

Development Mode

Spring Boot Devtools

Spring-Boot-Devtools

- Add spring-boot-devtools like a starter to your project
- Do a clean build of your project to enable it

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

Property Defaults

- Spring Boot enables a lot of defaults e.g. caching
- During development immediate feedback is needed
- Defaults could be overridden manually with properties
 - e.g. `spring.thymeleaf.cache = false`
- `spring-boot-devtools` overrides many properties defaults automatically

Automatic Restart

- File changes in classpath restarts the application
 - Not Main but restartedMain Thread is restarted
- Changes on static files and view templates do not need a restart
- Reducing time considerably to verify changes
 - Code and configuration

Live Reload

Live Reload

- Activates embedded LiveReload server
- Triggers browser refresh when resource is changed
- Browser needs LiveReload plugin
 - E.g. RemoteLiveReload for Google Chrome
- Only one LiveReload server can be started. If multiple Applications are running, only first has LiveReload support



Global Settings

- Add `.spring-boot-devtools.properties` to `$HOME`
 - note that the filename startswith ". "
- All properties added to file apply to all Spring Boot applications which use devtools

```
spring.devtools.restart.enabled=false  
spring.devtools.reload.trigger-file=.reloadtrigger  
spring.devtools.livereload.enabled=false
```


Remote Applications

Remote Applications

- Running remote client application
- Remote update via HTTP
- Remote Java debugging
- Remote debug tunnel via HTTP