

Flash Back to XML config (the old way)

File: web.xml

`<web-app>`

`</web-app>`

Just an FYI

Web App_INITIALIZER

- Spring MVC provides support for web app initialization
- Makes sure your code is automatically detected
- Your code is used to initialize the servlet container

AbstractAnnotationConfigDispatcherServletInitializer

Web App_INITIALIZER (more info)

AbstractAnnotationConfigDispatcherServletInitializer

- Your TO DO list
 - Extend this abstract base class
 - Override required methods
 - Specify servlet mapping and location of your app config

Step 3: Create Spring Dispatcher ServletInitializer

File: MySpringMvcDispatcherServletInitializer.java

```
import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;  
  
public class MySpringMvcDispatcherServletInitializer extends AbstractAnnotationConfigDispatcherServletInitializer {
```

```
<servlet>  
  <servlet-name>dispatcher</servlet-name>  
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
  
  <init-param>  
    <param-name>contextConfigLocation</param-name>  
    <param-value>/WEB-INF/spring-mvc-demo-servlet.xml</param-value>  
  </init-param>  
  
  <load-on-startup>1</load-on-startup>
```

Our config class
from Step 2

Whew!!!!

Now for the easy stuff

Create Spring MVC Controller and view page

easy peazy ...

Step 4: Develop our Spring Controller

File: DemoController.java

```
@Controller  
public class DemoController {  
  
    ,  
  
}
```



View name

/WEB-INF/view/home.jsp

Step 5: Develop our JSP view page

File: /WEB-INF/view/home.jsp

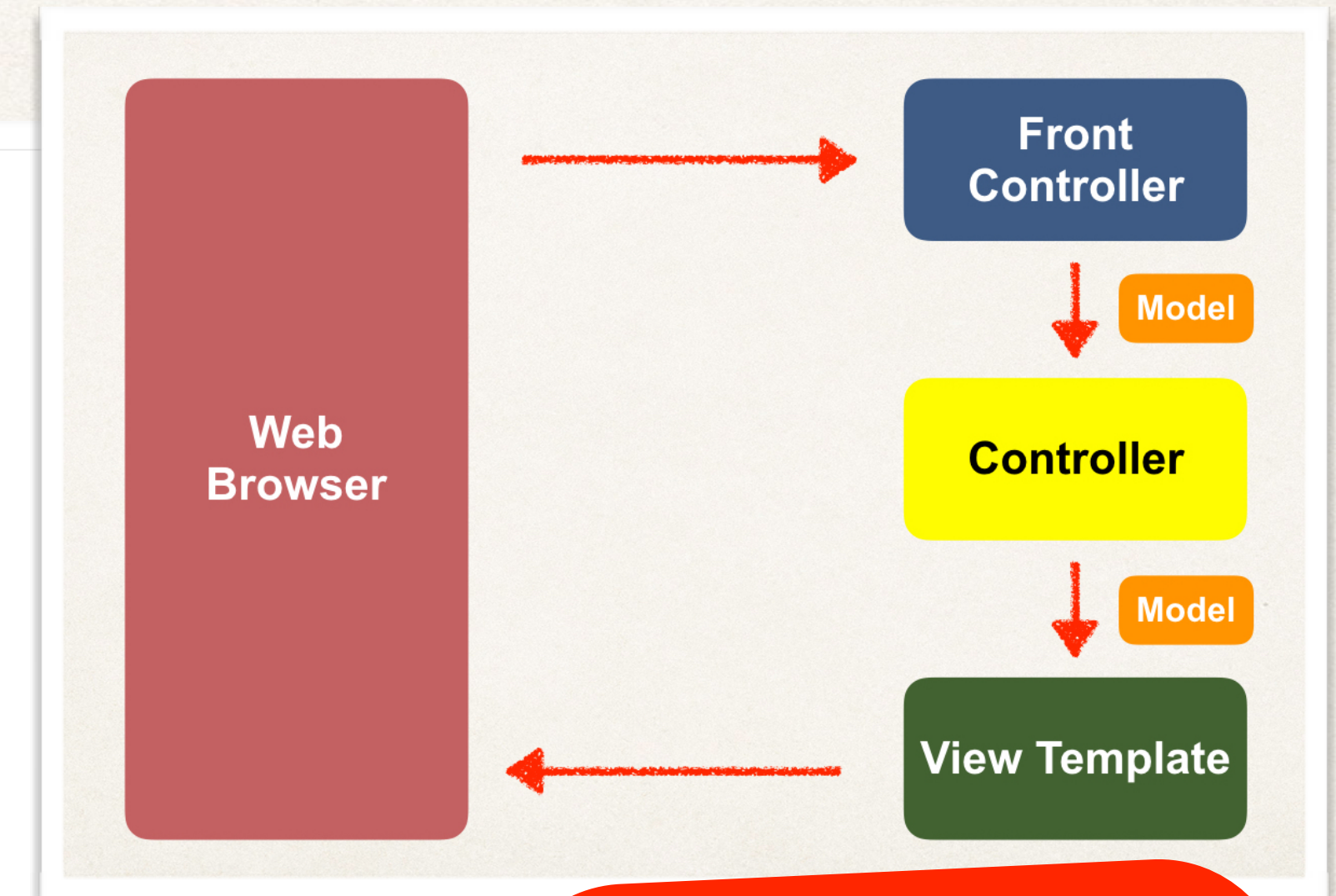
```
<html>
```

```
<body>
```

Welcome to the luv2code company home page!

```
</body>
```

```
</html>
```



No XML!