Spring MVC 4 REST and Ajax

Bryan Hansen
twitter: bh5k
http://www.linkedin.com/in/hansenbryan

pluralsight
hardcore dev and IT training

REST

ContentNegotiatingViewResolver

```xml
<bean class="org.springframework.web.servlet.view.ContentNegotiatingViewResolver">
    <property name="order" value="1" />
    <property name="contentNegotiationManager">
        <bean class="org.springframework.web.accept.ContentNegotiationManager">
            <constructor-arg>
                <bean class="org.springframework.web.accept.PathExtensionContentNegotiationStrategy">
                    <constructor-arg>
                        <map>
                            <entry key="json" value="application/json" />
                            <entry key="xml" value="application/xml" />
                        </map>
                    </constructor-arg>
                </bean>
            </constructor-arg>
        </bean>
    </property>

    <property name="defaultViews">
        <list>
            <bean class="org.springframework.web.servlet.view.json.MappingJacksonJsonView" />
            <bean class="org.springframework.web.servlet.view.xml.MarshallingView">
                <constructor-arg>
                    <bean class="org.springframework.oxm.xstream.XStreamMarshaller">
                        <property name="autodetectAnnotations" value="true" />
                    </bean>
                </constructor-arg>
            </bean>
        </list>
    </property>
</bean>
```

@**RestController**

- Convenience annotation for annotating a Controller and ResponseBody at the same time

- Assumes each RequestMapping is going to return the body of it's request

- Less verbose than having to define each method and it's response

- No longer need to use the ContentNegotiatingViewResolver
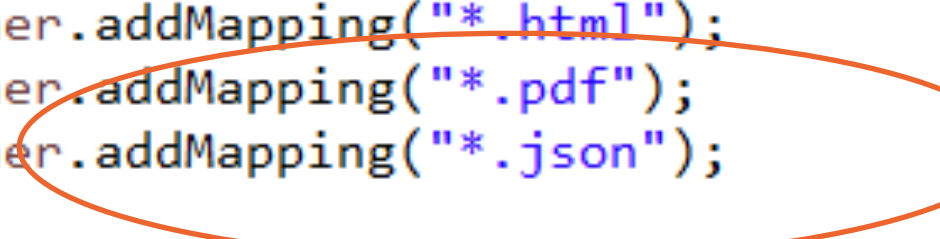
{ JSON }

Angular Client

- Consume a RESTful service using an Angular client

- Bind to JSON response

- Lightweight binding, pure HTML

DispatcherServlet

- DispatcherServlet needs to allow access for JSON (and XML) request types

```java
@Override
public void onStartup(ServletContext servle
    WebApplicationContext context = getCont
    servletContext.addListener(new Contextl
    ServletRegistration.Dynamic dispatcher
    dispatcher.setLoadOnStartup(1);
    dispatcher.addMapping("*.html");
    dispatcher.addMapping("*.pdf");
    dispatcher.addMapping("*.json");
}
```

Summary

- ContentNegotiatingViewResolver

- @RestController

- Angular Client

- DispatcherMapping