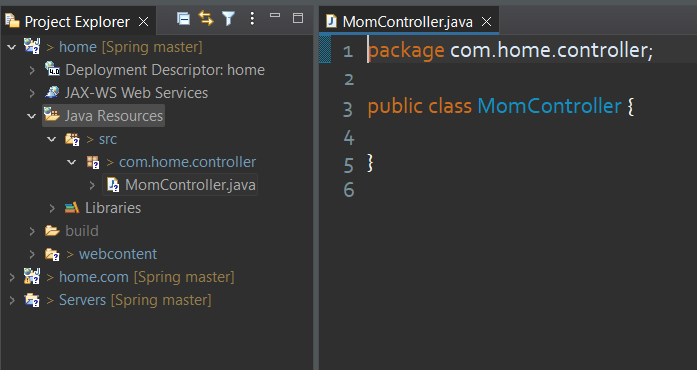
Controllers are actually the java classes. So if we want to define controller for mother, we can just create a class with the name **MomController.**

Let’s create the mother controller,

We’ll first create a package inside our **src** folder with the name **com.home.controllers**

Then we’ll create a java class **MomController** as a part of this package.



**@Controller Annotation**

The @Controller Annotation as similar as @Component Annotation.

Firstly, we have to make an entry of <**context:component-scan>** tag inside our **frontcontroller-servlet.xml** file by which whenever application load the spring automatically create object of the all the classes which is annotating with @Controller annotation.

**frontcontroller-servlet.xml :**

<**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:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/context*

*https://www.springframework.org/schema/context/spring-context.xsd "*>

<**context:component-scan** base-package=*"com.home.controllers"* />

</**beans**>

To make our **MomController** class a actual controller we have to use an annotation @Controller with it like below :

*@Controller*

public class MomController {

}

Now let’s make our MomController to handle the request url “**/sugar**”

*@Controller*

public class MomController {

public MomController() {

System.***out***.println("Contstructor called");

}

*@ResponseBody*

*@RequestMapping*("/sugar")

public String giveSugar() {

return "Here is your sugar";

}

}

You have noticed that we have used two new annotation. So here is description of both of two :

1. **@ResponseBody**

This annotation is responsible to whatever string is returning by the method should be put inside the response body. So if the request arrived for **/sugar** then the string which is returning the **giveSugar()** method will be shown on the webpage as a response.

1. **@RequestMapping(“/sugar”)**

As the name suggest, this annotation will be use to mention the request url for method. So whenever the “/sugar” url will be come together in the request url so the **giveSugar()** method will be invoke.

It’s time to run our application again but before do that let’s have look our **web.xml** file for properly write the url.

<**servlet**>

<**servlet-name**>frontcontroller</**servlet-name**>

<**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>

<**load-on-startup**>1</**load-on-startup**>

</**servlet**>

<**servlet-mapping**>

<**servlet-name**>frontcontroller</**servlet-name**>

<**url-pattern**>/myhome/\*</**url-pattern**>

</**servlet-mapping**>

* **Project Name :** home
* **Front Controller Mapping :** /myhome/\*
* **Controller Mapping :** /sugar

So the complete url will be :

<http://localhost:2020/home/myhome/sugar>

So hopefully your webpage should display the message sent by the MomController as response.

Your exercise is to create **SonController** and **DaughterController.**