Welcome

Spring Framework

ph no: 9513216462 | email : info@emexotechnologies.com | website: https://www.emexotechnologies.com

Spring MVC

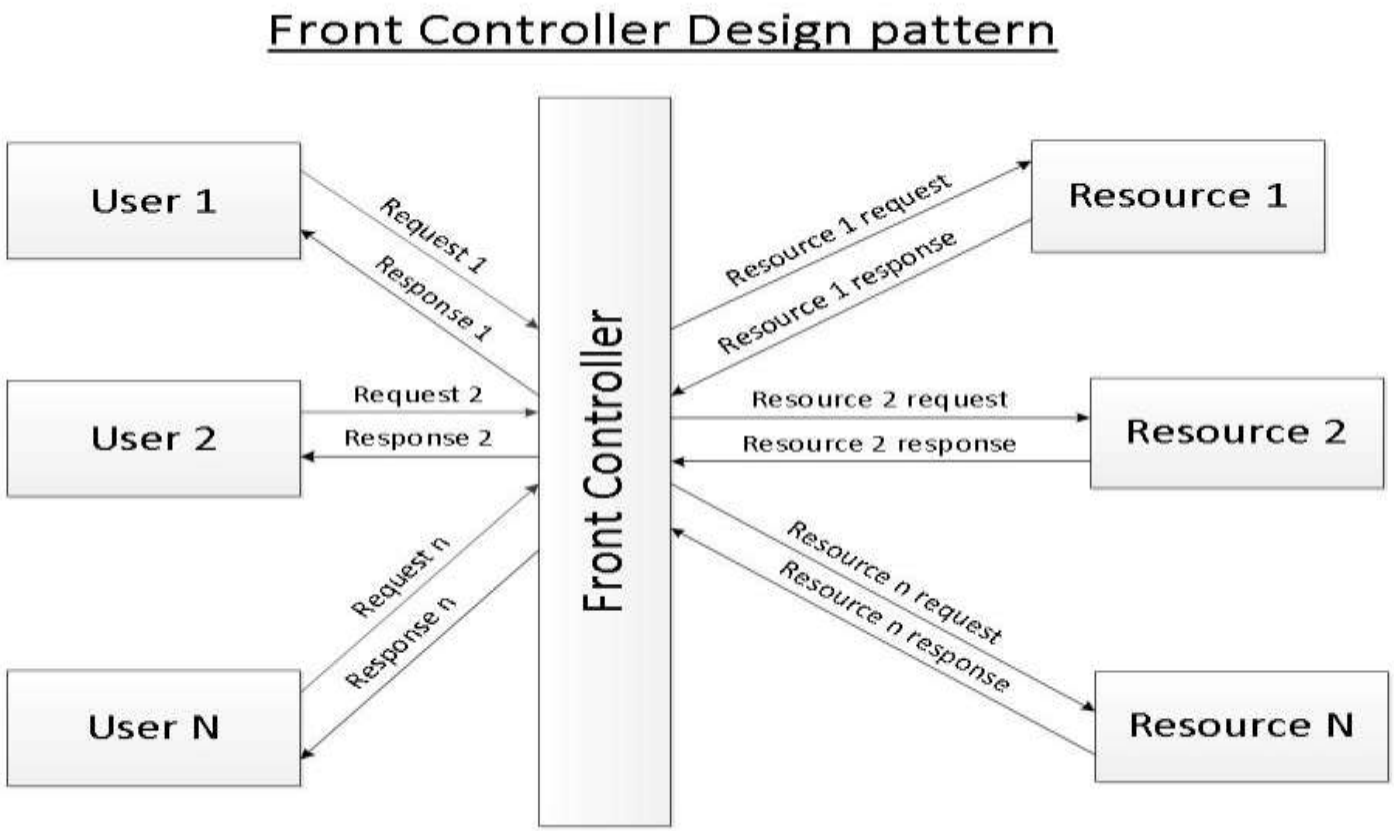
Introduction

- Spring framework makes the development of web applications very easy by providing the Spring MVC module.
- Spring MVC module is based on two most popular design patterns - Front controller and MVC.

Front Controller design pattern

- This design pattern enforces a single point of entry for all the incoming requests.
- All the requests are handled by a single piece of code which can then further delegate the responsibility of processing the request to further application objects.

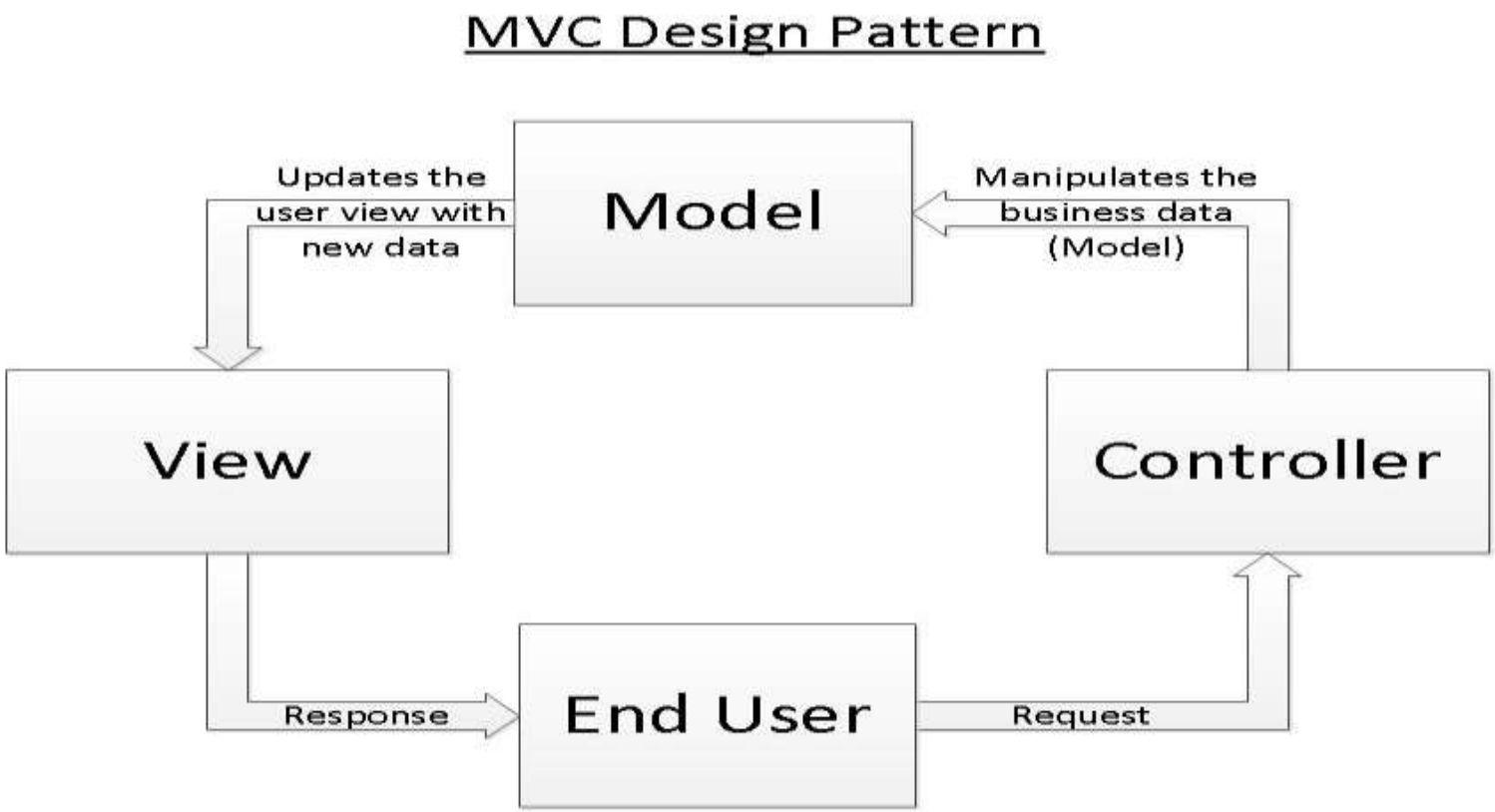
Front Controller design pattern (Cond..)



MVC Design Pattern

- This design pattern helps us develop loosely coupled application by segregating various concerns into different layers. MVC design pattern enforces the application to be divided into three layers, Model, View and Controller.
- Model: This represents the application data.
- View: This represents the application's user interface. View takes model as the input and renders it appropriately to the end user.
- Controller: The controller is responsible for handling the request and generating the model and selecting the appropriate view for the request.

MVC Design Pattern (Contd..)



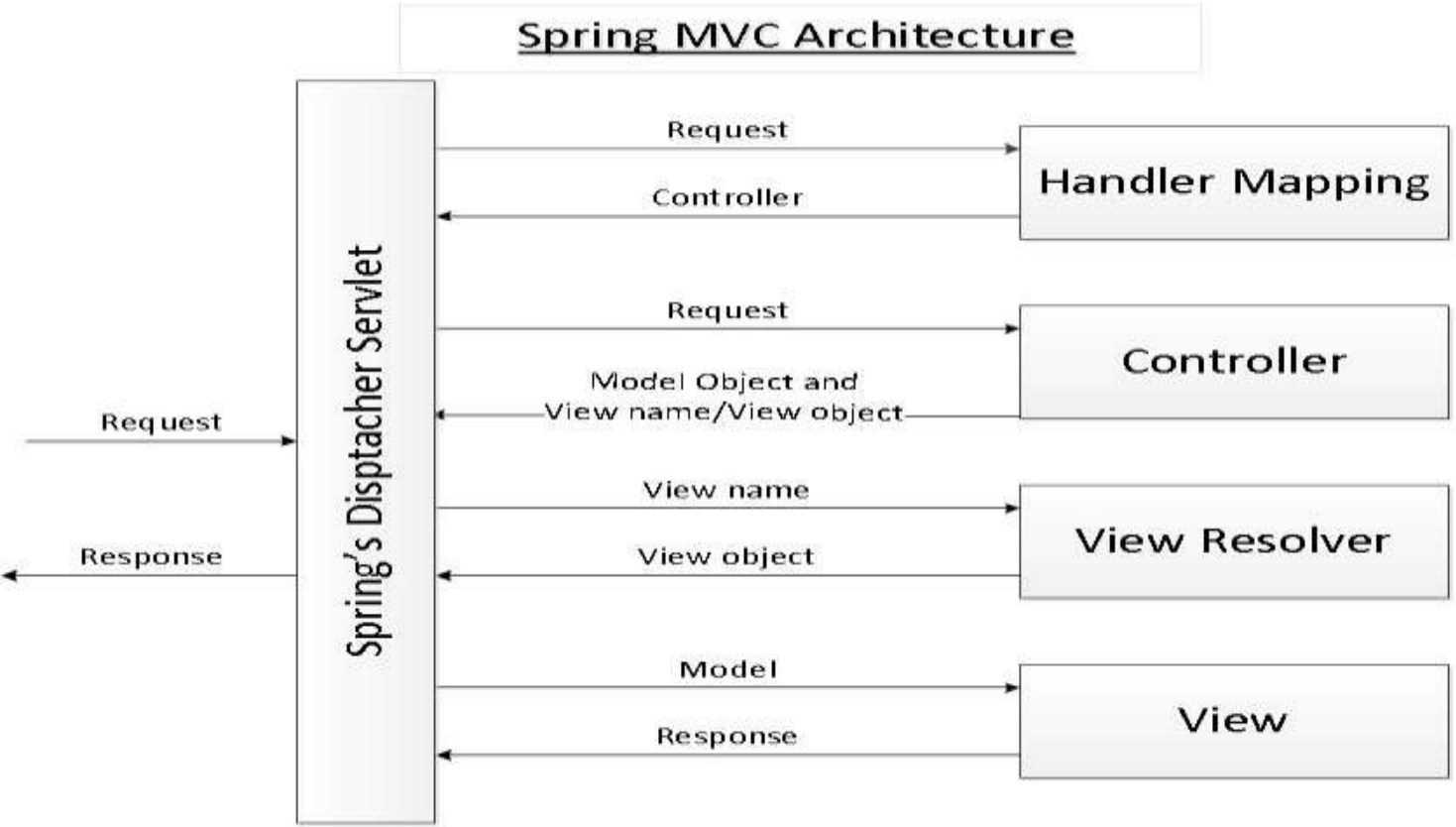
Spring's MVC module

- Spring's MVC module is based on front controller design pattern followed by MVC design pattern.
- All the incoming requests are handled by the single servlet named DispatcherServlet which acts as the front controller in Spring's MVC module.
- The DispatcherServlet then refers to the HandlerMapping to find a controller object which can handle the request.
- DispatcherServlet then dispatches the request to the controller object so that it can actually perform the business logic to fulfil the user request. (Controller may delegate the responsibility to further application objects known as service objects).

Spring's MVC module (Contd..)

- The controller returns an encapsulated object containing the model object and the view object (or a logical name of the view).
- In Spring's MVC, this encapsulated object is represented by class ModelAndView.
- In case ModelAndView contains the logical name of the view, the DispatcherServlet refers the ViewResolver to find the actual View object based on the logical name.
- DispatcherServlet then passes the model object to the view object which is then rendered to the end user.

Spring's MVC module (Contd..)



Dispatcher Servlet

- DispatcherServlet acts as the front controller in the Spring's MVC module. All the user requests are handled by this servlet.
- Since this is like any other servlet, it must be configured in the application's web deployment descriptor file i.e. web.xml.

Dispatcher Servlet (Cond..)

```
<web-app>
  <display-name>TrainingInstituteManagement</display-name>
  <servlet>
    <servlet-name>TrainingInstituteManagement</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>TrainingInstituteManagement</servlet-name>
    <url-pattern>*.html</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>welcome.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Spring Application Context -Default Application context file

- By default the dispatcher servlet loads the Spring application context from XML file with name [servlet name]-servlet.xml.
- Thus when our servlet TrainingInstituteManagement is loaded by the container, it will load the Spring application context from XML file “/WEB-INF/TrainingInstituteManagement-servlet.xml”.

Spring Application Context -User defined application context file

- We can override the name and location of the default XML file by providing the initialization parameters to the dispatcher servlet. The name of the initialization parameter is contextConfigLocation. The parameter value specifies the name and location of the application context which needs to be loaded by the container.

```
<servlet>
  <servlet-name>myLibraryAppFrontController</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:TimService.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Spring Application Context - Multiple application context fi

```
<servlet>
  <servlet-name>myLibraryAppFrontController</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:TimServicexml
                 classpath:TimDataSource.xml
                 classpath:TimUserService.xml
                 classpath:TimConfig.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

ModelAndView and ViewResolver - ModelAndView

- Spring's MVC module encapsulates the model object and the view object in a single entity which is represented by the object of class ModelAndView.
- This object contains the model object and view object or the logical name of the view.
- The model object is the application data and the view is the object that renders the output to the user.
- The controller returns an object of ModelAndView to the dispatcher servlet for further processing.

ModelAndView and ViewResolver - ViewResolver

- In case ModelAndView object contains the logical name of the view then the DispatcherServlet needs resolving the view object based on its logical name.
- To resolve the view object, DispatcherServlet take the help of ViewResolver.
- There are number of implementation of view resolver provided by Spring.
- All the view resolvers implement the interface org.springframework.web.servlet.ViewResolver.

InternalResourceViewResolver

- It resolves the logical name of the view to an internal resource by prefixing the logical view name with the resource path and suffixing it with the extension.

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix"value="/WEB-INF/jsp/" />
    <property name="suffix"value=".jsp" />
</bean>
```

- If the logical name of the view returned by the controller in ModelAndView object is Welcome then the view which is shown to the user is /WEB-INF/jsp/Welcome.jsp

Spring Controllers

- Controller is the actual piece of code which performs the business logic to fulfil the incoming request. Controllers may delegate this responsibility to further service objects as well.


eMexo
TECHNOLOGIES
CALL AT : 09513216462
amazon web services
Dev
Ops
Microsoft Azure
Scala
APACHE Spark
BLOCKCHAIN
UiPath
python
ANGULARJS
hadoop
Data Science
eMexo
TECHNOLOGIES
www.emexotechnologies.com

THANK YOU!

Any questions?

You can find us at

- Email: info@emexotechnologies.com
- Call/WhatsApp: +91 9513216462