**Spring Boot-JSP-CRUD-Application**

**Overview:**

Quick introduction of the **SpringBootServletInitializer**

This is an extension of **WebApplicationInitializer** which runs a **Spring Application** from a traditional **WAR** archive deployed on a web container. This class binds **Servlet**, **Filter** and **ServletContextInitializer** beans from the application context to the server.

Extending the **SpringBootServletInitializer** class also allows us to configure our application when it's run by the **Servlet container**, by overriding the **configure**() method.

**Application. Properties File:**

spring.view.prefix: /WEB-INF/jsp/

spring.view.suffix: .jsp

ModelAndView

Spring ModelAndView - > use of ModelAndView in a controller in a spring web application.

**ModelAndView**

ModelAndView is a holder for both Model and View in the web MVC framework. These two classes are distinct; ModelAndView merely holds both to make it possible for a controller to return both model and view in a single return value.

The view is resolved by a ViewResolver object; the model is data stored in a Map.

**Differences between @RequestParam and @PathVariable annotations in Spring MVC?**

The Spring MVC framework, one of the most popular frameworks for developing a web application in Java world also provides several useful annotations **to extract data from the incoming request and mapping the request to the controller, like @RequestMapping, @RequestParam, and @PathVariable.** Even though both @**RequestParam** and @**PathVariable** are used to extract values from the HTTP request, there is a subtle difference between them.

As the name suggests, @RequestParam is used to get the request parameters from URL, also known as query parameters, while @PathVariable extracts values from URI.

For example, if the incoming HTTP request to retrieve a book on topic "Java" is **http://localhost:8080/shop/order/1001/receipts?date=12-05-2017**, then you can use the @**RequestParam** annotation to retrieve the query parameter date and you can use @**PathVariable** to extract the orderId i.e. "1001" as shown below:

@RequestMapping(value="/order/{orderId}/receipts", method = RequestMethod.GET)

public List listUsersInvoices( @PathVariable("orderId") int order,

@RequestParam(value = "date", required = false) Date dateOrNull) {

...

}

The required=false denotes that the query parameter can be optional, but the URL must have the same URI.

**@modelattribute-vs-@requestbody**

As the javadoc suggests, it's the usage that sets them apart, i.e., use @**ModelAttribute** if you want to bind the object back to the web view, if this is not needed, use @**RequestBody**