phpmyadmin installation you can load it by the web interface; otherwise, use this shell command:

```
mysql -u $username -p $password -h $host $dbname < sample.sql
```



## Contacts

You can contact the DEViati team via email at [softeng@lists.ahohenstaufen.net](mailto:softeng@lists.ahohenstaufen.net) or one of our members:

Federico Scrinzi ([fox91@anche.no](mailto:fox91@anche.no))

Andrea Nodari ([andrea.nodari91@gmail.com](mailto:andrea.nodari91@gmail.com))

Alberto Buratti ([alberthohenstaufen@gmail.com](mailto:alberthohenstaufen@gmail.com))

Lorenzo Rigato ([lore91tanz@gmail.com](mailto:lore91tanz@gmail.com))

Giacomo Zanesco ([jack.zanesco@gmail.com](mailto:jack.zanesco@gmail.com))

## Technical Reference Paper License

Courses Management Technical Reference Paper by Federico Scrinzi, Andrea Nodari, Alberto Buratti, Lorenzo Rigato, Giacomo Zanesco is licensed under a [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).

