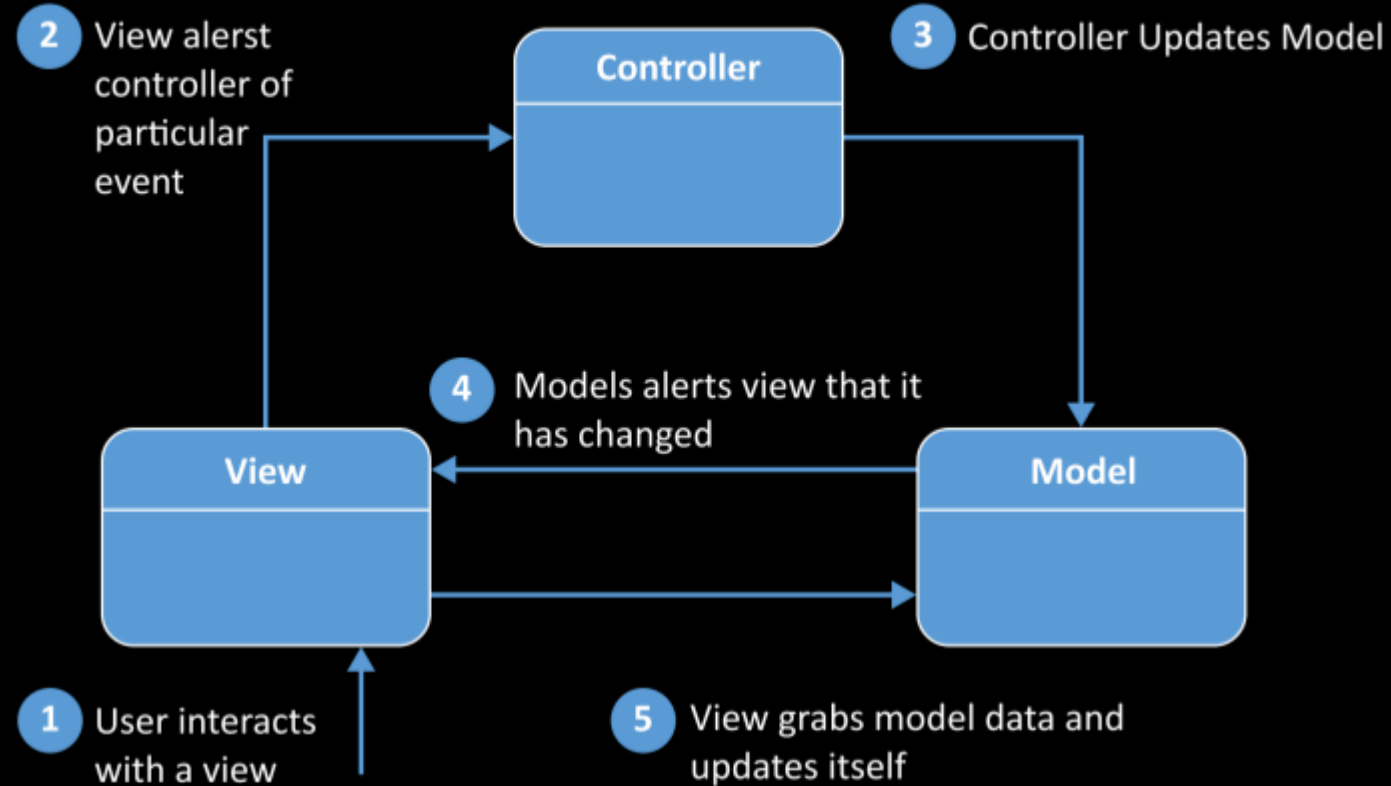


Spring MVC

MVC Architecture



Spring Web MVC

Model

The **Model** encapsulate s the application data and in general they will consist of POJO

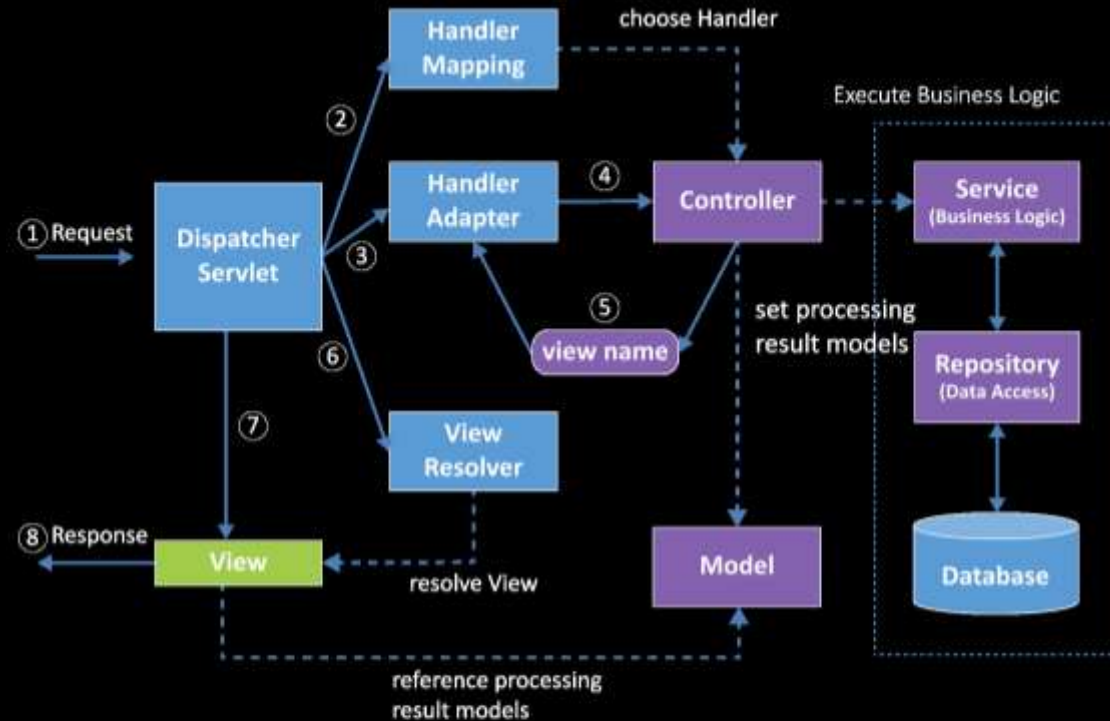
View

The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret

Controller

The **Controller** is responsible for processing user requests and building an appropriate model and passes it to the view for rendering.

request lifecycle diagram



@Controller annotation

There are some important points regarding the controller defined above:

```
@Controller
public class HelloController{

    @RequestMapping(value = "/hello", method = RequestMethod.GET)
    public String printHello(ModelMap model) {
        model.addAttribute("message", "Hello Spring MVC Framework!");
        return "hello";
    }
}
```

- 1) You will define the necessary business logic inside the service method. You can call another method within this method as per requirement.
- 2) Based on the defined business logic, you will create a model in this method. You can set various attributes on the model and these attributes will be available to the view to represent the result. This example creates a model with the attribute "message".
- 3) A specific service method can return a string that contains the name of the **view** that will be used to render the model. This example returns "hello" as the name of the logical view.