search-trains.jsp

```html
<body>
    <form action="${pageContext.request.contextPath}/findAvailableTrains.htm">
        source: <input type="text" name="source"/> <br/>
        destination: <input type="text" name="destination"/><br/>
        train type:
            <select name="trainType">
                <option value="passenger">passenger</option>
                <option value="goods">goods</option>
                <option value="express">express</option>
                <option value="super fast">super fast</option>
                <option value="vande bharat">vande bharat</option>
            </select><br/>
        travel date: <input type="date" name="travelDate"><br/>
        <input type="submit" value="search trains"/>
    </form>
</body>
```

available-trains.jsp

```html
<body>
    <table>
        <tr>
            <th>train no</th>
            <th>source station</th>
            <th>end station</th>
            <th>train type</th>
            <th>duration</th>
            <th>source departure time</th>
            <th>destination arrival time</th>
        </tr>
        <c:forEach items="${availableTrains}" var="train">
            <tr>
                <td>${train.trainNo}</td>
                …
            </tr>
        </c:forEach>
    </table>
```

requestwrappingweb

```
requestwrappingweb
|-src
    |-main
        |-java
        |-resources
        |-webapp
            |-WEB-INF
                |-jsp
                    |-search-trains.jsp
                    |-available-trains.jsp
```

```java
class SearchTrainsForm {
    String source;
    String destination;
    String trainType;
    LocalDate travelDate;
    // accessors
}
```

/search-trains.htm
/findAvailableTrains.htm

```java
class RWDispatcherServletInitializer
extends
AbstractAnnotationConfigDispatcherServlet
Initializer {
    Class<?>[] getRootConfigClasses(){}
    Class<?>[] getServletConfigClasses() {}
    String[] getServletMappings(){}
}
```

```java
@Controller
class TrainController {
    @RequestMapping("/findAvailableTrains.htm")
    public String
searchAvailableTrains(@ModelAttribute("searchTrainsF
orm") SearchTrainsForm form, Model model) {
        // business operation
        // bind the response data into model
        return "available-trains";
    }
}
```

@ModelAttribute = indicates bind the request data
into the model object, attributes
RequestMappingHandlerAdapter takes
1. query parameters
2. form data (www-form-url-encoded) (requestbody)
and wraps into the specified model object attributes
and binds the object into request scope with given
name and pass it as parameter in calling the
handlerMethod of the controller

```java
@Configuration
@EnableWebMvc
@ComponentScan(basePackages={"com.rw.controller"})
class WebMvcConfig implements WebMvcConfigurer {

    public void
addViewControllers(ViewControllerRegistry
registry) {
        registry.addViewController("/search-
trains.htm").setViewName("search-trains");
    }

    public void
configureViewResolver(ViewResolverRegistry
registry){
        registry.jsp("/WEB-INF/jsp/", ".jsp");
    }
}
```

The RequestMappingHandlerAdapter by looking at @ModelAttribute("key") in the form object
paramter of the handlerMethod
1. it understands it has to extract the data from query parameters of the URI and request body
(form data)
2. then instantiate the object of the form class specified
3. Bind the data into the form object based on paramName matching with attributeName of the form
4. put this form object into request scope with name as "key"
5. then pass it as an parameter to the Controller class handlerMethod and invoke it.

HTTP REQUEST

```
HTTP/1.1
URI      --------[PATH VARIABLES /
                  QUERY PARAMS /
                  MATRIX PARAMS]
METHOD:
-------------------------
HEADER


-------------------------
BODY
[www-form-url-encoded]
emailAddress=alex@gmail.com;password=welcome1;firstName=alex;lastNam
e=L;age=23;gender=Male
```

To RequestMappingHandlerAdapter:
if the data is send aspart of the http request in
1. URI as query parameters
2. request body as www-form-url-encoded
then wrap this key/value pair data into an form object using
@ModelAttribute annotation