

102045605

Advanced Java

Unit-7

Spring MVC

Subject Overview

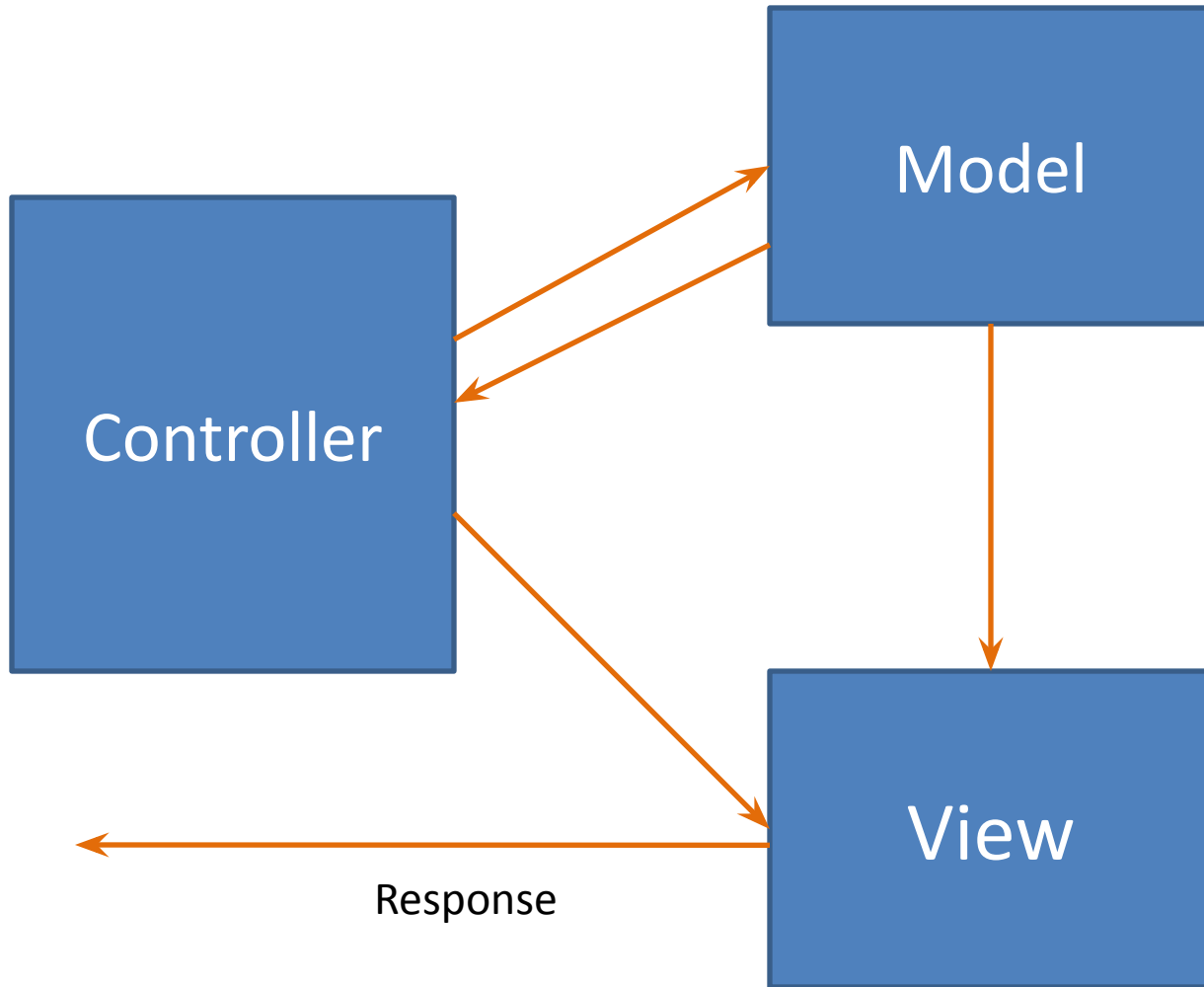
Sr. No.	Unit	% Weightage
1	Java Networking	5
2	JDBC Programming	10
3	Servlet API and Overview	25
4	Java Server Pages	25
5	Java Server Faces	10
6	Hibernate	15
7	Java Web Frameworks: Spring MVC	10

Reference Book:

Black Book “ Java server programming” J2EE, 1st ed., Dream Tech Publishers, 2008. 3. Kathy walrath ”
Chapter 21

What is MVC?

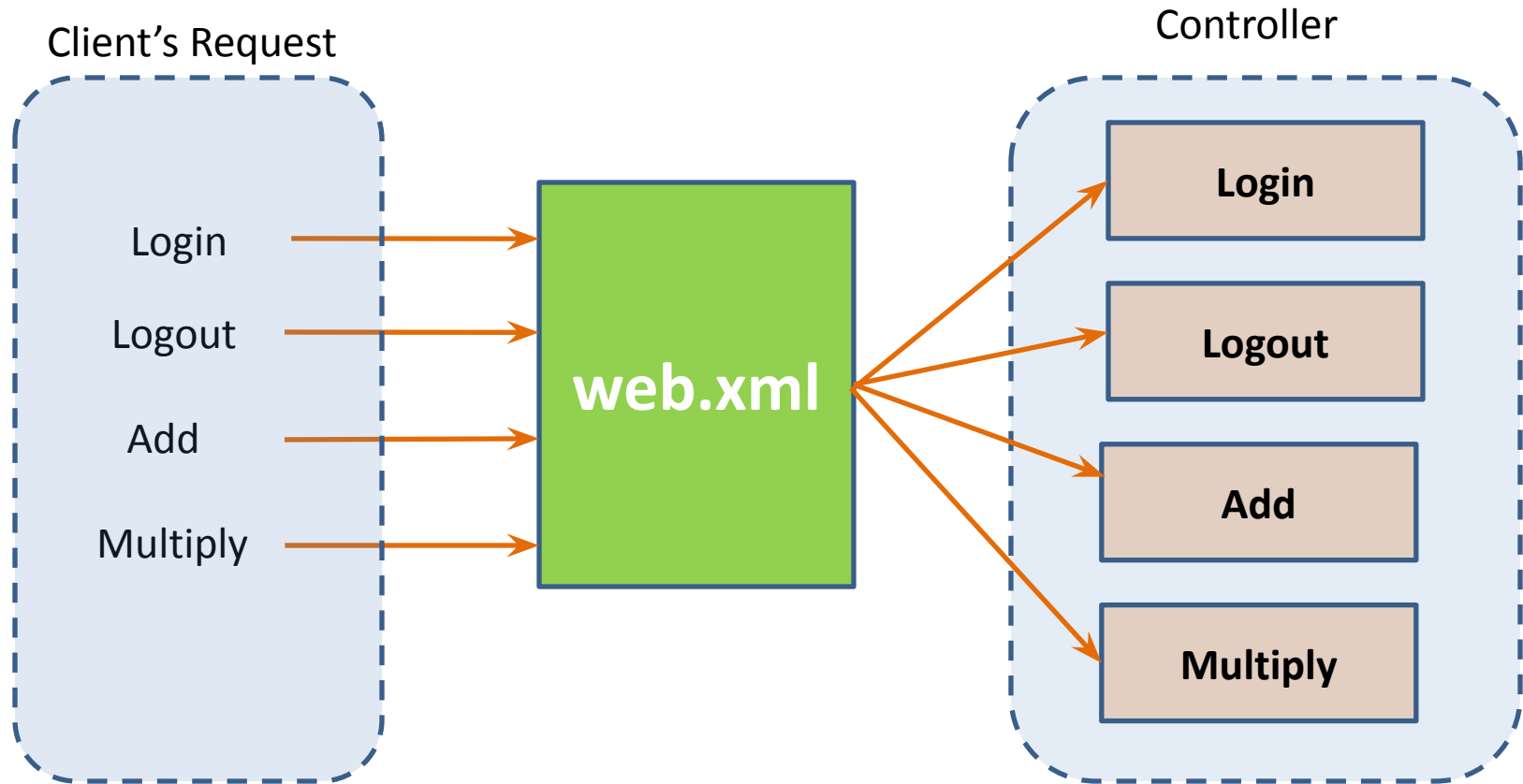
MVC – An overview



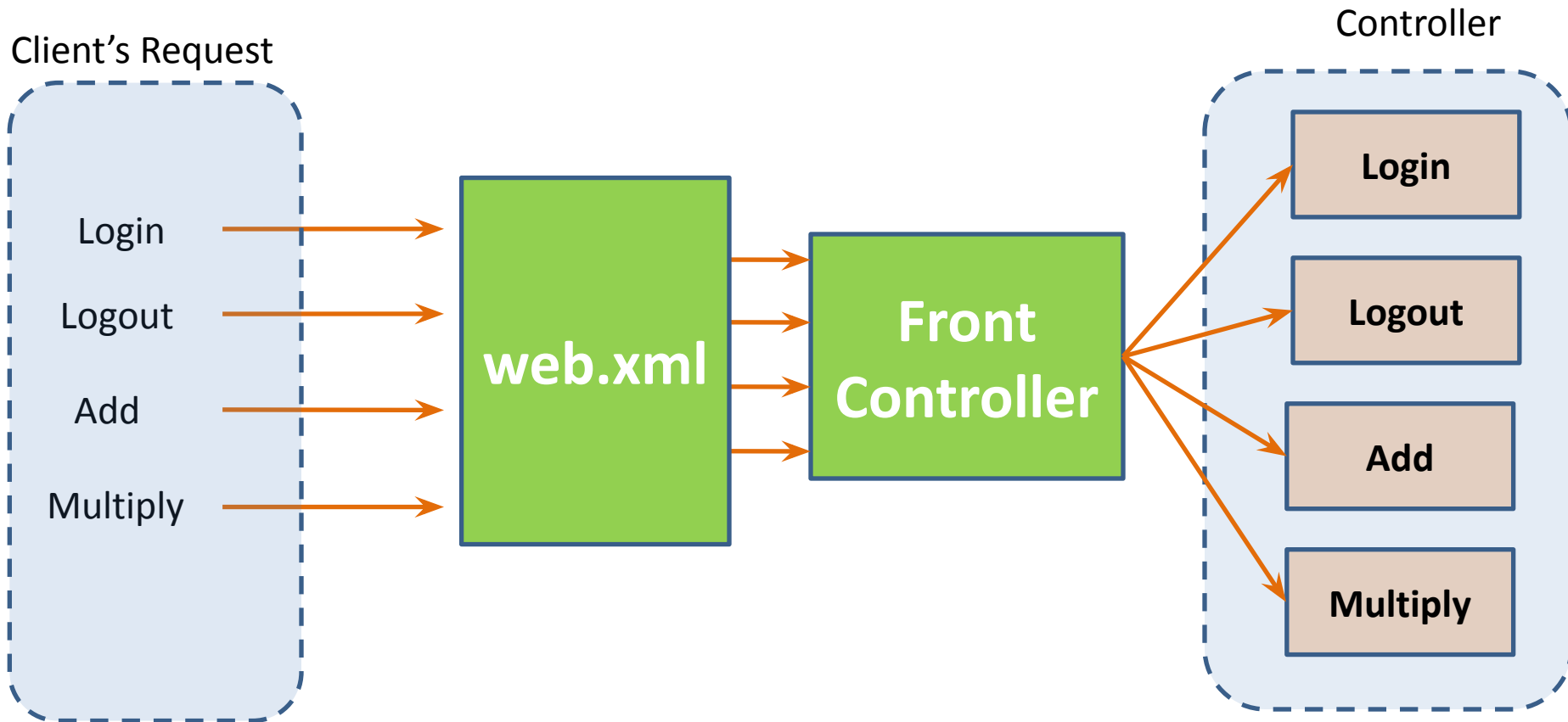
Advantage of MVC

- Navigation control is centralized. Now only controller contains the logic to determine the next page.
- Easy to maintain
- Easy to extend
- Easy to test
- Better separation of concerns
- Reusability

MVC Architecture



Spring MVC Architecture



Front Controller - Responsibilities

- **Initialize** the framework to supply to the requests.
- **Load** the map of all the URLs and the components responsible to handle the request.
- Prepare the map for the **views**.

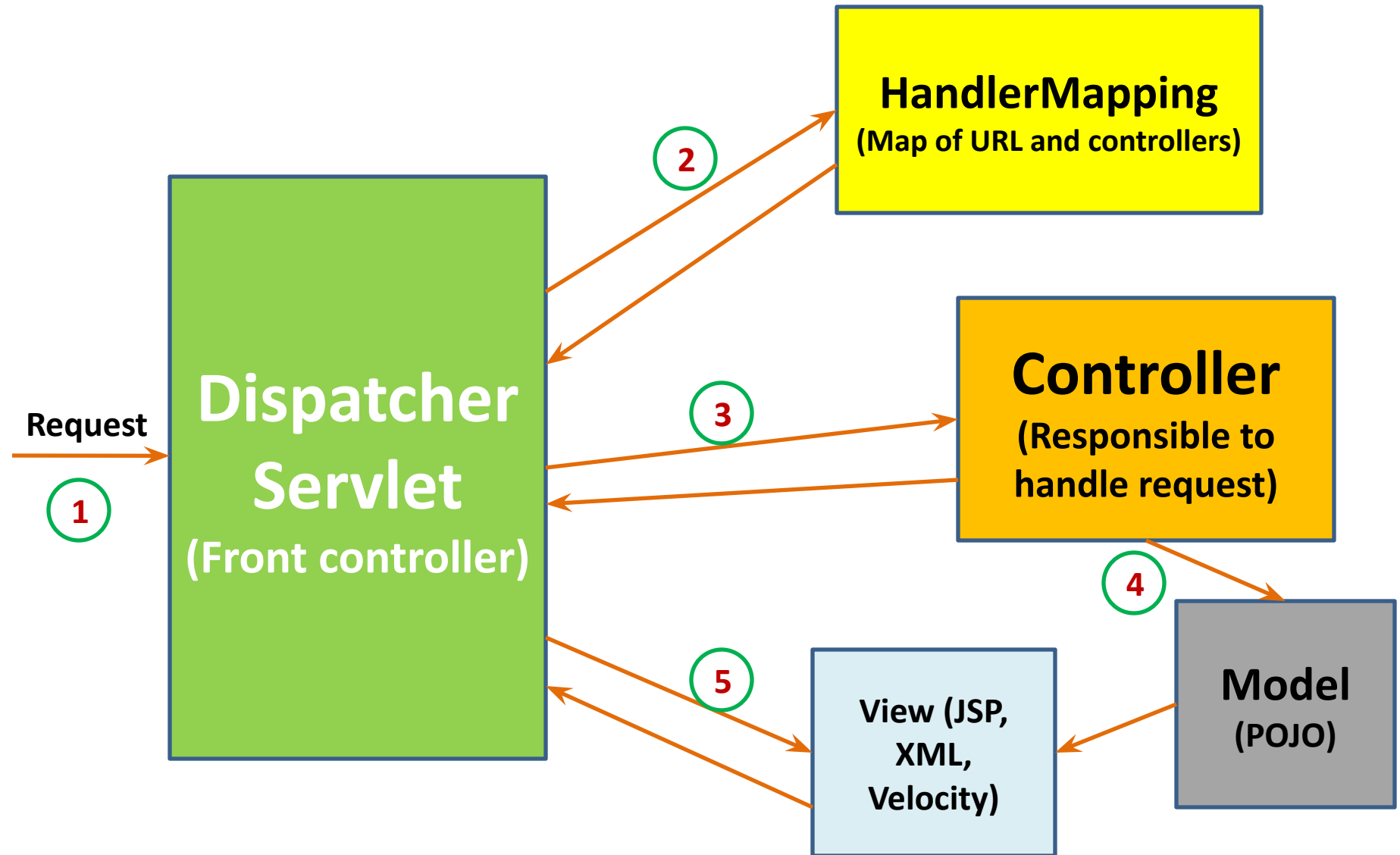
What is Spring MVC Framework?

- Spring **links objects** together instead of the objects linking themselves together.
- Spring object linking is defined in **XML** files, allowing easy changes for different application configurations thus working as a plug in architecture.

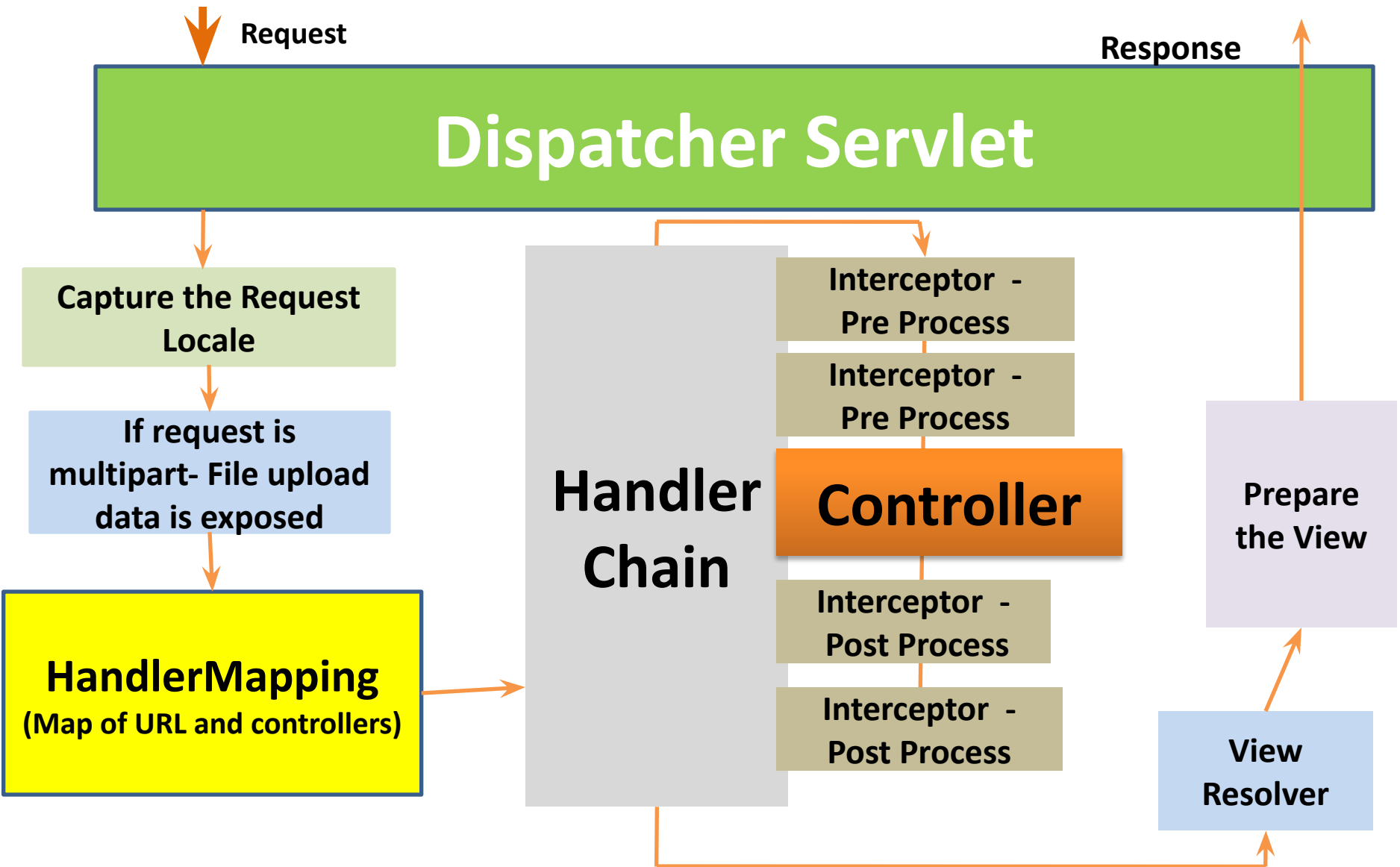
What is Spring MVC Framework?

- In an MVC architecture the controllers handle all requests.
- Spring uses a “DispatcherServlet” defined in the web.xml file to analyze a request URL pattern and then pass control to the correct controller by using a URL mapping defined in a “Spring bean” XML file.

Spring 3 MVC- Basic Architecture



Spring 3.0 MVC Request Flow



Why Spring Framework?

- All **frameworks integrate** well with Spring.
- Consistent Configuration, open **plug-in architecture**.
- Integrates well with different O/R Mapping frameworks like **Hibernate**.
- Easier to **test** applications with.
- Less complicated than other frameworks.
- Active user community.
- Spring is well organized and seems easier to learn comparatively
- Spring also supports **JDBC Framework** that makes it easier to create JDBC Apps.

Features of Spring MVC Framework

1. Inversion of Control (IoC) Container

It is used to provide object reference to class during runtime.

2. Data Access Framework

3. Transaction Management

It enables developers to easily write code to access the persistent data throughout the application.

4. Spring Web Services

It enables developers to model a wide range of transaction by providing Java Transaction API (JTA).


It provides powerful mapping for transmitting incoming XML request to any object.

Advantage of Spring MVC Framework

- Predefined Templates
- Loose Coupling
- Easy to test
- Lightweight
- Fast Development
- Declarative Support
- Hibernate and JDBC Support
- MVC Architecture and JavaBean Support

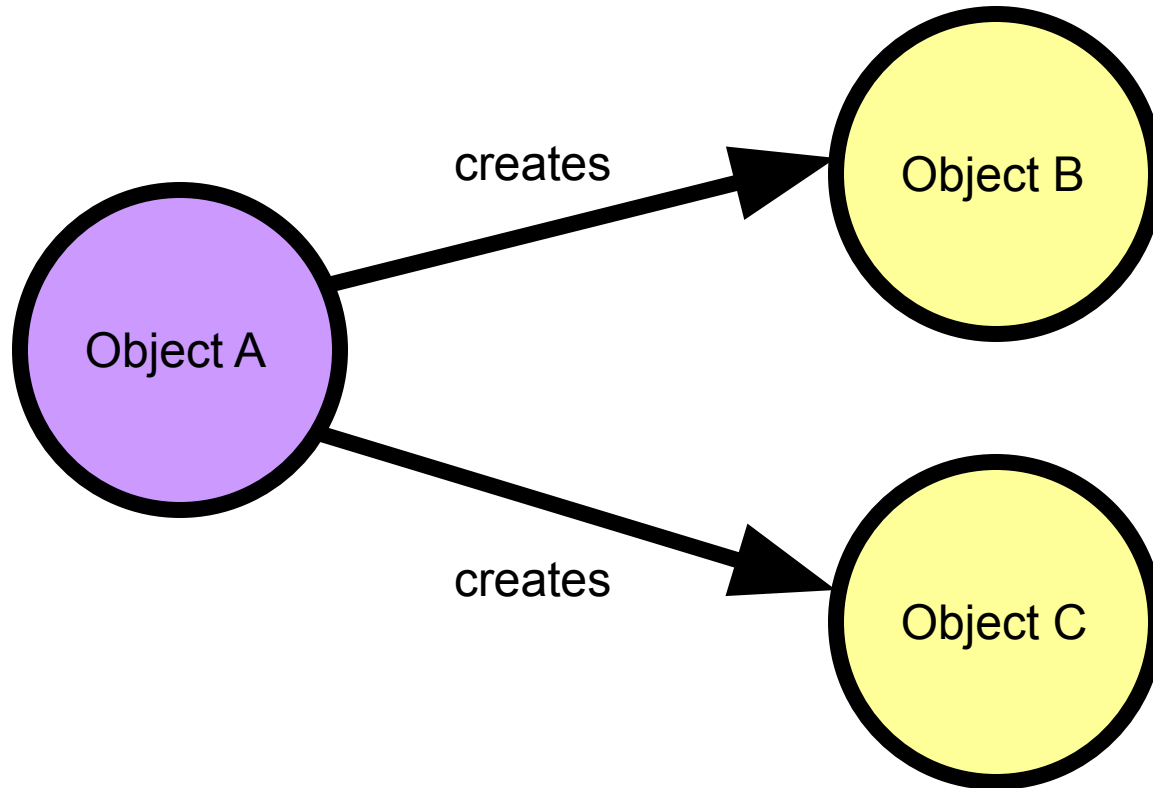
Example of web.xml file

```
<web-app>
<servlet>
    <servlet-name>tradingapp</servlet-name>
    <servlet-class>DispatcherServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>tradingapp</servlet-name>
    <url-pattern>*.html</url-pattern>
</servlet-mapping>
</web-app>
```



*** Any URL ending with an “.html” pattern is routed to the DispatcherServlet, the DispatcherServlet loads the **tradingapp-servlet.xml** file and routes the user to the correct controller.

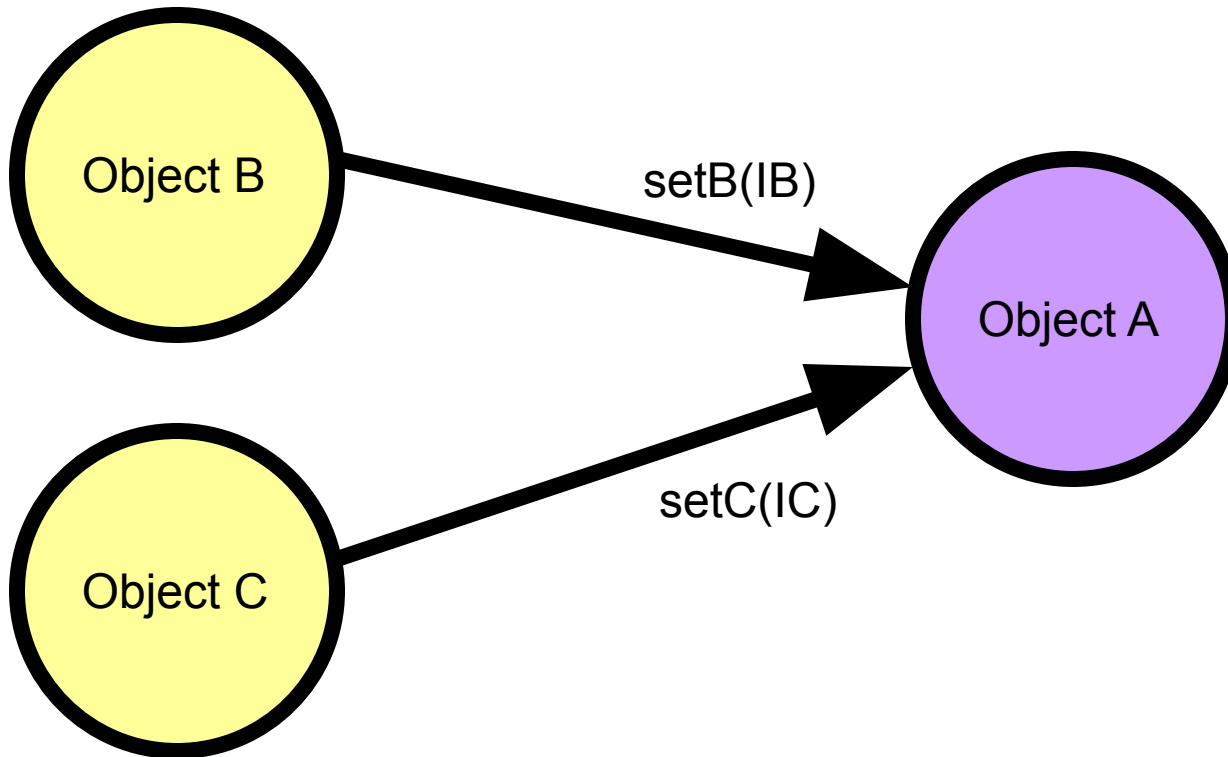
Without Dependency-Injection/loC



An object creating its dependencies without IoC leads to tight object coupling.

With Dependency-Injection/loC

- Allows objects to be created at higher levels and passed into object so they can use the implementation directly



Object A contains setter methods that accept interfaces to objects B and C. This could have also been achieved with constructors in object A that accepts objects B and C.

Advantage of Spring MVC over MVC

- Spring is a powerful Java application framework, used in a wide range of Java applications.
- Spring provides a very clean division between controllers, JavaBean models, and views.
- Spring's MVC is very flexible.
- Spring MVC is entirely based on interfaces.
- Every part of the Spring MVC framework is configurable via plugging in your own interface.
- No Action Forms, bind directly to domain objects.
- More testable code (validation has no dependency on Servlet API).
- Spring offers better integration with view technologies other than JSP (Velocity / XSLT / FreeMarker / XL etc.)

Important Intefaces

Interface	Default bean name	purpose
org.springframework.web.servlet. HandlerMapping	handlerMapping	Maps the Request to Handlers(Controllers)
org.springframework.web.servlet. HandlerAdapter	none	Plugs the other frameworks handlers
org.springframework.web.servlet. ViewResolver	viewResolver	Maps the view names to view instances
org.springframework.web.servlet. HandlerExceptionResolver	handlerExceptionResolv er	Mapping of the exceptions to handlers and views
org.springframework.web. multipart.MultipartResolver	multipartResolver	Interface to handle the file uploads
org.springframework.web.servlet. LocaleResolver	localeResolver	Helps to resolve the locale from the request
org.springframework.web.servlet. ThemeResolver	themeResolver	Resolves a theme for a Request.

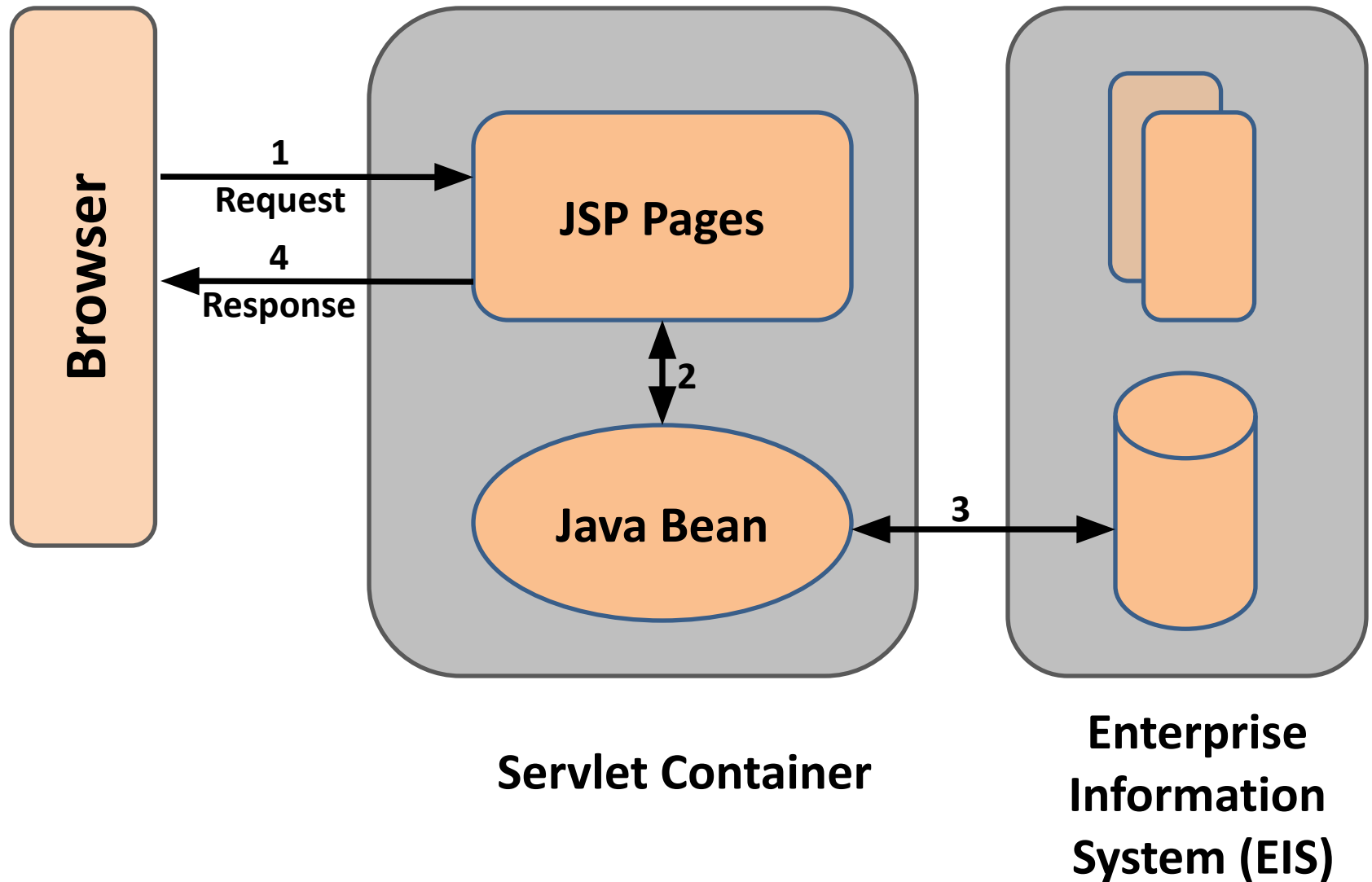
Java Bean vs Basic Java Class

- A Bean is a Java class, but a Java class does not have to be a bean.
- A Java Bean is a java class that should follow following conventions:
 1. It should have a **no-arg constructor**.
 2. It should be **Serializable**.
 3. It should provide methods to set and get the values of the properties, known as **getter** and **setter** methods.

Why use Java Bean?

- It is a **reusable** software component.
- A bean **encapsulates** many objects into one object, so we can access this object from multiple places.
- Moreover, it provides the easy **maintenance**.

Java Bean Architecture



Bean Life Cycle

- The life cycle of a Spring bean is easy to understand.
 1. When a bean is instantiated, it may be required to perform some initialization to get it into a usable state.
 2. Similarly, when the bean is no longer required and is removed from the container, some cleanup may be required.

Bean Life Cycle

```
1.  public class HelloWorld {  
2.      private String message;  
3.      public void init() {  
4.          System.out.println("Bean is going through init.");  
5.      }  
6.      public void setMessage(String message) {  
7.          this.message = message;  
8.      }  
9.      public void getMessage() {  
10.         System.out.println("Your Message : " + message);  
11.     }  
12.     public void destroy() {  
13.         System.out.println("Bean will destroy now.");  
14.     }  
15. }
```

GTU Questions

1.	Explain MVC Architecture
2.	What are the differences between Java Bean and basic java class? Explain Java Bean Architecture.
3.	Differentiate : Java Bean and basic java class and Explain Java Bean Architecture and Show the use of JSP inbuilt objects: request and response, with their use in application
4.	What is Spring Web MVC framework? List its key features.
5.	Explain MVC architecture in detail with figure.
6.	Write a java bean named “student” having roll no and name having getter & setter methods. Write a JSP page to set the roll number and name for a student object and then print them by reading from object.
7.	What is MVC architecture? Explain Spring architecture with neat sketch.
8.	What is Dependency Injection?
9.	Briefly explain spring bean life cycle.