Savic

blished

l 14, 2016

t Posts

ud with
and Eu-
ailability
r swarm

remote
tutorial

ustering

on local
MacOS

Collec-
tors

in Spring

Spring for dummies

Introduction

Spring is awesome, it is easy to use and setup,but before we jump into big projects. Lets see few application contexts in action.

Possible choices are:

1. ClassPathXmlApplicationContext, it allows us to load application context from file located in the class path.
2. FileSystemXmlApplicationContext, it allows us to load application context from file using absolute path.
3. AnnotationConfigApplicationContext,it allows us to load application context from class that it annotated with @Configuration.

For each example we are going to use same POJO named MajesticKnight that implements Knight interface.

```
public interface Knight {
    String getKnightName();
    void setKnightName(String name);
}
```

```
public class MajesticKnight implements Knight {
 private String knightName = "Majestic Knight";

 public void setKnightName(String knightName) {
   this.knightName = knightName;
 }

 public String getKnightName() {
   return knightName;
 }

}
```

XML configuration file

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="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.xsd">

    <bean id="knight" class="rs.pscode.model.MajesticKnight"></bean>
</beans>
```

Configuration class equivalent to above xml configuration

```
@Configuration
public class KnightConfig {

    @Bean
    public Knight getKnight() {
        return new MajesticKnight();
    }
}
```

Examples

In the next 3 examples I will show you how to create contexts , obtain knight bean from them and print the knights name.

ClassPathXmlApplicationContext

```
 ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("META-INF/spring/
 Knight knight = context.getBean(MajesticKnight.class);

 System.out.println(knight.getKnightName());
 context.close();
```

output will be : *Majestic Knight*

FileSystemXmlApplicationContext

Important note is that here we must add “**file:**” before the absolute path, here is the part of spring documentation that relates to this note.

Plain paths will always be interpreted as relative to the current VM working directory, even if they start with a slash. (This is consistent with the semantics in a Servlet container.) Use an explicit “file:” prefix to enforce an absolute file path.

```
String path  ="/opt/.../META-INF/spring/knight.xml";
FileSystemXmlApplicationContext context = new FileSystemXmlApplicationContext("<strong>file:</

Knight knight = context.getBean(MajesticKnight.class);
System.out.println(knight.getKnightName());
context.close();
```

AnnotationConfigApplicationContext

Observe that KnightConfig class is used as parameter to AnnotationConfigApplicationContext.

```
AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(KnightConf
Knight knight = context.getBean(MajesticKnight.class);

System.out.println(knight.getKnightName());
context.close();
```

Unit tests

We have seen how to create application context and how to use it, now we will see how to create a unit test and verify that all is correct and that name of the Knight instance is correct. First lets see how to use KnightConfig configuration class.

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(classes = KnightConfig.class)
public class KnightTest {

    @Autowired
    Knight knight;

    @Test
    public void test() {
        assertEquals("Majestic Knight", knight.getKnightName());
    }

}
```

We can configure context configuration from xml as well.

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration({"file:src/main/resources/META-INF/spring/knight.xml"})
public class KnightTest {

    @Autowired
    Knight knight;

    @Test
    public void test() {
        assertEquals("Majestic Knight", knight.getKnightName());
    }

}
```

As you have seen using spring is just awesome, easy and straightforward.

Write a Comment

WEBMENTIONS

Spring Bean Lifecycle – pscode.rs

[…] previous blog post I have shown you how to use ClassPathApplicationContext and run your examples. Lets repeat this […]