Spring Boot



Spring konfiguracja

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:p="http://www.sprinaframework.ora/schema/p"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:mvc="http://www.sprinaframework.ora/schema/mvc"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context
                           http://www.springframework.org/schema/context/spring-context.xsd
                           http://www.springframework.org/schema/mvc
                           http://www.springframework.org/schema/mvc/spring-mvc.xsd">
    <context:component-scan base-package="org.springbyexample.web.servlet.mvc" />
    <mvc:annotation-driven />
    <mvc:view-controller path="/index.html" />
    <bean id="tilesConfigurer"</pre>
          class="org.springframework.web.servlet.view.tiles2.TilesConfigurer"
          p:definitions="/WEB-INF/tiles-defs/templates.xml" />
    <bean id="tilesViewResolver"</pre>
          class="org.springframework.web.servlet.view.UrlBasedViewResolver"
          p:viewClass="org.springbyexample.web.servlet.view.tiles2.DynamicTilesView"
          p:prefix="/WEB-INF/jsp/"
          p:suffix=".jsp" />
    <bean id="messageSource" class="org.springframework.context.support.ResourceBundLeMessageSource"</pre>
          p:basenames="messages" />
    <mvc:interceptors>
        <bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor"</pre>
              p:paramName="locale" />
    </mvc:interceptors>
    <bean id="localeResolver" class="org.springframework.web.servlet.i18n.SessionLocaleResolver" />
</beans>
```

Live coding



Spring boot – jak to działa

@Conditional
@Configuration
public class MessageSourceAutoConfiguration {



Conditional - przykład

```
@Configuration
@ConditionalOnClass(EmbeddedDatabaseType.class)
public class DataSourceAutoConfiguration {
   @Conditional(DataSourceAutoConfiguration.EmbeddedDatabaseCondition.class)
   @ConditionalOnMissingBean(DataSource.class)
   protected static class EmbeddedConfiguration {
      @Bean
       public DataSource dataSource() {
           EmbeddedDatabaseBuilder builder = new EmbeddedDatabaseBuilder()
                      .setType(EmbeddedDatabaseConnection.get().getType());
          this.database = builder.setName(this.name).build();
          return this.database;
```

Live coding



Spring Boot





AngularJS + Bootstrap





AngularJS

Two way data binding

• MV*

• DI

Directive

Testing

Live coding



AngularJS

- AngularJS + Bootstrap = AngularUI
- Router → UI Router
- Batarang
- AngularFire
- IE 9+





