Rêve – Event Booking System

~~**Friday**: https://meet.google.com/pyw-mzyk-zrg?hs=224~~

~~**Monday**:~~

/ = Root directory
. = This location
.. = Up a directory
./ = Current directory
../ = Parent of current directory
../../ = Two directories backwards

To run Project:

>> Run as: Spring Boot App

>> Browser: http://localhost:8082/ (homepage)

Further Enhancements:

- Better login pages
- Incorporate **Spring security** (Only logged-in Users have access)
- **Portal for customers** to see their bookings after login (* Bookings made, cancel bookings, view past bookings)
- Event master can see bookings done for each event – portal better
- MAX seat that can be fixed set to a fixed limit (10 seats per booking) – Javascript + consequent error message

- Default value for attendance >> Set 0
  *Event Attendance + no of seats >> [?] EDit disable Attendance
- Event Name editable, Event name not required in Bookings Table
- Event Details changes by admins:(Created By, Modified Date, Modified By)
* Dynamically changing date and time: JQuery AJAX-----CODE CHECK
* Booking Success Page - JQuery AJAX message
- alert for startDate < endDate: edit/Add Event

>> GitHub Pages:

https://stackoverflow.com/questions/36782467/set-subdirectory-as-website-root-on-github-pages

>> **Font Awesome Pro CDN**: https://www.grepper.com/answers/590388/fontawesome+pro+icon+cdn

```
<!-- FA cheat pro link v5.10-->
<link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css" integrity="sha384-AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p" crossorigin="anonymous">
```

JSP Pages:

Can't add JAVA code to HTML…either use Thymeleaf templating engine (for HTML) or use JSP pages.

https://www.baeldung.com/spring-boot-jsp

ThymeLeaf: Templating engine for using HTML pages, instead of JSP.

```
<span th:text="${}"></span>
<html xmlns:th="https://www.thymeleaf.org">
```

Changing default Thymeleaf templates location…. [Added in src/main/resources/templates/pages]

```
spring.thymeleaf.prefix=classpath:templates/pages/
```

```
th:href="@{/styles/cssandjs/main.css}"<img th:src="@{/images/test.png}"/>
```

```
th:href="@{/}"
```

JSTL Tags

Core (c) tags

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

<a href="<c:url value="/test.html"/>">TEST

<c:url value="/test"/> ->for external URLs…ex. CSS stylesheet..put in href="…"

<c: if>
For test Conditions: https://community.oracle.com/tech/developers/discussion/1401699/jsp-attribute-in-c-if-to-add-class-on-li

If-Else Conditions:
https://stackoverflow.com/questions/4587397/how-to-use-if-else-option-in-jstl

For formatting dates while putting in HTML

```
<%@ taglib prefix = "fmt" uri = "http://java.sun.com/jsp/jstl/fmt" %>
```

https://www.tutorialspoint.com/jsp/jstl_format_formatdate_tag.htm

<fmt:formatDate pattern="MM/dd/yyyy" value="${obj.date}" /> [custom]

Working with Date Parameter in Spring & JavaScript

Accepted JavaScript inputs: https://www.w3schools.com/js/js_date_formats.asp

`Iso format:` To **set value="" to date/datetime-local type <input>** in html. Can be hardcoded in HTML to set with the help of JavaScript, to specific dates.

E.g., value="2018-06-12T19:30" min="2018-06-07T00:00" max="2018-06-14T00:00"

To Enable selection of only a specific date range>> Use min="${event.dateStart}" max="${event.dateStart}" attributes of HTML date element. Format with JSTL/Thymeleaf if necessary…(Only reads yyyy-MM-dd'**T**'HH:mm format) in ISO Format.

ref: MDN Docs

[ISO Format: JSTL Tag for JSP Pages]

```
<fmt:formatDate pattern="yyyy-MM-dd'T'HH:mm:ssz" value="${event.starttime}" />
```

[local time -offset.] After converting to ISO through toISOstring() in JavaScript

- ***Solution – 1-***

```
var tzoffset = (new Date()).getTimezoneOffset() * 60000; //offset in milliseconds
var localISOTime = (new Date(Date.now() - tzoffset)).toISOString().slice(0, -1);
```

- ***Solution – 2-***

```
var date = new Date();
date.setMinutes(date.getMinutes() - date.getTimezoneOffset());
var datestr = date.toISOString().substring(0, 16);
```

- ***Solution – 3 - [?]***

Use getDate, getYear etc and format a new string in ISO format for local time…in javascript (not using toISOstring that causes the offset).

Joda-Time library http://joda-time.sourceforge.net/api-release/org/joda/time/format/ISODateTimeFormat.html

https://www.joda.org/joda-time/

https://www.marklogic.com/learn/university/

```
yyyy-MM-dd'T'HH:mm >> input sent in this format from the Form
```

E.g.: 2002-03-11T18%3A58: yyyy-MM-dd'T'HH:mm

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/datetime-local 🌙

```
@DateTimeFormat(pattern = "yyyy-MM-dd'T'HH:mm")
```

https://stackoverflow.com/questions/53547378/spring-form-using-date-and-time-as-input-type-to-be-assigned-to-a-localdate-an

- https://stackoverflow.com/questions/31627992/spring-data-jpa-zoneddatetime-format-for-json-serialization

<input type="datetime-local" id="meeting-time" value="2018-06-12T19:30"

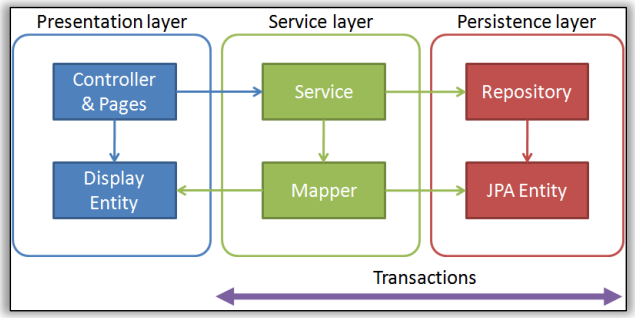
min="2018-06-07T00:00" max="2018-06-14T00:00"> {…format sent to backend – example}

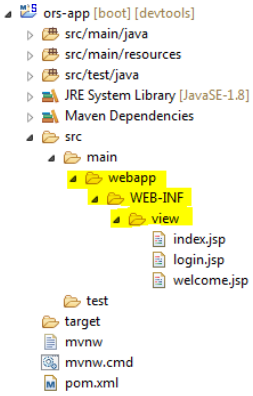
HH:mm – 24 Hour format…. hh:mm aa – 12hour format w am/pm ref. >> More after Post, Mapping section.

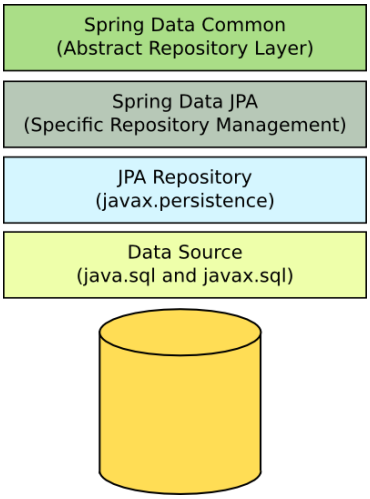
Spring Boot

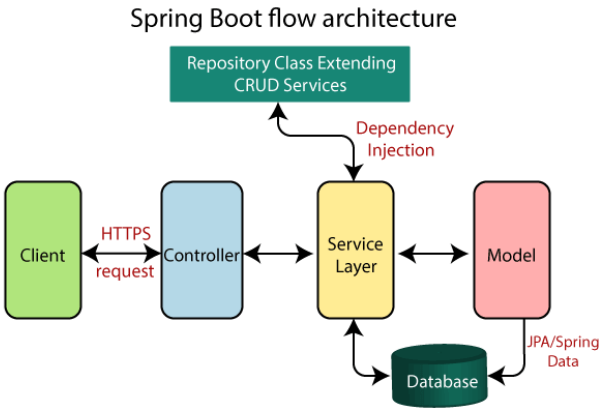
Localhost:8081/

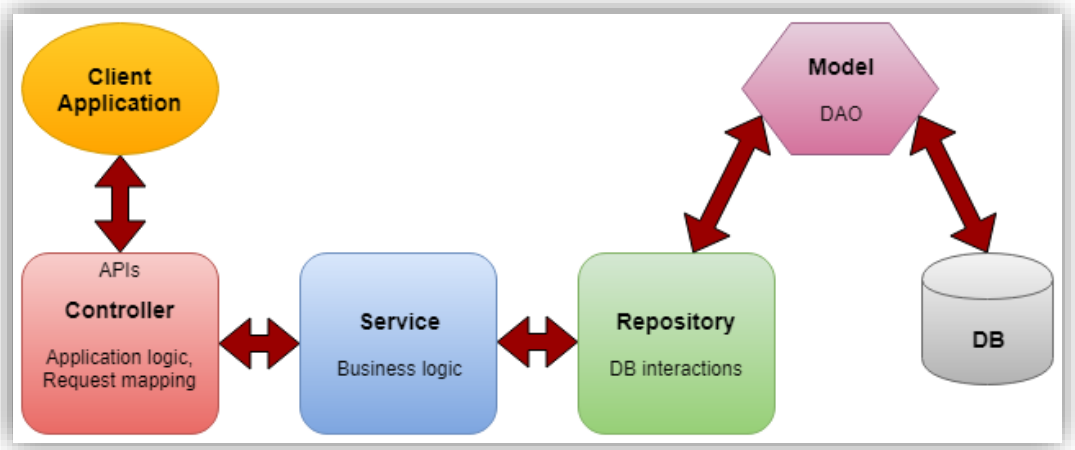
netstat -ano | findstr : port number









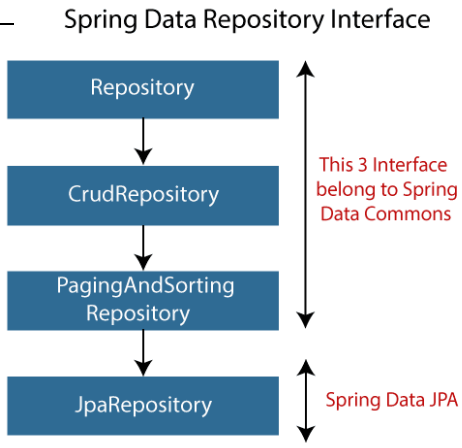


CRUD = Create Read Update Delete

HTTP Requests….

- ✓ POST – Create
- ✓ GET – Read
- ✓ PUT – Update
- ✓ DELETE - Delete

JPA Repo = CRUD + P&S Repos



POST METHOD:

Inserting data into JPA Database from Form

```
<form action="/submitForm" method="post">
```

```
@PostMapping("/submitForm")
```

modelAttribute, commandName

>> for preventing URL change from <form> action="/submitForm" attribute -

Found the answer for my exact case on **17.5.3 Redirecting to views** from Spring's Docs

https://stackoverflow.com/questions/29451670/prevent-form-action-from-changing-url-after-file-upload

> It is sometimes desirable to issue an HTTP redirect back to the client, before the view is rendered. This is desirable, for example, when one controller has been called with POST data, and the response is actually a delegation to another controller (for example on a successful form submission). In this case, a normal internal forward will mean that the other controller will also see the same POST data, which is potentially problematic if it can confuse it with other expected data.

This can be done with the following:

```
return "redirect:/form";
```

A nice benefit to the redirect is that it prevents the user from accidentally re-POSTing the same data by performing a refresh. The refresh forces a GET of the result page, not a resend of the initial POST data.

(/) url becomes "root/form"

DISPLAY DATA FOR A LIST OF RECORDS: Repeating element for each.. like a table row in table for each record.

Thymeleaf: https://www.codementor.io/@olebueziobinnadavid/setting-up-and-displaying-a-list-of-objects-in-a-table-thymeleaf-1cifoviz5e

JSTL Tag: <c:forEach> https://www.baeldung.com/spring-boot-jsp

- Use <c: if> inside it if records are to be displayed selectively.

Mappings:

DBMS Entity relationships… E.g.: @ManyToOne…defined at the time of Entity creation.

>> `CascadeType.ALL` >> All Operations, deletions, updating, detach

>> `CascadeType.PERSIST` >> refresh(), delete, save/edit

```
@ManyToOne
@OnDelete(action = OnDeleteAction.CASCADE)
@JoinColumns({
	@JoinColumn(name = "eventid", referencedColumnName = "eventID"),
	@JoinColumn(name = "event_name", referencedColumnName = "eventName")
})
private Event event;
```

```
@ManyToOne(cascade = CascadeType.PERSIST)
@JoinColumns({
	@JoinColumn(name = "eventid", referencedColumnName = "eventID"),
	@JoinColumn(name = "event_name", referencedColumnName = "eventName")
})
private Event event;
```

`@DateTimeFormat(pattern=”yyyy-MM-dd”)`

Specifies pattern for the date passed into backend.

E.g.: `@RequestParam @DateTimeFormat(pattern="MM/dd/yyyy") Date dateReceived`
https://stackoverflow.com/questions/15164864/how-to-accept-date-params-in-a-get-request-to-spring-mvc-controller

- also used while defining Entity Date member type

https://youtu.be/2JPJ1OOVT3E?feature=shared

`@Temporal(TemporalType.DATE)`

Controls how it is stored in database… Date/Datetime/Time

https://youtu.be/zixMjIbjx0Y?feature=shared

- https://www.tutorialandexample.com/cascade-in-hibernate

---

jQuery AJAX

Prevent redirect after form is submitted, Dynamically update/refresh parts of Webpage

https://youtu.be/3U6R5LJtzjk?feature=shared

Example-1

```
$('#registerform').submit(function(e) {
  e.preventDefault();
  $.ajax({
      type: 'POST',
      url: 'submit.php',
      data: $(this).serialize(),
      beforeSend: //do something
      complete: //do something
      success: //do something for example if the request response is success play your animation...
  });

})
```

Example-2

```
$(function() {
   $('form').submit(function() {
      $.ajax({
         type: 'POST',
         url: 'submit.php',
         data: { username: $(this).name.value,
               password: $(this).password.value }
      });
      return false;
   });
})
```

https://youtu.be/82hnvUYY6QA?feature=shared

>> AJAX Vanilla JS Tutorial

https://youtu.be/8IcTBJ7D0Ek?si=TJrVC8XsKtM-MIWy >> JQuery AJAX Get & Post Examples

References

Study Material

Documentation:

GitHub Documentation: https://github.com/spring-projects

Spring Docs: https://docs.spring.io/spring-framework/docs/current/reference/html/index.html

Spring-boot Docs: https://docs.spring.io/spring-boot/docs/current/reference/html/index.html

https://docs.spring.io/spring-boot/docs/

Tutorials & Playlists:

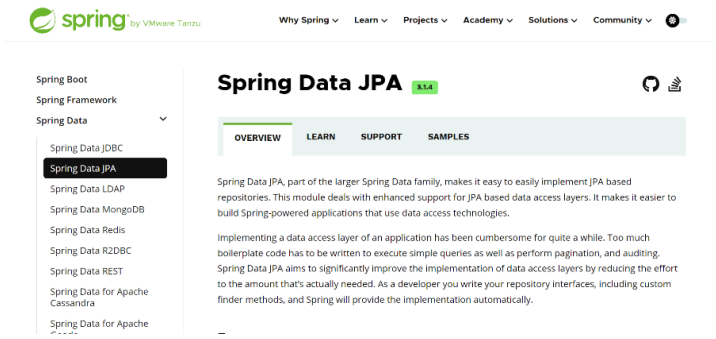
>> Edureka (Basic Applications tutorial)

https://youtu.be/UfOxcrxhC0s?si=67baEeo4tHPMwl58

Spring.io >>

https://spring.io/guides#getting-started-guides

https://spring.io/projects/spring-boot



Java t-point >> https://www.javatpoint.com/spring-boot-tutorial

tutorials point >> https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm

Random Videos:

>> RoadMap (Java, SB):

https://youtu.be/8WUIXGhZURk?si=wR9JXs_xNc4bNDGu , https://youtu.be/TE3LyYW-AHQ?si=dvbNBQ8piuXxWqIa

>>

Videos (for basic tutorials) !! https://youtu.be/_thI-4AF7M8?si=wi5vtmdeN01kQBBB
https://youtu.be/9SGDpanrc8U?feature=shared
https://youtu.be/zvR-Oif_nxg?feature=shared

>> RANDOM

https://youtu.be/aS9SQITRocc?feature=shared - ultimate basics: Spring Beans, Spring Container

https://youtu.be/cRdjxG-Qbj8?feature=shared – Models

https://youtu.be/SFVrzsvknOA?si=XNoTou5TSExzNyFj – understanding @Model

https://youtu.be/aKjJ906lKZ0?feature=shared – DAO Layer

https://youtu.be/c3gKseNAs9w?feature=shared - Full In-depth Course

>> Selenium Express Vids:

https://youtu.be/G39Z3J_6lKY?feature=shared (Service Layer)

https://youtu.be/x_-Tx6OFYLg?feature=shared (annotations)

Code Guides:

https://spring.io/guides/gs/serving-web-content/
https://spring.io/guides/gs/spring-boot/#_create_an_application_class

https://www.baeldung.com/spring-mvc-form-tutorial
https://learningprogramming.net/category/java/spring-boot/
https://mkyong.com/tutorials/spring-boot-tutorials/

https://livebook.manning.com/book/spring-boot-in-action/chapter-1/
https://dzone.com/articles/beginner%E2%80%99s-guide-jpa-and

LOGIN-REGISTRATION

https://www.codejava.net/frameworks/spring-boot/user-registration-and-login-tutorial - Code Java (on YouTube as well:)

https://youtu.be/TRhQjr3-0wk?feature=shared

https://www.jackrutorial.com/2018/04/spring-boot-user-registration-login.html

[?] https://hellokoding.com/registration-and-login-example-with-spring-security-spring-boot-spring-data-jpa-hsql-jsp/

CRUD + Angular [?]

https://www.youtube.com/playlist?list=PLHMsczabAQ7TSOwhL9SdLGKqgi6vY9l1G

DISPLAYING DATA FROM DATABASE TABLES IN HTML TABLE

JSTL Core tag <c:forEach> is used

```
<c:forEach var="product" items="${products }">
	<tr>
		<td>${product.id }</td>
		<td>${product.name} </td>
	</tr>
</c:forEach>
```

https://learningprogramming.net/java/spring-mvc/read-data-from-database-with-spring-data-jpa-in-spring-mvc/

https://youtu.be/rg4JSM8podw?feature=shared – Dynamic HTML Table

Thymeleaf: https://youtu.be/2RJ0kLZxV60?feature=shared

Similar Projects

- https://github.com/PuneethReddyHC/event-management
- https://github.com/vishal1605/Movie-seat-project
  https://github.com/vishal1605/Movie-seat-project/blob/main/src/main/resources/templates/dashboard.html >> Seats Linking
- https://youtu.be/70uBNBJNR2Q?feature=shared
- Spring + Angular: Movie Ticket Booking
  https://github.com/thejesh812/Movie-Ticket-Booking-Application-Using-Spring-Boot-And-Angular
  https://youtu.be/Dav4QtBi6aU?feature=shared
- BookMyShow: System Design
  https://youtu.be/s4mO1Ak9G84?feature=shared
  https://youtu.be/lBAwJgoO3Ek?feature=shared

Hibernate annotations:

- https://access.redhat.com/documentation/en-us/red_hat_jboss_web_server/1.0/pdf/hibernate_annotations_reference_guide/red_hat_jboss_web_server-1.0-hibernate_annotations_reference_guide-en-us.pdf

Resource Dump

Java Serialization: Object Serialization-deserialization

https://stackoverflow.com/questions/239280/which-is-the-best-alternative-for-java-serialization

https://stackoverflow.com/questions/4548816/when-should-we-implement-serializable-interface

https://youtu.be/smLsR7dZlfU?feature=shared

Edureka >> https://youtu.be/6B6vp0jZnb0?feature=shared
https://youtu.be/5dnJusuGbIY?feature=shared @Serialization annotation

```
import java.io.Serializable;
public class Event implements Serializable
```

>> Put 'implements Serializable' in class definition of Event

https://www.baeldung.com/jpa-entities-serializable

Formats: JSON & Protobuf

Spring Security – for webpage authorization:

Spring Security Authorization
https://youtu.be/wjtLUKbPKXI?feature=shared
https://youtu.be/1ERV-6cz2xk?feature=shared

Seat Booking Page:

JavaScript
https://youtu.be/kEa5KO7jPx4?feature=shared
https://youtu.be/MJOnckN_0D0?feature=shared
https://youtu.be/Kb856Ts8Bws?feature=shared

Email Configuration >> mail shoot on booking

https://youtu.be/ugIUObNHZdo?feature=shared

>> Export Data to PDF (Print Ticket): https://youtu.be/yu5sQBkgoRk?si=IGmT8mvIp1vpQHAt ,
https://youtu.be/S7udzd3xjGQ?si=0EkEEmzZbrpuYSi6

>> Spring MVC @PathVariable with dot (.) is getting truncated

---

ROUGH:

- @RestController - @ResponseBody // @PutMapping - @RequestBody
- Add Admin registration
- EM Bookings Page

// DB SAVE VALIDATION
- Form Submission...Errors HAndling - Already registered

- Admin Login + registration- Logged in User Page
$ Login Access: Flag or status-field:0/1

------remove form action: ajax submission

-Booking Success Page
??? x redirect x simple display
        static variable bookingID b/w controllers
        /alt/ result from AJAX function added to href for deetailsLink.
                "/bookSeats-{bid}" Mapping :: Security Issue

        // @SessionAtrributes("savedBooking")

??? Serialization // JSON-Jackson
??? CAscadeType >> Update Foreign Key

You can try something like below to get available seats from DB based on data selection.

```
$("#datepicker").datepicker({

   onSelect: function(date, instance) {

      $.ajax
      ({
          type: "Post",
          url: "/getAvaiableSeat",
          data: "date="+date,
          success: function(result)
          {
             //handle results and show available seats based on the data you got from table
          }
       });
   }
```

});
}

-jar/war...webapp foldername
-native Query=true, @Repository

- @PutMapping, @RequestBody...form action post/get not supported
- There was an unexpected error (type=Unsupported Media Type, status=415).
Content type 'application/x-www-form-urlencoded' not supported
org.springframework.web.HttpMediaTypeNotSupportedException: Content type 'application/x-www-form-urlencoded' not supported

- repository package new/models?

- @DateTimeFormat..pattern? what is passed to backend?

-ErrorMessages Page
-Registartion Success Page

Documentation "redirect:/pagename" >> redirectAtrributes...regular attributes dont survive
---

```
<input class="form-control" type="email"  required="" placeholder="username"
 oninvalid="this.setCustomValidity('Please Enter valid email')"
 oninput="setCustomValidity('')"></input>
```

Task Details:

=> ADD ADMIN REGISTRATION

. BookEvent ->

```
1.BookEventController ->@Autowired BookEventService bookEventService;
@getMapping("\getEventDetail")
GetEventDetail(@Requestparam string xyx)
{
 EventDetails eventDetails = bookEventService.getallEventDetail();
 Return eventDetails;
}
```

2. BookEventServiceImpl->

```
@Autowired BookEventRepo repo;
public EventDetails getallEventDetail(){
```

```
EventDetails eventDetails = new EventDetails();
eventDetails = repo.findAll();
 return eventDetails;
}
```

3. BookEventRepo.java

4. EventDetails- > events

lists of events ->

user can select future events -
he will click on the select button - you will redirect on seat selection.
user submitted after seat selection (Seat number , price , event details)
BookEventController- > save all these details into your event detail/Seat table table.