

Enhance Your Workflow for Spring Boot App development with DevTools and LiveReload

1. For a new Spring Boot project, simply include DevTools in your dependencies selection. For an existing project, simply add the spring-boot-devtools dependency to the pom, as:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
</dependency>
```

2. If using IntelliJ IDEA, enable the following 2 settings and restart the IDE:
 - a. Build, Execution, Deployment -> Compiler -> Make Project Automatically.
 - b. Open the Options/Actions dialog by pressing Shift+Ctrl+A and search for 'Registry'. In the Registry options dialog, look for 'compiler.automake.allow.when.app.running'. And check the box to enable it.
 - c. Restart IntelliJ.
3. Spring Boot DevTools also provides a LiveReload server, which gets started along with the WebContainer and Spring context. And talks to the client browser via websocket. We can use this to avoid needing to manually refresh the browser whenever our source files are updated. To do this, we do the following:
 - a. Install a Google Chrome extension named 'LiveReload' (simply, open the Chrome WebStore and search 'LiveReload' and Add it to Chrome).
 - b. For Mozilla Firefox, use this - <https://addons.mozilla.org/de/firefox/addon/livereload-web-extension/>
4. To start using the LiveReload and auto-restart spring-context devtools, simply start the Spring Boot application in your IDE, go to the url with your browser, click on the icon to enable the LiveReload function and viola.
5. Changes to any static content (e.g. html markup, css, js etc) are automatically refreshed and visible in the browser. And changes to compiled java source code and template files (e.g. thymeleaf templates) are reloaded in the SpringContext and pushed to and will also be visible in the browser, without any need to click browser's Refresh button.
6. //-- The End --//