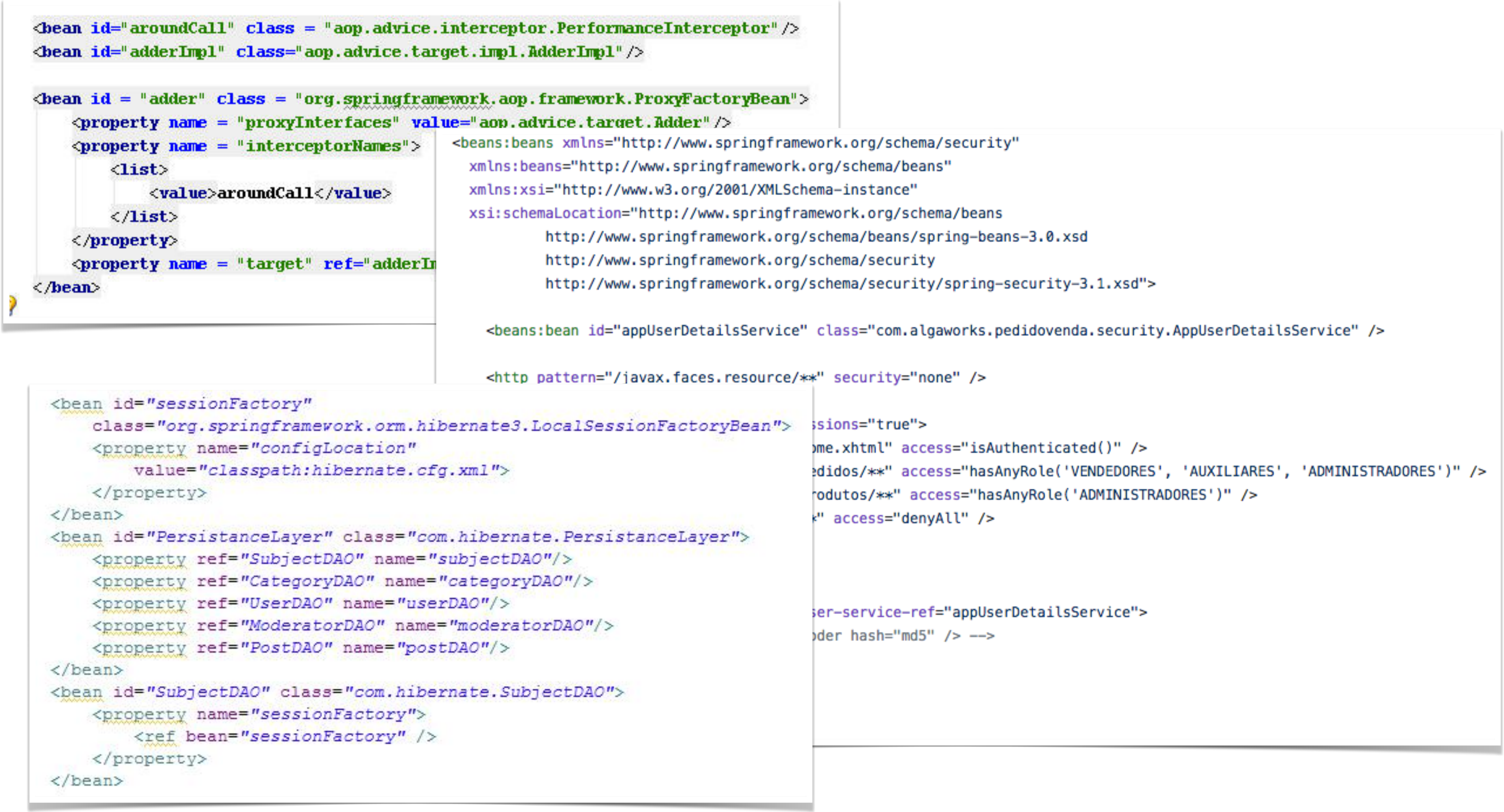
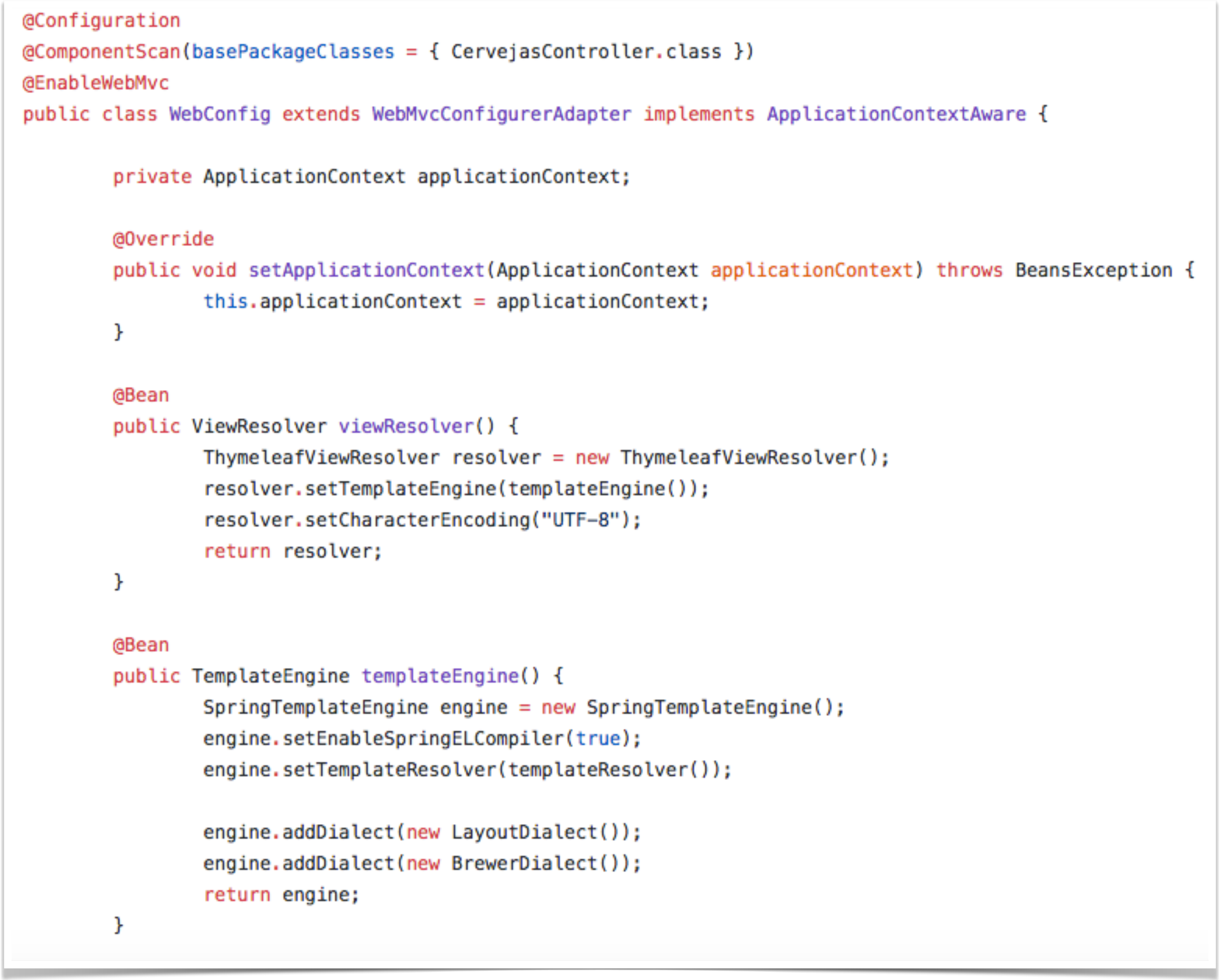
Conhecendo o Spring Boot



```
<bean id="aroundCall" class = "aop.advice.interceptor.PerformanceInterceptor" />
<bean id="adderImpl" class="aop.advice.target.impl.AdderImpl" />

<bean id = "adder" class = "org.springframework.aop.framework.ProxyFactoryBean">
    <property name = "proxyInterfaces" value="aop.advice.target.Adder" />
    <property name = "interceptorNames">
        <list>
            <value>aroundCall</value>
        </list>
    </property>
    <property name = "target" ref="adderI
</bean>
```

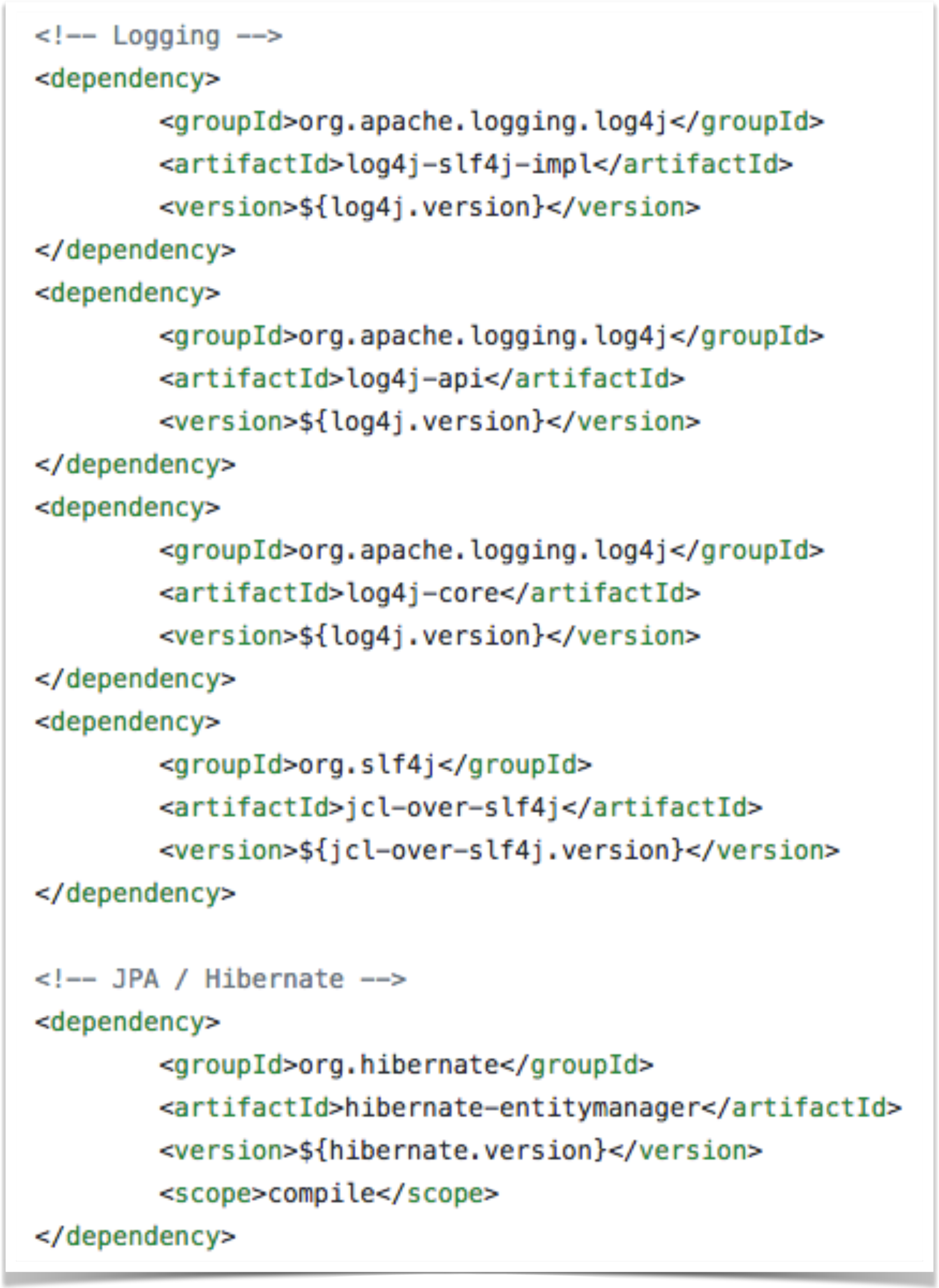
```
<beans:beans xmlns="http://www.springframework.org/schema/security"
  xmlns:beans="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
          http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
          http://www.springframework.org/schema/security
          http://www.springframework.org/schema/security/spring-security-3.1.xsd">

    <beans:bean id="appUserDetailsService" class="com.algaworks.pedidovenda.security.AppUserDetailsService" />

    <http pattern="/javax.faces.resource/**" security="none" />

ssions="true">
ome.xhtml" access="isAuthenticated()" />
edidos/**" access="hasAnyRole('VENDEDORES', 'AUXILIARES', 'ADMINISTRADORES')" />
rodutos/**" access="hasAnyRole('ADMINISTRADORES')" />
*" access="denyAll" />



ser-service-ref="appUserDetailsService">
oder hash="md5" /> -->
```

```
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="configLocation"
        value="classpath:hibernate.cfg.xml">
    </property>
</bean>
<bean id="PersistanceLayer" class="com.hibernate.PersistanceLayer">
    <property ref="SubjectDAO" name="subjectDAO"/>
    <property ref="CategoryDAO" name="categoryDAO"/>
    <property ref="UserDAO" name="userDAO"/>
    <property ref="ModeratorDAO" name="moderatorDAO"/>
    <property ref="PostDAO" name="postDAO"/>
</bean>
<bean id="SubjectDAO" class="com.hibernate.SubjectDAO">
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
</bean>
```

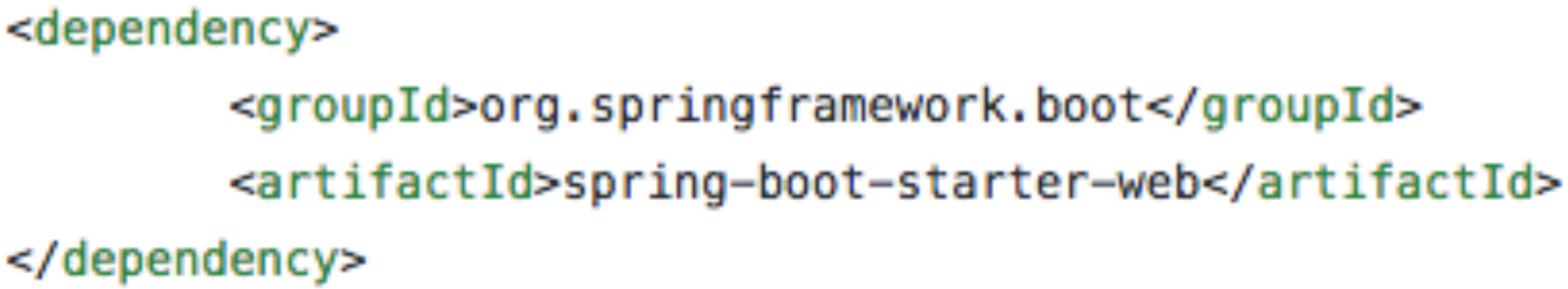


```java
@Configuration
@ComponentScan(basePackageClasses = { CervejasController.class })
@EnableWebMvc
public class WebConfig extends WebMvcConfigurerAdapter implements ApplicationContextAware {

        private ApplicationContext applicationContext;

        @Override
        public void setApplicationContext(ApplicationContext applicationContext) throws BeansException {
                this.applicationContext = applicationContext;
        }

        @Bean
        public ViewResolver viewResolver() {
                ThymeleafViewResolver resolver = new ThymeleafViewResolver();
                resolver.setTemplateEngine(templateEngine());
                resolver.setCharacterEncoding("UTF-8");
                return resolver;
        }

        @Bean
        public TemplateEngine templateEngine() {
                SpringTemplateEngine engine = new SpringTemplateEngine();
                engine.setEnableSpringELCompiler(true);
                engine.setTemplateResolver(templateResolver());

                engine.addDialect(new LayoutDialect());
                engine.addDialect(new BrewerDialect());
                return engine;
        }
```

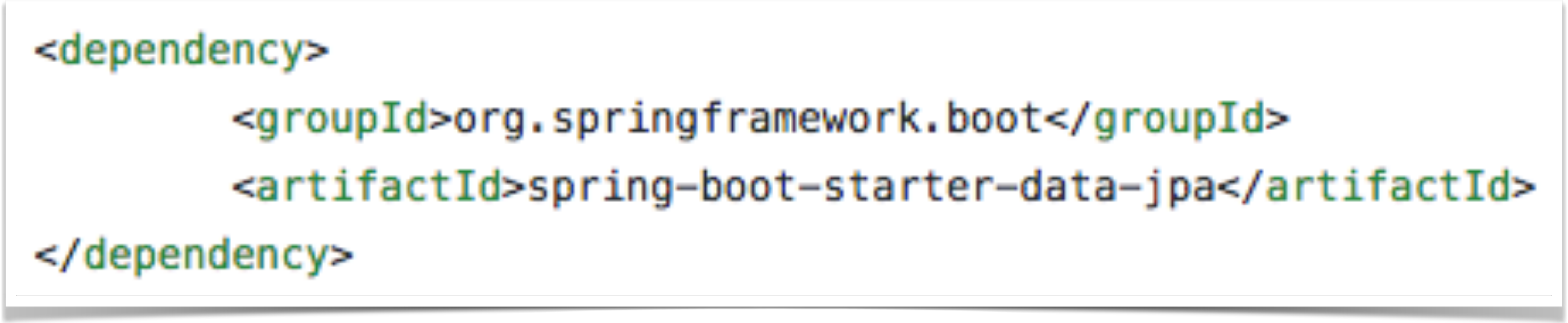
Spring Boot



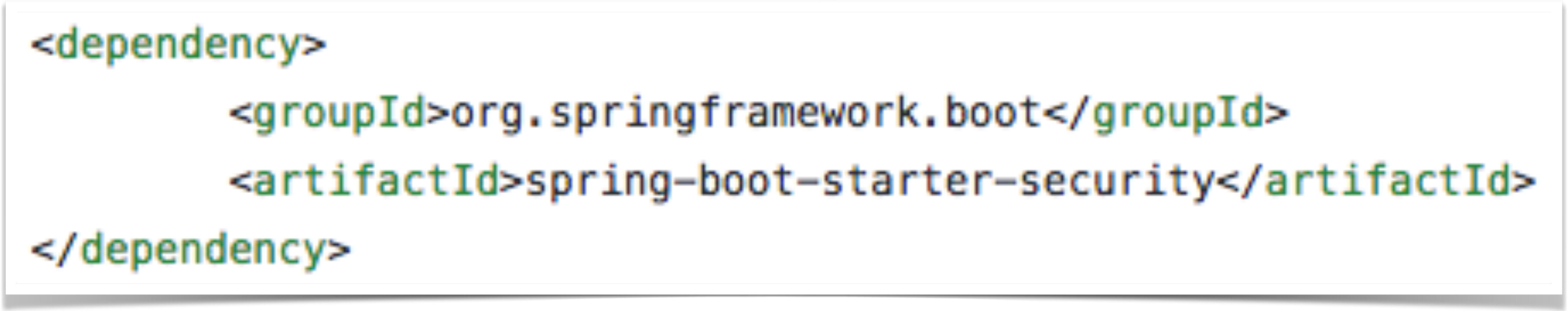
```
<!-- Logging -->
<dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-slf4j-impl</artifactId>
        <version>${log4j.version}</version>
</dependency>
<dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-api</artifactId>
        <version>${log4j.version}</version>
</dependency>
<dependency>
        <groupId>org.apache.logging.log4j</groupId>
        <artifactId>log4j-core</artifactId>
        <version>${log4j.version}</version>
</dependency>
<dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>jcl-over-slf4j</artifactId>
        <version>${jcl-over-slf4j.version}</version>
</dependency>

<!-- JPA / Hibernate -->
<dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-entitymanager</artifactId>
        <version>${hibernate.version}</version>
        <scope>compile</scope>
</dependency>
```



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



```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```



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

Spring Boot ou Spring MVC?