Conference Planner

This service provides functionalities for planning and retrieving conference schedules.

Endpoints:

* POST /v1/conferences/planning: Endpoint for planning a conference schedule. Path is given below;
* <http://localhost:8081/conference-app/api/v1/conferences/planning>
* GET /v1/conferences: Endpoint for retrieving conference schedules. Path is given below;

- <http://localhost:8081/conference-app/api/v1/conferences>

1. Plan Conference Endpoint (POST /v1/conferences/planning):

* Request Payload: List of ConferencePlannerCreatePayload objects representing presentations to be scheduled.
* Request Body Example:

[

{

"title": "Architecting Your Codebase",

"duration": 60

},

{

"title": "Overdoing it in Python",

"duration": 45

},

{

"title": "Flavors of Concurrency in Java",

"duration": 30

},

{

"title": "Ruby Errors from Mismatched Gem Versions",

"duration": 45

},

{

"title": "JUnit 5 - Shaping the Future of Testing on the JVM",

"duration": 45

},

{

"title": "Cloud Native Java",

"duration": 5

},

{

"title": "Communicating Over Distance",

"duration": 60

},

{

"title": "AWS Technical Essentials",

"duration": 45

},

{

"title": "Continuous Delivery",

"duration": 30

},

{

"title": "Monitoring Reactive Applications",

"duration": 30

},

{

"title": "Pair Programming vs Noise",

"duration": 45

},

{

"title": "Rails Magic",

"duration": 60

},

{

"title": "Microservices 'Just Right'",

"duration": 60

},

{

"title": "Clojure Ate Scala (on my project)",

"duration": 45

},

{

"title": "Perfect Scalability",

"duration": 30

},

{

"title": "Apache Spark",

"duration": 30

},

{

"title": "Async Testing on JVM",

"duration": 60

},

{

"title": "A World Without HackerNews",

"duration": 30

},

{

"title": "User Interface CSS in Apps",

"duration": 30

},

{

"title": "Apache Kafka",

"duration": 48

},

{

"title": "LIGHTNING TALK",

"duration": 4

}

]

* Response: Responds with a success message upon successful planning of the conference.
* Response Example:

{

"messageCode": "SUCCESS"

}

2. Get Conference Planning Endpoint (GET /v1/conferences):

* Response: Retrieves the planned conference schedules.
* Response Example:

{

"messageCode": 1000,

"body": {

"Track 2": [

" 09:00AM Ruby Errors from Mismatched Gem Versions 45min",

" 09:45AM JUnit 5 - Shaping the Future of Testing on the JVM 45min",

" 10:30AM AWS Technical Essentials 45min",

" 11:15AM Pair Programming vs Noise 45min",

" 12:00PM Lunch 60min",

" 13:00PM Clojure Ate Scala (on my project) 45min",

" 13:45PM Flavors of Concurrency in Java 30min",

" 14:15PM Continuous Delivery 30min",

" 14:45PM Monitoring Reactive Applications 30min",

" 15:15PM Perfect Scalability 30min",

" 15:45PM Apache Spark 30min",

" 16:15PM A World Without HackerNews 30min",

" 16:45PM Networking Event 15min"

],

"Track 1": [

" 09:00AM Architecting Your Codebase 60min",

" 10:00AM Communicating Over Distance 60min",

" 11:00AM Rails Magic 60min",

" 12:00PM Lunch 60min",

" 13:00PM Microservices 'Just Right' 60min",

" 14:00PM Async Testing on JVM 60min",

" 15:00PM Apache Kafka 48min",

" 15:48PM Overdoing it in Python 45min",

" 16:33PM Cloud Native Java 5min",

" 16:38PM LIGHTNING TALK 4min",

" 16:42PM Networking Event 18min"

],

"Track 3": [

" 09:00AM User Interface CSS in Apps 30min",

" 12:00PM Lunch 60min",

" 16:00PM Networking Event 60min"

]

}

}

Tech stacks used in this project;

* Java
* Spring Boot
* Spring Data Jpa
* Hibernate
* Open Api
* JUnit5 & Mockito
* H2 Database

Note-1: Swagger has been added to project. Path is given below;

* <http://localhost:8081/conference-app/api/swagger-ui/index.html#/>

Note-2: H2 database can be accessed from the following path;

* <http://localhost:8081/conference-app/api/h2-console>

Note-3: Postman Collection has been added to project under the resource folder.

Note-4: The service handles errors and exceptions such as missing parameters, insufficient time, and presentation not found, and responds with appropriate error messages and HTTP status codes.