Using javascript, can you open a new window? window.open

Using javascript, can you access every element present in a html file? Xpath

What is Web socket?

|  |
| --- |
| Introduction to UI Engineering HTML 5 Basics (Overview, Syntax, Attributes, Events, Web Forms, Web Storage, Web Socket, Canvas, Geolocation, Web Workers, CORS etc.,) |
| Introduction to CSS 3 CSS3 Basics, CSS 3 Advacned. Designing paper layout case study |
| Introduction Bootstrap 3, Forms, Utilities, Layout Components, Plugins |
| Case study of building responsive pages using HTML 5, CSS 3, Bootstrap 3 |
| Introduction to Javascript Syntax, Variables, Operators, Conditional Statements, Functions |
| Arrays (traverse,map,reduce, foreach), chaining of functions, String functions, Accessing Collections,Classes and objects Introduction to DOM, using getElementsBy tag,class,id ,innerText/HTML Case study to implement DOM functionalities |

Enterprise Java Beans EJB

Working in EJB was like winter. Difficult to live.

So a new f/w was introduced to simplify the life of programmers: Spring

Java Beans

What is a bean? Any re-usable piece of code in java (component)

Any java code independent of any type of application (console/web/mobile)

Is a java bean

interface EmployeDao

{

Employee addEmployee(Employee employee);

}

class EmployeeDaoImpl implements EmployeeDao

{

List<Employee> empList=new ArrayList<>();

public Employee addEmployee(Employee employee)

{

//code to add an employee to the existing list of emps  
 return null;

}

}

EJB = Java Bean + services

What are those services?

Scalability – ability to adapt to growing or reducing needs

Security

Transaction

Life cycle management

Instance pool minPoolSize maxPoolSize

When we use spring framework / ejb, we do not create an object of a class (java bean)

These services were taken care by programmers, before EJB.

Once, EJB was introduced, programmers focus only on the business logic (code inside the methods)

Who takes care of these service????

These services are called as Generic service and are taken care by

Container.

Container = special JVM

In EJB = EJB Container

In spring = Spring IOC container

IOC = Inversion Of Control

Dependency Injection

class Car

{

Engine engine;

Audio audio;

}

Interface Audio

{

}

class Jbl implements Audio{}

class Sony implements Audio{}

class Car

{

private Engine engine;

private Audio audio;

public Car(){}

public Car(Audio audio)

{

this.audio=audio;

}

public void setAudio(Audio audio)

{

this.audio=audio;

}

}

Car bmw=new Car();

bmw.audio=new Sony(); //NOT POSSIBLE because, audio is private

bmw.setAudio(new Sony()); //POSSIBLE using setter

Car benz=new Car(new Jbl()); //POSSIBLE using constructor

Demo1:

Lets create a spring project.

It is a maven project.

In pom.xml if we add the following dependencies, then it is becoming a spring project

Spring framework has lot of modules:

Spring core

Spring AOP

Spring MVC

Spring DAO

Spring REST

Spring Security

We are learning spring core now.

1. In eclipse, create a maven project

Group id: com.ust.spring

Artifcat id: 29-may-01

Package is same as group id

pom.xml

Project Object Model

1. In pom.xml, under “dependencies”

**<dependency>**

**<groupId>org.springframework</groupId>**

**<artifactId>spring-context</artifactId>**

**<version>4.3.6.RELEASE</version>**

**</dependency>**

1. Lets create a java bean

package com.ust.spring;

public class Jbl implements Audio

{

@Override

public void playMusic() {

System.out.println("Jbl Makes noise");

}

}

package com.ust.spring;

public class Sony implements Audio {

@Override

public void playMusic() {

System.out.println("Sony makes good sound");

}

}

package com.ust.spring;

public class Car {

private Audio audio;

public Car() {}

public Car(Audio audio) {

super();

this.audio = audio;

}

public Audio getAudio() {

return audio;

}

public void setAudio(Audio audio) {

this.audio = audio;

}

public void play()

{

audio.playMusic();

}

}

<?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-3.0.xsd">

<bean id="maths" class="com.ust.spring.Mathematics">

<property name="no1" value="12" />

<property name="no2" value="13" />

</bean>

<bean id="jbl" class="com.ust.spring.Jbl">

</bean>

<bean id="sony" class="com.ust.spring.Sony">

</bean>

<bean id="car1" class="com.ust.spring.Car">

<property name="audio" ref="sony" />

</bean>

</beans>

package com.ust.spring;

import org.springframework.context.support.ClassPathXmlApplicationContext;

/\*\*

\* Hello world!

\*

\*/

public class App

{

public static void main( String[] args )

{

System.out.println( "Hello World!" );

//how to get an instance of bean???????

ClassPathXmlApplicationContext ctx=new ClassPathXmlApplicationContext("bean.xml");

// Mathematics maths= (Mathematics) ctx.getBean("maths");

// maths.setNo1(20);

// maths.setNo2(30);

// System.out.println(maths.sum());

// ctx.getBean("car1");

Car car1 = ctx.getBean(Car.class);

car1.play();

Jbl jbl=(Jbl) ctx.getBean("jbl");

car1.setAudio(jbl);

car1.play();

}

}

<property name=*"audio"* ref=*"sony"* />

Task:

---------

Create the following classes and define them as beans in xml file

Book

Author

Book has Author

1. Create a maven project (quickstart archetype / skip archetype)
2. In pom.xml, add spring-context dependency

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.3.6.RELEASE</version>

</dependency>

1. Create Book class and Author class
2. Create spring.xml and configure these 2 classes as beans
3. App.java main method,

Use ClassPathXmlApplicationContext to represent spring.xml as an object

Using that object, getBean of type Book

Display the author’s name from book instance

class Book

{

List<Author> authorList=new ArrayList<>();

}

By default, bean is singleton.

config.xml

--------

<?xml version="1.0" encoding="UTF-8"?>

<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"

xmlns:p="http://www.springframework.org/schema/p"

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">

</beans>

----------------------------------

Singleton (1) even if we dont get Bean

Prototype

<bean name="" class="">

<property name="">

<value></value>

</property>

</bean>

---------------------

<bean name="" class="">

<property name="" value="" />

</bean>

-----------------------

<bean name="" class="" p:name="value" />

-----------------------------

<bean name="" class="">

<property name="">

<list>

<value></value>

<value></value>

</list>

</property>

</bean>

--------------------------------------------

<bean name="" class="">

<property name="">

<list>

<ref bean="" />

<ref bean="" />

</list>

</property>

</bean>

------------------------------------

<bean name="" class="">

<property name="">

<set>

<ref bean="" />

<ref bean="" />

</set>

</property>

</bean>

--------------------------

<bean name="" class="">

<property name="">

<map>

<entry key="" value="" />

<entry key="">

<value></value>

</entry>

</map>

</property>

</bean>

Instead of using xml configuration file,

We can configure the beans, using “Annotation”s

To configure a bean, we can use:

@Bean

@Component

If you want to declare a class as a bean,

@Component

@Component

public class Book

{}

Suppose, if there is a built-in class JdbcTemplate

We are not the owner of the class: we cannot modify the source code of that class

@Component

public class JdbcTemplate //NOT POSSIBLE

@Bean

public JdbcTemplate jt()

{}

--------------------------

Spring core without xml configuration file / config class:

package com.ust.spring;

import org.springframework.stereotype.Component;

@Component("emp")

public class Employee {

private Integer id;

private String name;

private String department;

public Employee() {}

public Employee(Integer id, String name, String department) {

super();

this.id = id;

this.name = name;

this.department = department;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDepartment() {

return department;

}

public void setDepartment(String department) {

this.department = department;

}

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", department=" + department + "]";

}

}

package com.ust.spring;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

/\*\*

\* Hello world!

\*

\*/

public class App

{

public static void main( String[] args )

{

System.out.println( "Hello World!" );

// ClassPathXmlApplicationContext ctx=new ClassPathXmlApplicationContext("spring.xml");

AnnotationConfigApplicationContext ctx=new AnnotationConfigApplicationContext();

ctx.scan("com");

ctx.refresh();

// ctx.getBean(Employee.class);

Employee emp=(Employee) ctx.getBean("emp");

emp.setId(20);

emp.setName("Jag");

emp.setDepartment("Training");

System.out.println(emp);

}

}

Task:

Create a bean Department

id

name

location

create a config class

in app main method, get bean of department.

Implement this using

1. @Component
2. @Bean