

Spring MVC

Presented by Anil Joseph(anil.jos@gmail.com)

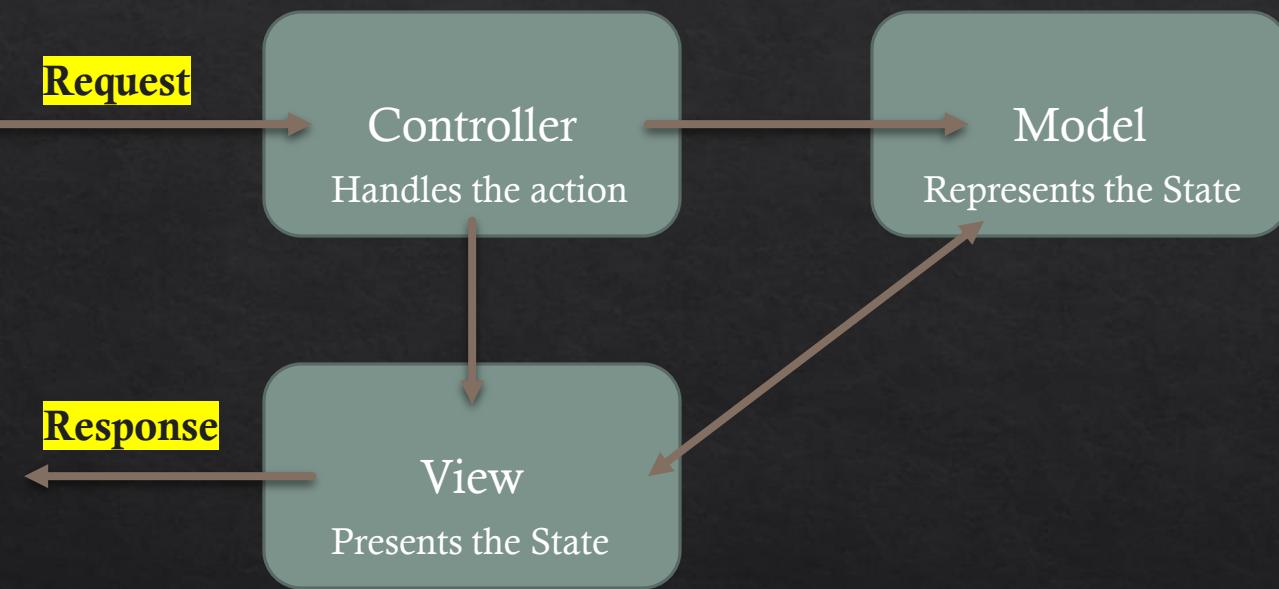
Spring MVC

- ❖ Spring provides a framework for building web applications using the MVC design pattern.

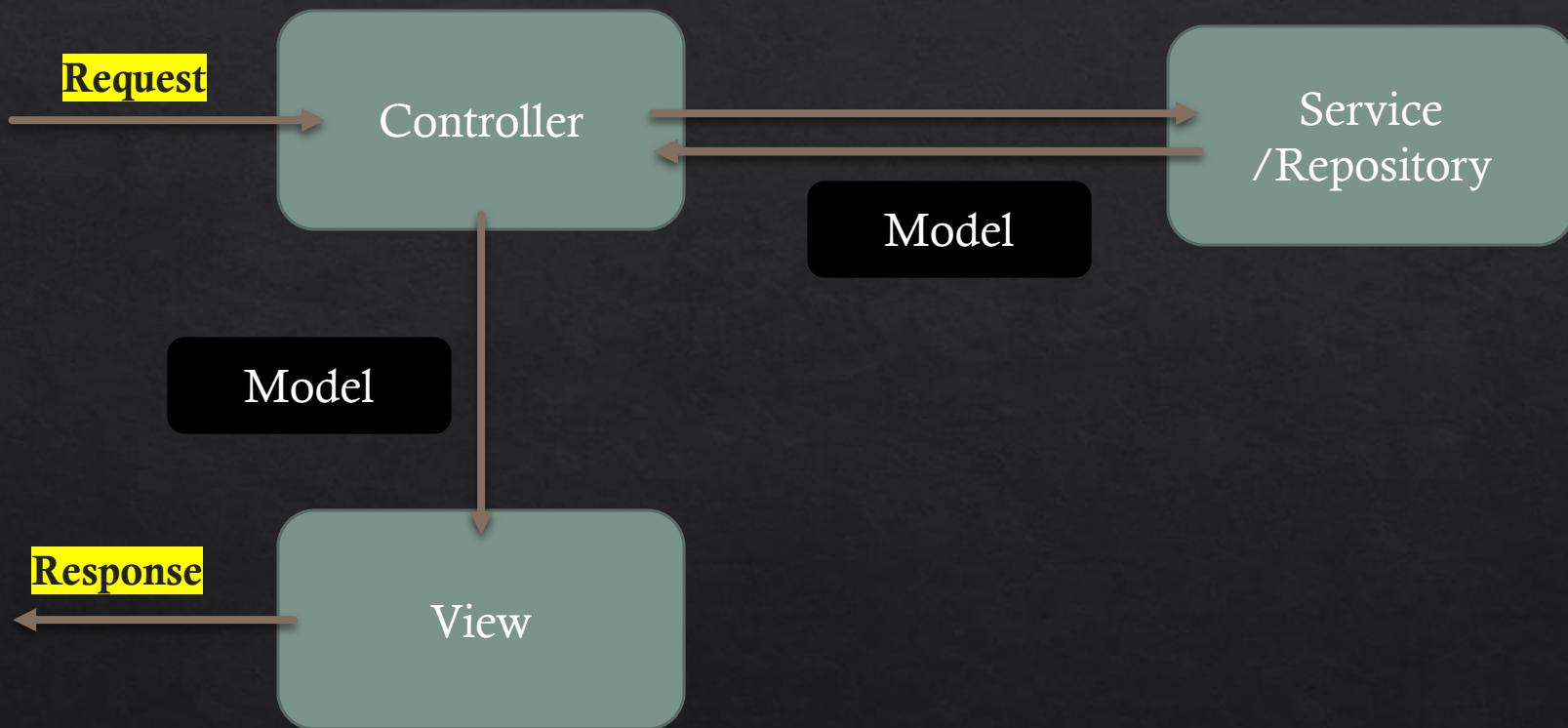
Features

- ❖ Clear separation of roles.
- ❖ Adaptability, non-intrusiveness, and flexibility.
- ❖ Reusable business code, no need for duplication.
- ❖ Customizable binding and validation.
- ❖ Customizable handler mapping and view resolution
- ❖ A JSP form tag library, introduced in Spring 2.0, that makes writing forms in JSP pages much easier.

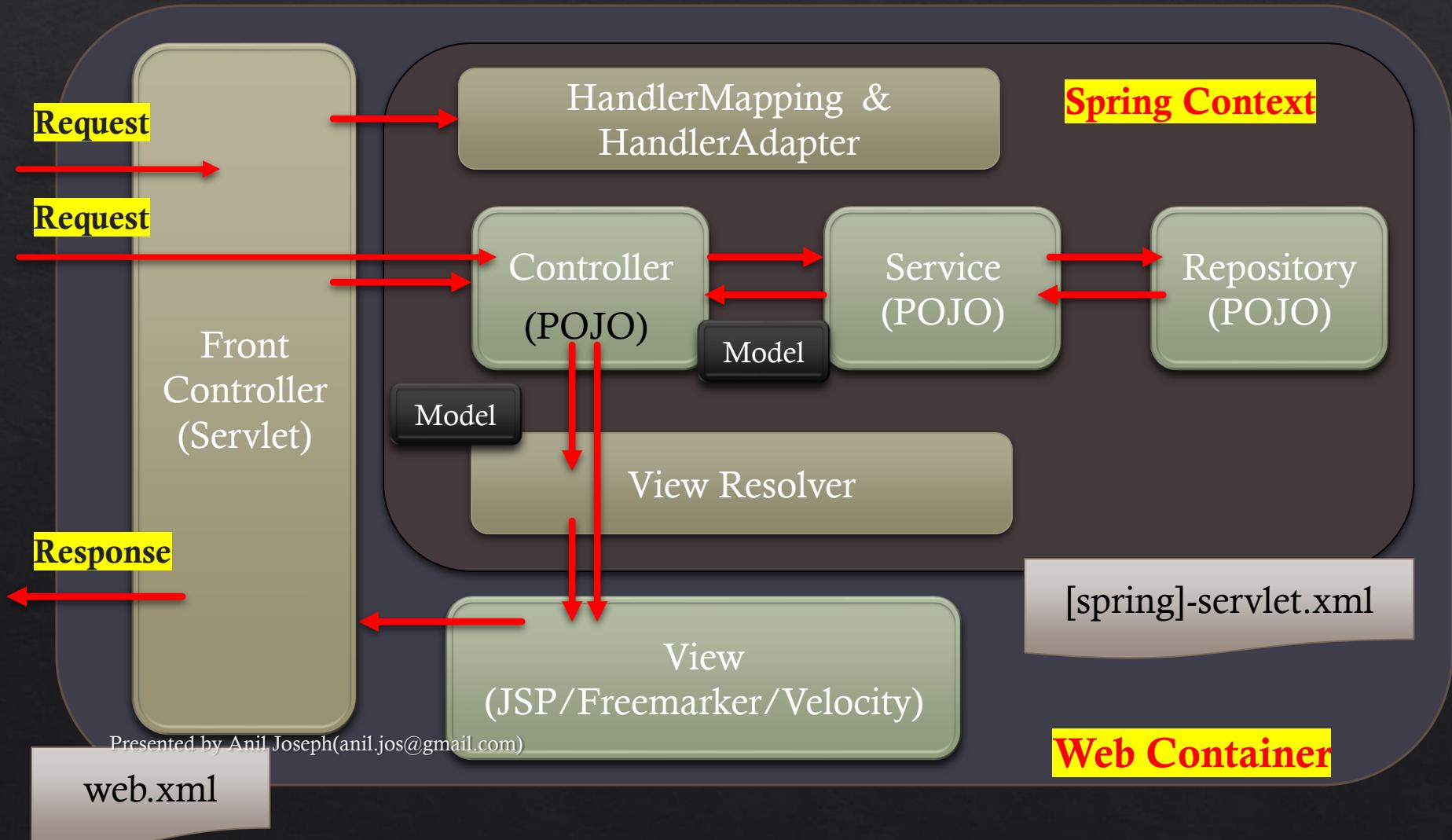
MVC(Model View Controller)



MVC(Model View Controller)



Spring MVC Stack



FrontController

- ❖ The DispatcherServlet is the front controller in the MVC stack of spring
- ❖ The primary role of the servlet is to dispatch the requests to the controllers for processing
- ❖ The servlet loads the spring application context and hence has access to all the spring beans and features
- ❖ The implementing class is
org.springframework.web.servlet.DispatcherServlet

Controller

- ❖ Spring 1.0 & 2.0
 - ❖ Controller(interface) & AbstractController
- ❖ Spring 2.5 onwards
 - ❖ @Controller annotation

Handler Mappings

- ❖ BeanNameUrlHandlerMapping
 - ❖ Maps controller URL's based on the controller bean's name
- ❖ SimpleUrlHandlerMapping
 - ❖ Maps controllers to URL's using a property collection defined in the context configuration file.
- ❖ For annotation based controllers we have 2 handlers
 - ❖ **RequestMappingHandlerMapping**
 - ❖ For mapping controller annotated classes to URL
 - ❖ **RequestMappingHandlerAdapter**
 - ❖ The component invokes the handler method

Handler Mappings Attributes

- ❖ Order
 - ❖ Based on the value of the order, Spring decides the order of invoking handlers.
- ❖ Interceptors
 - ❖ The list of interceptors to use
 - ❖ Interceptors implement ***HandlerInterceptor*** interface

View Resolvers

- ❖ InternalResourceViewResolver
 - ❖ Looks for the view internally
- ❖ XmlViewResolver
 - ❖ accepts a configuration file written in XML with the same DTD as Spring's XML bean factories
 - ❖ The default configuration file is */WEB-INF/views.xml*.
- ❖ VelocityViewResolver
- ❖ FreeMarkerViewResolver

Handler Method Arguments

- ❖ Handler methods that are annotated with `@RequestMapping` can have very flexible signatures.
- ❖ The possible parameters are
 - ❖ Request or response objects (Servlet API).
 - ❖ Session object (Servlet API): of type `HttpSession`.
 - ❖ `java.io.InputStream` / `java.io.Reader` for access to the request's content.
 - ❖ `java.io.OutputStream` / `java.io.Writer` for generating the response's content.
 - ❖ Annotated method parameters

Handler Method Annotated Arguments

- ❖ `@RequestParam`
 - ❖ Maps a request parameter to a handler methods parameter.
- ❖ `@PathVariable`
 - ❖ Maps a path on the request to a method parameter.
- ❖ `@CookieValue`
 - ❖ Maps a request cookie to a handler methods parameter.
- ❖ `@RequestHeader`
 - ❖ Maps a request header to a handler methods parameter
- ❖ `@ModelAttribute`
 - ❖ Maps the request parameters to a model object

Handler method return values

- ❖ A String value that is interpreted as the logical view name or as a redirect URL.
- ❖ A ModelAndView object
 - ❖ **org.springframework.web.servlet.ModelAndView**
- ❖ A View object
 - ❖ **org.springframework.web.servlet.View**
- ❖ A String value that is the response. Use the @ResponseBody annotation.

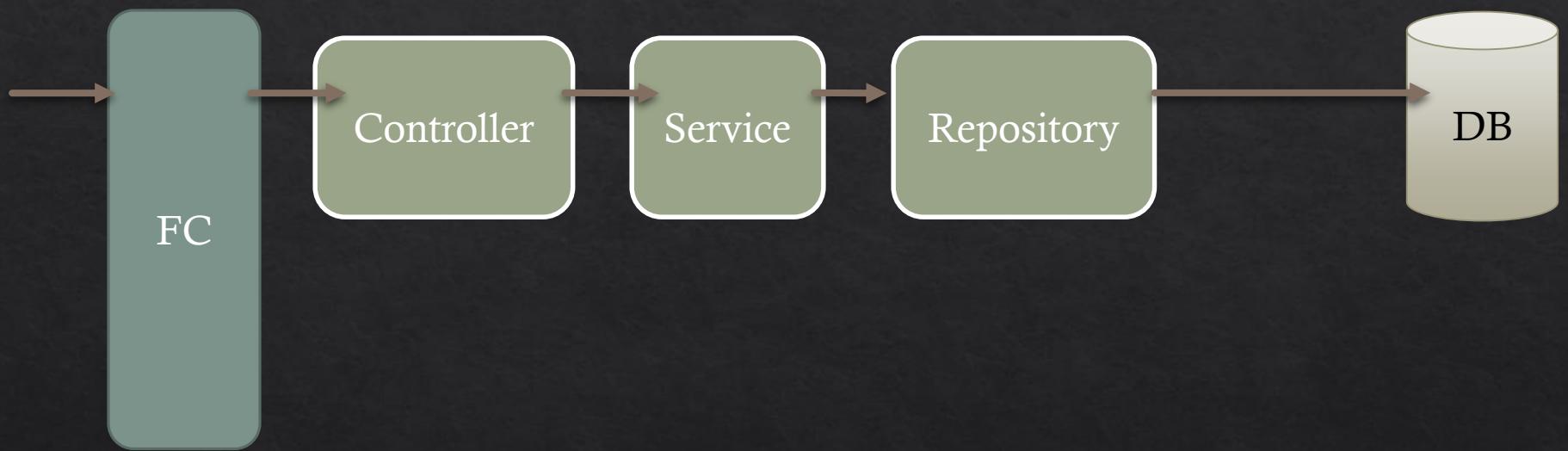
Form Validations

- ❖ JSR 303 Bean Validation API's
 - ❖ Annotation based
 - ❖ Defined at the Model
 - ❖ Custom annotations for custom validations
- ❖ Validator Interface
 - ❖ Implementation can define validations on a single field or across fields
 - ❖ Defined at the controller

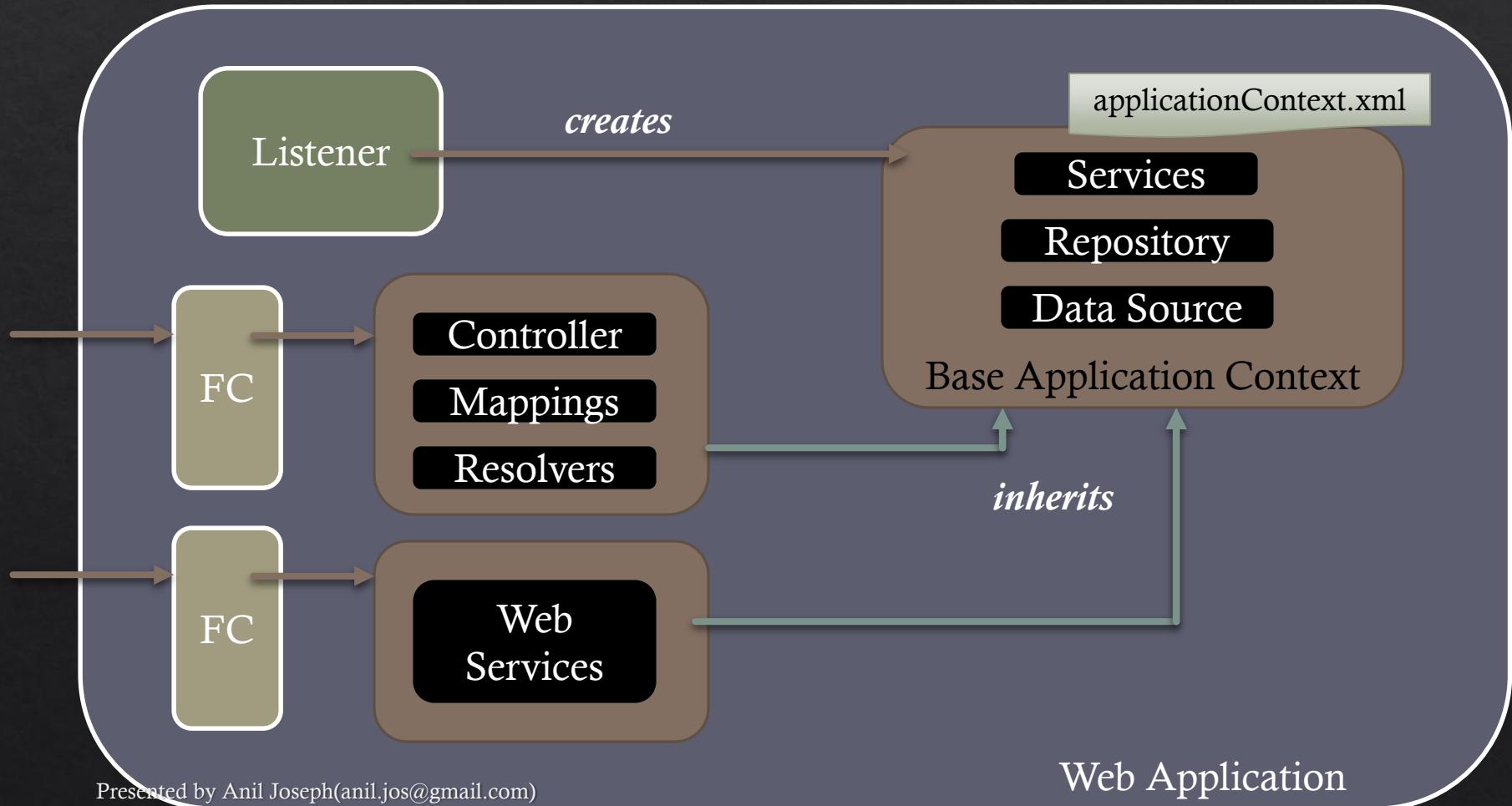
MVC Annotations

- ❖ `@Controller`
 - ❖ To define a controller
- ❖ `@RequestMapping`
 - ❖ Mapping a request URI to a controller or a controller method
- ❖ `@RequestParam`
 - ❖ Maps a request parameter to a handler methods parameter
- ❖ `@CookieValue`
 - ❖ Maps a request cookie to a handler methods parameter
- ❖ `@RequestHeader`
 - ❖ Maps a request header to a handler methods parameter
- ❖ `@ResponseBody`
 - ❖ Applied to a handler method
 - ❖ Indicates that the return type should be written straight to the HTTP response body
- ❖ `@ModelAttribute`
 - ❖ Maps the request parameters to a model object

Application Design



Application Design



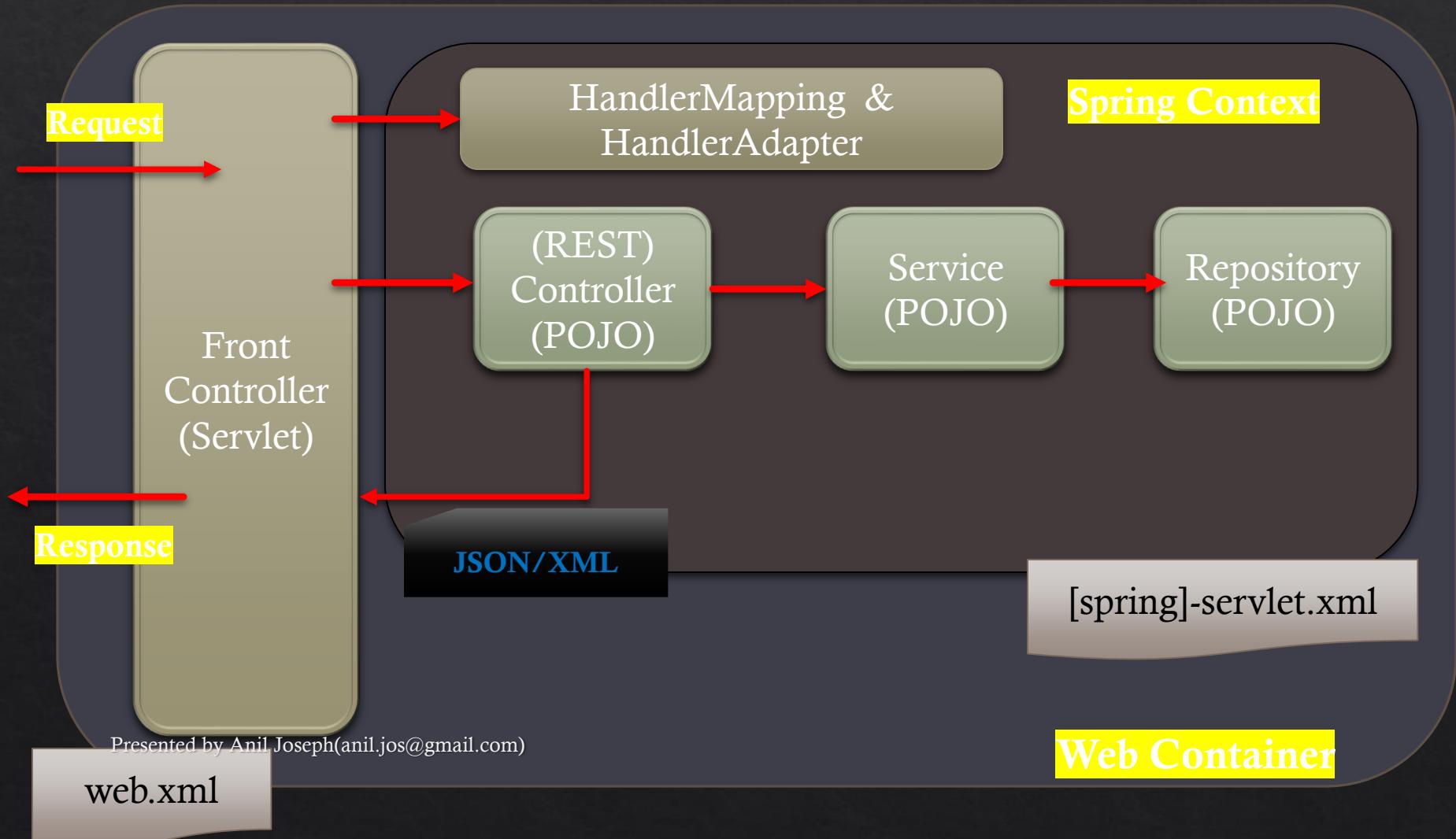
RESTful Web Services

- ❖ *RE*presentational *S*tate *T*ransfer *Web Services*.
- ❖ REST provides a set of design rules for creating stateless services that are viewed as *resources*,
- ❖ A client accesses the resource using the URI and a *representation of the resource is returned*
- ❖ Services are accessed by an HTTP method without the use of a message protocol(SOAP).

Spring MVC REST

- ❖ Spring MVC supports the creation of REST Web Services.
- ❖ `@RestController` introduced in Spring 4.0
 - ❖ Defines a controller for REST
 - ❖ Combines `@Controller` & `@ResponseBody`
- ❖ `@RequestMapping`
 - ❖ Maps a method to a request URL

Spring MVC Stack



Spring MVC REST

- ❖ `@RequestMapping`
 - ❖ value: defines the path
 - ❖ method: defines the HTTP method to invoke the resource
 - ❖ produces: defines the MIME types exposed
 - ❖ consumes: defines the MIME types accepted
 - ❖ params: defines the request parameters accepted
 - ❖ headers: defines the headers accepted.

Data Exchange Types

- ❖ Data Exchange Types
 - ❖ Plain Text
 - ❖ XML
 - ❖ JSON
 - ❖ HTML
 - ❖ Or any type represented with a MIME type

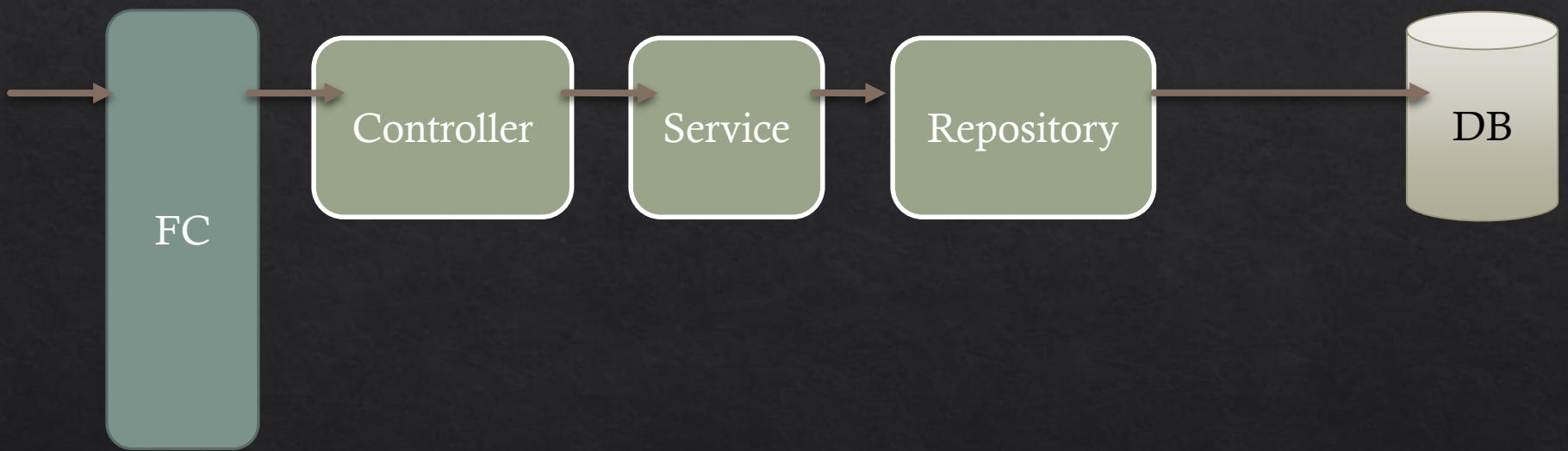
Spring Jdbc

- ❖ The value-add provided by the Spring Framework as a JDBC abstraction.
- ❖ Spring JDBC takes care of all the low-level details that can make JDBC such a tedious API to develop with.
- ❖ Advantages its provides
 - ❖ Open the connection.
 - ❖ Prepare and execute the statement.
 - ❖ Handles the code of iteration
 - ❖ Processing Exceptions
 - ❖ Handles transactions
 - ❖ Close the connection, statement and resultset.

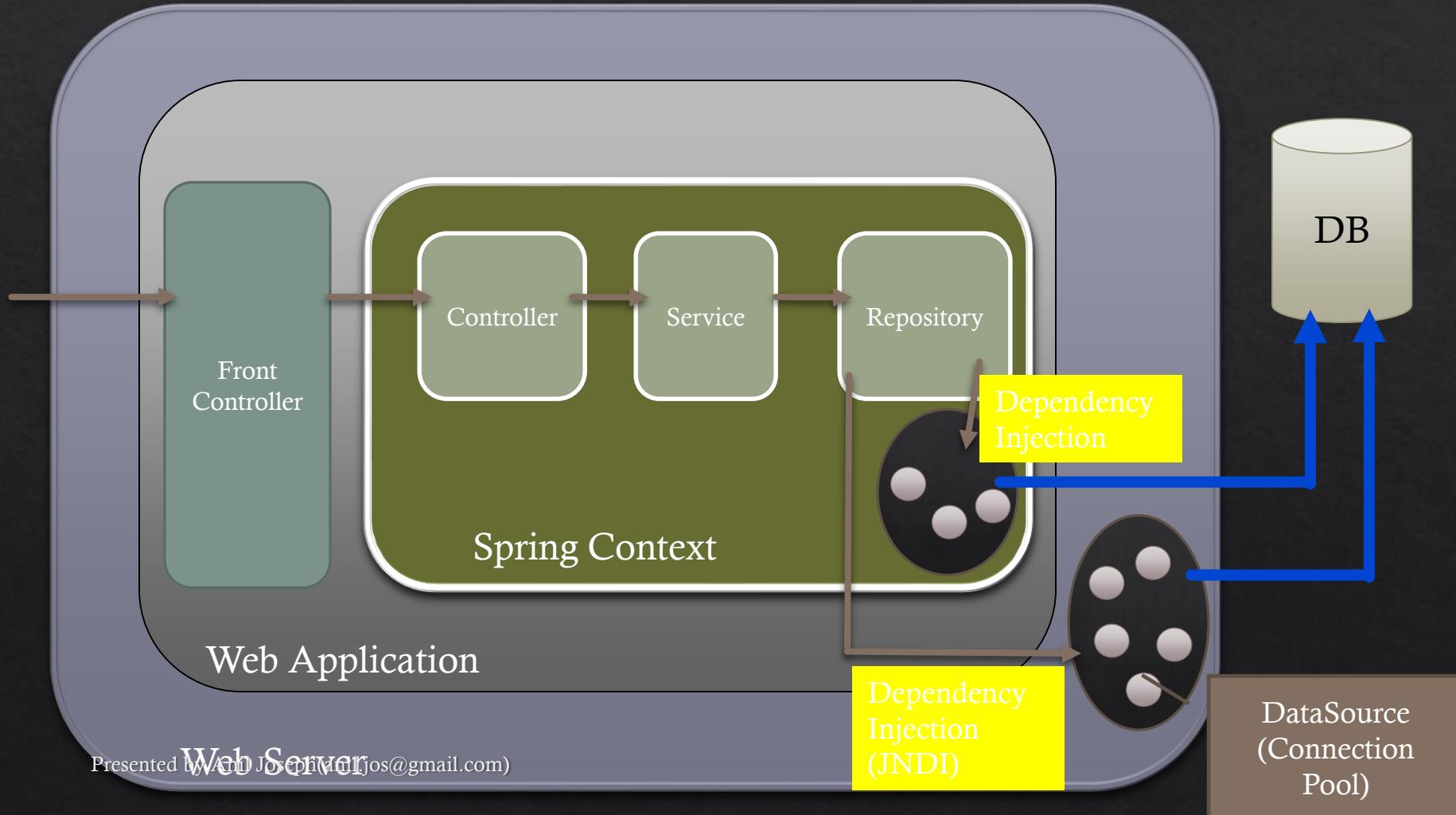
Jdbc Classes

- ❖ JdbcTemplate
 - ❖ The classic Spring JDBC class
- ❖ NamedParameterJdbcTemplate
 - ❖ To provide named parameters instead of the traditional JDBC "?" placeholders.
- ❖ SimpleJdbcInsert and SimpleJdbcCall
 - ❖ Optimize database metadata to limit the amount of necessary configuration.

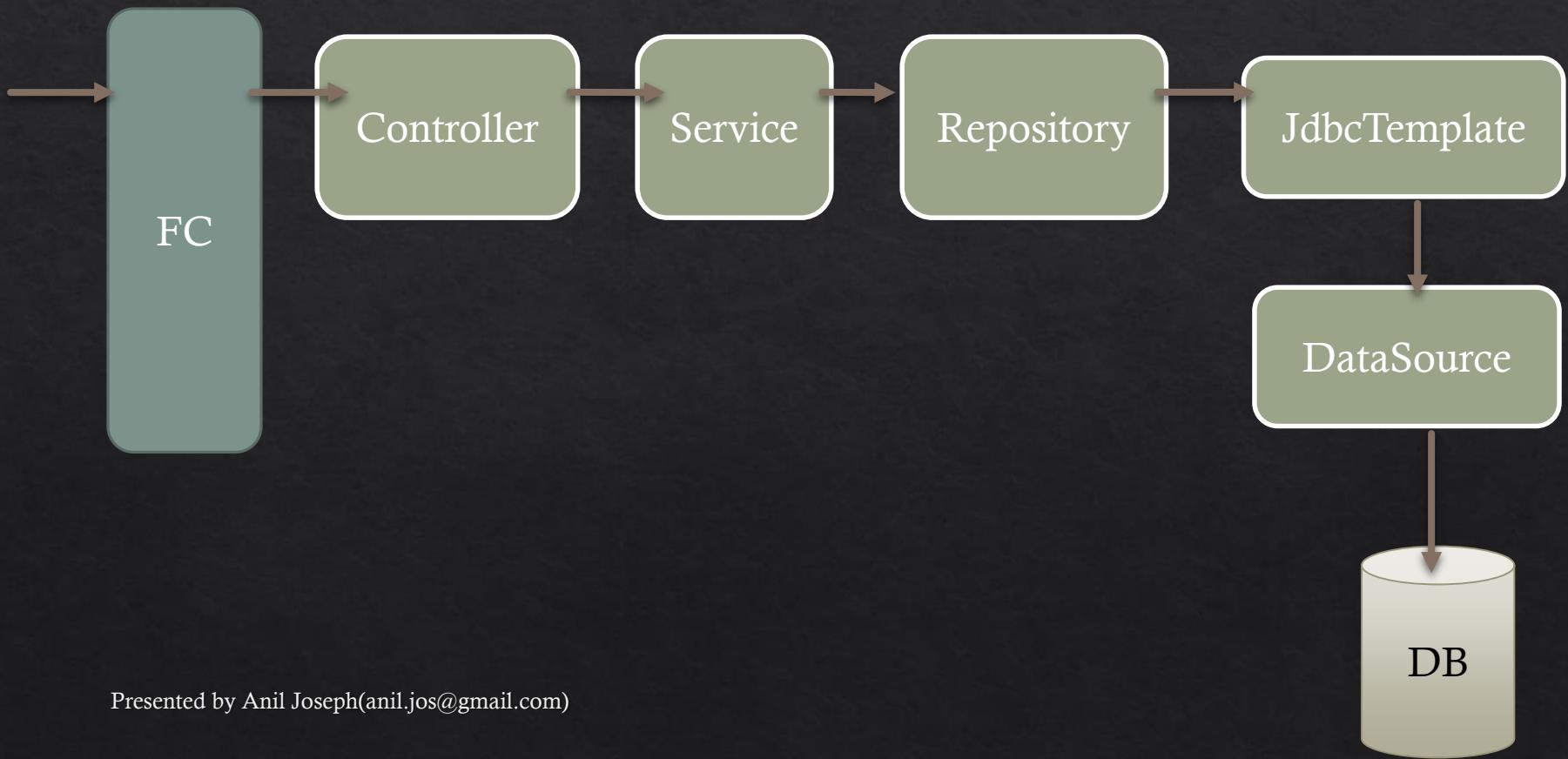
JDBC Application Design



JDBC Application Design



JDBC Application Design



Spring 4 New Features

- ❖ Annotation
 - ❖ @Conditional
 - ❖ @RestController
 - ❖ @CrossOrigin
 - ❖ @Description
- ❖ Java 8 Support
- ❖ WebSockets, Http Streaming & Server-sent events

Spring 5 New Features

- ❖ Spring 5 needs a of minimum JDK 8 and Java EE 7
- ❖ Baseline Framework versions
 - ❖ Hibernate 5
 - ❖ Jackson 2.6
 - ❖ EhCache 2.10
 - ❖ JUnit 5
 - ❖ Tiles 3
- ❖ Reactive programming support
- ❖ A functional web framework(WebFlux)
- ❖ Kotlin support

Thank You

Presented by Anil Joseph(anil.jos@gmail.com)