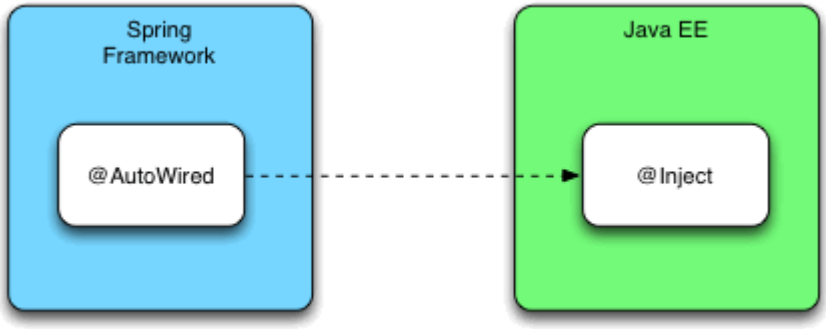
Acabamos de ver como usar Java equals y hashcode

La competencia entre los standares de Java EE y el framework Spring es cada vez más dura ya que las similitudes entre ambos son muchas. Elegir uno u otro depende de muchas cosas. En estos momentos existen muchos proyectos de Spring y algunos de ellos están valorando pasarse a Java EE en futuras evoluciones. Por ejemplo proyectos con JSF ya que la integración en Java EE es más sencilla al pertenecer al propio estandar. Si nos encontramos en situaciones como estas o similares. Podemos programar la aplicación de Spring para que use anotaciones que están dentro de los standares de de Java EE y nos facilite las migraciones.



Spring @Inject

Los servicios serán anotados con las siguientes anotaciones del standard y no con las de Spring:

@Named: Identifica una clase para que sea registrada y accesible via expression Language

@Inject: Permite injectar una dependencia dentro de una clase.

Vamos a verlo en código para ello el primer paso será añadir las dependencias que necesitemos a nuestro proyecto utilizando maven o algo similar.

```
 &amp;lt;dependency&amp;gt;
 &amp;lt;groupId&amp;gt;org.springframework&amp;lt;/groupId&amp;gt;
 &amp;lt;artifactId&amp;gt;spring-core&amp;lt;/artifactId&amp;gt;
 &amp;lt;version&amp;gt;4.1.0.RELEASE&amp;lt;/version&amp;gt;
 &amp;lt;/dependency&amp;gt;
 &amp;lt;dependency&amp;gt;
 &amp;lt;groupId&amp;gt;org.springframework&amp;lt;/groupId&amp;gt;
 &amp;lt;artifactId&amp;gt;spring-context&amp;lt;/artifactId&amp;gt;
 &amp;lt;version&amp;gt;4.1.0.RELEASE&amp;lt;/version&amp;gt;
 &amp;lt;/dependency&amp;gt;
 &amp;lt;dependency&amp;gt;
 &amp;lt;groupId&amp;gt;javax.inject&amp;lt;/groupId&amp;gt;
 &amp;lt;artifactId&amp;gt;javax.inject&amp;lt;/artifactId&amp;gt;
 &amp;lt;version&amp;gt;1&amp;lt;/version&amp;gt;
```

Una vez tenemos las dependencias generamos el fichero XML de Spring:

```
&amp;lt;beans xmlns="http://www.springframework.org/schema/beans"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:context="http://www.springframework.org/schema/context"
 xsi:schemaLocation="http://www.springframework.org/schema/beans
 http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
 http://www.springframework.org/schema/context
 http://www.springframework.org/schema/context/spring-context.xsd"&gt;

 &amp;lt;context:component-scan base-package="com.arquitecturajava"
/&amp;gt;

&amp;lt;/beans&amp;gt;
```

Como vemos unicamente realizamos un scan entre los paquetes de “com.arquitecturajava” hecho esto vamos mostrar el código fuente de las clases e interfaces:

```
package com.arquitecturajava.servicios;

public interface ServicioMensaje {

public String mensaje();
}
```

```
package com.arquitecturajava.servicios;

import javax.inject.Inject;
import javax.inject.Named;

@Named(value="servicioMensaje")
public class ServicioMensajeImpl implements ServicioMensaje {

 @Inject
 ServicioHijo componente;

 public String mensaje() {
 //
 return componente.getMensaje();
 }
```

```
}
```

```
package com.arquitecturajava.servicios;

import javax.inject.Named;

@Named
public class ServicioHijo {

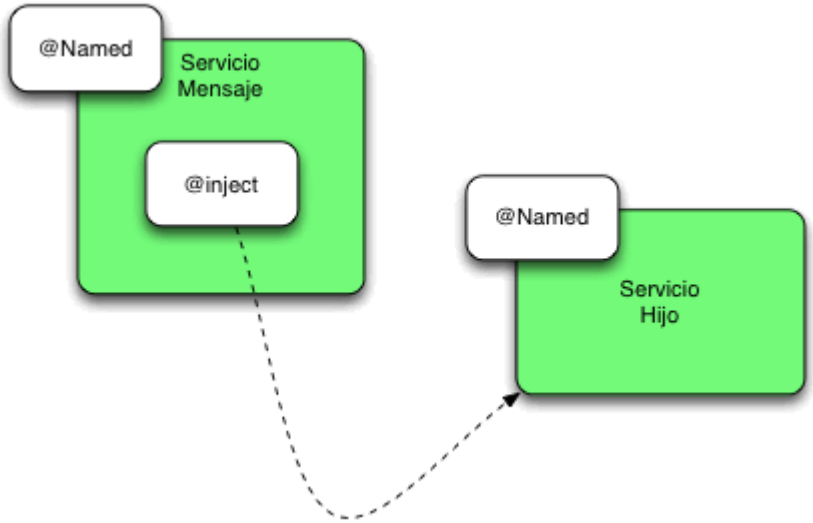
 private String mensaje;

 public String getMensaje() {
 return mensaje;
 }

 public void setMensaje(String mensaje) {
 this.mensaje = mensaje;
 }

 public ServicioHijo() {
 super();
 this.mensaje = "hola @inyect";
 }
}
```

De esta forma estamos anotando todas nuestras clases apoyándonos en las anotaciones del standard @Named y @Inject



Una vez hecho esto construimos el programa principal e invocamos al servicio:

```
package com.arquitecturajava.servicios;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationContext;

public class Principal {

public static void main(String[] args) {

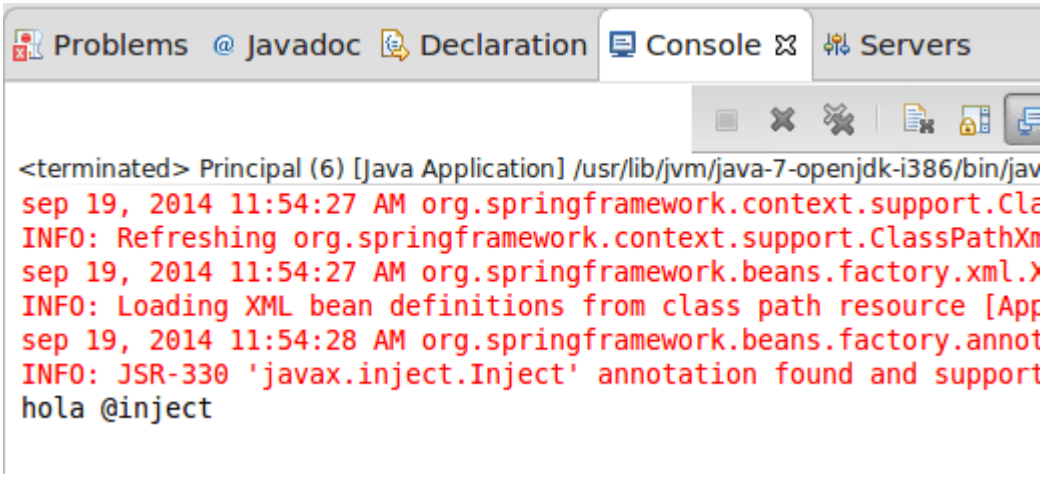
ApplicationContext contexto = new ClassPathXmlApplicationContext(
"ApplicationContext.xml");

ServicioMensaje servicio = (ServicioMensaje) contexto
.getBean("servicioMensaje");
System.out.println(servicio.mensaje());
```

```
}

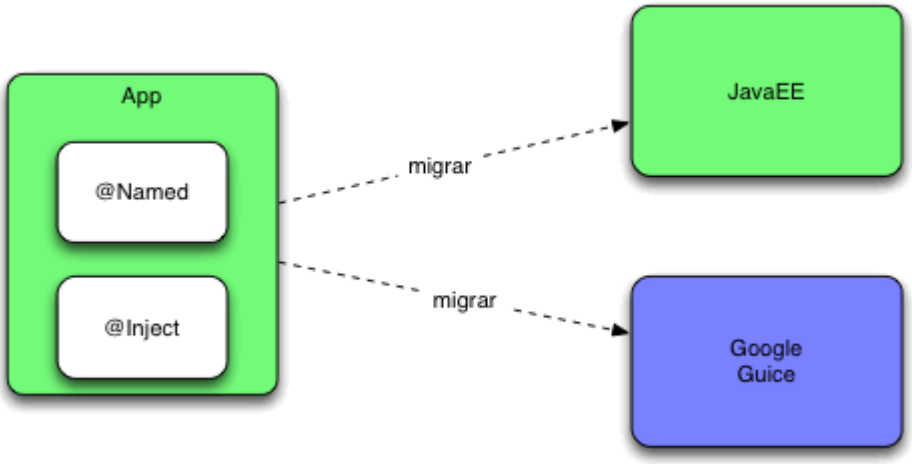
}
```

El programa se ejecutará sin problemas:



Anotaciones de Spring o JavaEE

Las anotaciones de Java EE nos permitiran migrar la aplicación de una forma más sencilla al mundo de los standars . Sin embargo no tienen las mismas capacidades que tienen las anotaciones de Spring a la hora de sacar partido al framework . Si para nosotros estas otras anotaciones nos son críticas nos seguiremos apoyando en las propias anotaciones de Spring framework . En el caso de no necesitarlas el uso de anotaciones standards nos puede simplificar las migraciones a otras plataformas.



Otros artículos relacionados:

JAX-RS

Spring Responsabilidades

Spring MVC