Spring MVC - I

MVC Overview

Harish B Rao

Talent Transformation | Wipro Technologies

MVC Overview

MVC stands for Model-View-Controller

The Spring Web MVC framework provides MVC architecture and ready components, using which we can develop flexible and loosely coupled web application.

The MVC pattern helps us in creating web applications that separate different aspects of the application development like input logic, business logic, navigation and presentation logic while maintaining loose coupling between these elements.

It is a proven mechanism for building a thin & clean web-tier.

MVC Overview (Contd.).

Three core collaborating components of MVC :

- **Controller**
  - The Controller processes user requests, builds appropriate model and passes it to the view for rendering
- **Model**
  - The Model encapsulates the application data usually represented by POJOs. This essentially means that Models are objects that are used to hold business data. They may also handle business logic.
- **View**
  - The view is responsible for rendering the model data and it generates HTML response for a particular request.

Advantages of using MVC pattern

Eases maintenance burden

- Changes to business logic are less likely to break the presentation logic and Vice versa

Facilitates multi-disciplined team development

- Developers can focus on creating robust business code without having to worry about breaking the UI
- Designers can focus on building usable and engaging UIs without worrying about Java

Use the best tool for the job

- Java is especially suited to creating business logic code
- Markup languages are more suited to creating User Interfaces

Easy testability

- Business and navigation logic are separated from presentation logic meaning they can be tested separately

Separation of Concern

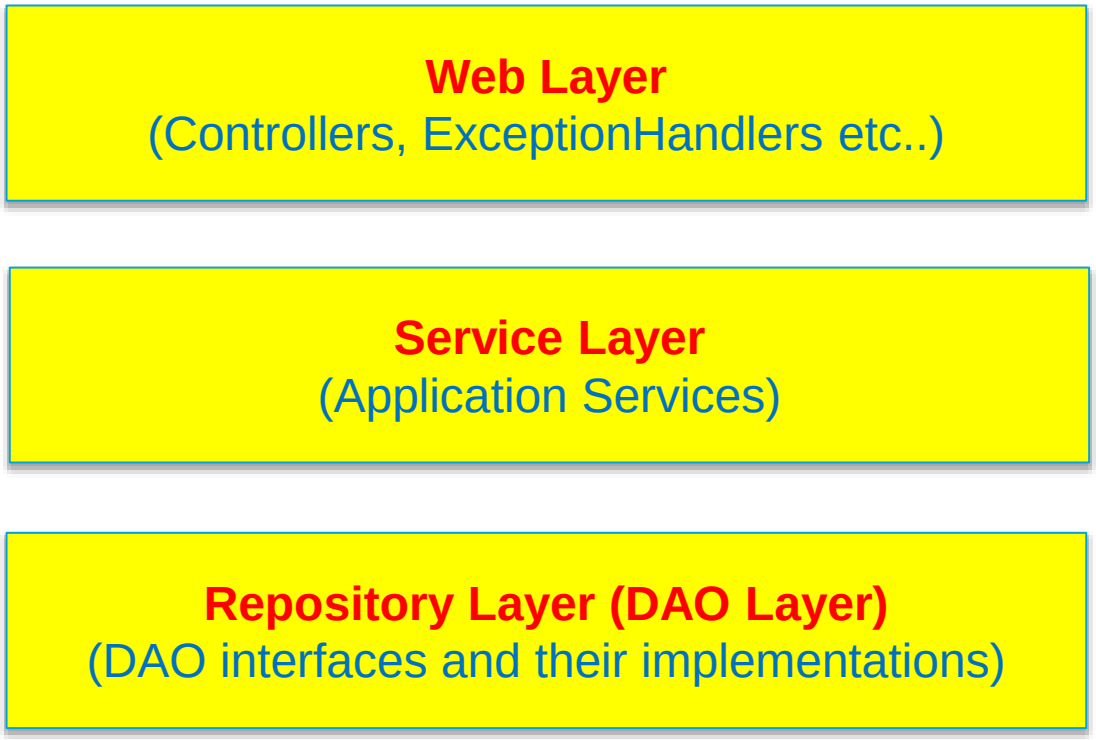
Separation of concern is a design principle for separating a computer application into distinct sections, so that each section addresses a separate concern.

Our job as developers would be to identify these concerns and decide where we would like to handle them.

In short, Separation of concern helps us in identifying these layers and responsibilities of each layer.

Three Layers of a web application

The high level architecture of a typical Spring web application may look as follows :



Three Layers of a web application (Contd.).

Web Layer :

The web layer is the uppermost layer of a web application. It is responsible for processing user's inputs and takes care of the response to be sent to the user. This layer is also responsible for handling exceptions thrown by other layers.

Service Layer :

The service layer is stacked below the web layer. It acts as a transaction boundary and contains application services. Service layer exposes the business functionality of the application, allowing better support for multiple client types.

Repository Layer :

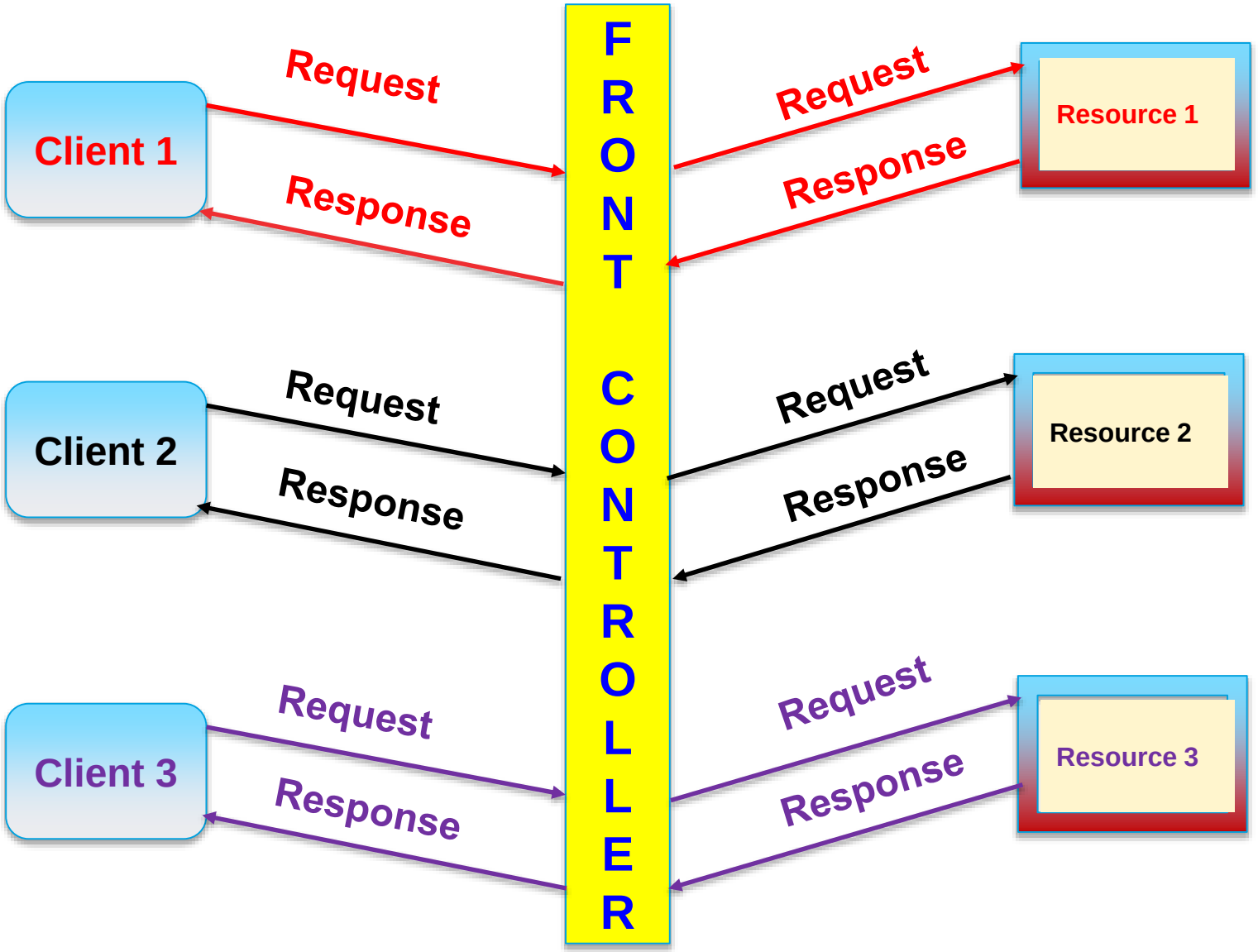
The repository layer(also known as DAO Layer) is the lowest layer of a web application. It is responsible for your application's communication with the database(repository).

Front Controller Design Pattern

The front controller design pattern is used to provide a centralized request handling mechanism so that all requests will be handled by a single handler.

This handler can do the authentication, authorization, logging or tracking of request and then pass the requests to corresponding handlers.

Front Controller Design Pattern (Contd.).



Thank You