Spring Boot

More Spring, Less Configuration

Craig Walls
craig@habuma.com
Twitter: @habuma @SpringSocial
http://github.com/habuma

Spring: Toward Simpler Configuration

Spring 1.0: XML configuration (DTD)

Spring 1.1: `value` and `ref` attributes

Spring 2.0: XML namespaces

Spring 2.5: @Component / @Autowired

Spring 3.0: Java configuration

Spring 3.1: Profiles

Spring 3.2: –

Spring 4.0: Conditional beans

Spring: Toward Simpler Configuration

No Configuration

Introducing Spring Boot

Takes away boilerplate configuration

Heavily opinionated
(but allows you to override opinions)

Can run with Groovy at the command line

Can build to JAR or WAR

Spring Boot: CLI

Runs Groovy apps from the command line

Starts an embedded Tomcat (or Jetty)

Heavily employs Spring Boot autoconfiguration

Spring Boot: Auto-Configuration

Automatically...

...creates beans as needed

...resolves dependencies (CLI)

...adds imports (CLI)

Under the covers: Conditional

```
@Configuration
public class MyConfig {

  @Bean
  @Conditional(MagicExistsCondition.class)
  public MagicBean magicBean() {
    return new MagicBean();
  }

}
```

```
public class MagicExistsCondition implements Condition {

  public boolean matches(ConditionContext context, AnnotatedTypeMetadata metadata) {
    Environment env = context.getEnvironment();
    return env.containsProperty("magic");
  }
}
```

(Spring 4.0.0.M1 and up)

Under the covers: Conditional

```
@Configuration
public class MyConfig {



             l(MagicExistsCondition.class)
             cBean magicBean() {
             v MagicBean();



}
```

ConditionContext offers access to...
- The BeanFactory
- The ClassLoader
- The Environment
- The BeanDefinitionRegistry
- The ResourceLoader

```
public class MagicExistsCondition implements Condition {

  public boolean matches(ConditionContext context, AnnotatedTypeMetadata metadata) {
    Environment env = context.getEnvironment();
    return env.containsProperty("magic");
  }
}
```

(Spring 4.0.0.M1 and up)

Under the covers: Conditional

```
@Configuration
public class MyConfig {

  @Bean
  @Conditional(MagicExistsCondi
  public MagicBean magicBean()
    return new MagicBean();
  }

}
```

AnnotatedTypeMetadata offers access to...
- All annotation attributes
- Whether or not the underlying type has been annotated with a given annotation

```
public class MagicExistsCondition implements Condition {

  public boolean matches(ConditionContext context, AnnotatedTypeMetadata metadata) {
    Environment env = context.getEnvironment();
    return env.containsProperty("magic");
  }
}
```

(Spring 4.0.0.M1 and up)

Spring Boot: Starters

Pre-baked POM files

Add a dependency in Maven or Gradle

Makes dependencies available

Spring Boot: Actuator

Reasonable error pages

Endpoints for runtime metrics and info

`/beans` : All beans

`/env` : Configuration properties

`/trace` : Recent requests

`/dump` : Thread details

Example Time