



Project Phase II: A Simple RESTful Web Service Development with Spring Boot

Test Report

Atakan Gundeger - 2587673

Mete kabakci – 2587673

/addfestival :

POST http://localhost:8080/addfestival Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "festivalname": "SpringFest",
3   "city": "Nicosia"
4 }
```

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 163 ms Size: 245 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "festivalid": 2,
3   "festivalname": "SpringFest",
4   "city": "Nicosia",
5   "festivalruns": null
6 }
```

1 • SELECT * FROM festivaldb.festival;

Result Grid Filter Rows: Edit:

	festivalid	city	festivalname
▶	1	Nicosia	SpringFest
	2	Nicosia	SpringFest
✱	NULL	NULL	NULL

/addfestivalrun :

POST http://localhost:8080/addfestivalrun Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☐ JSON Beautiful

```
1 {
2   ... "name": "Springfest21",
3   ... "duration": "16",
4   ... "date": "2022-05-13T11:44:39.833+00:00"
5 }
6 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 177 ms Size: 306 B Save Response

Pretty Raw Preview Visualize JSON Q

```
1 {
2   "festivalrunid": 2,
3   "name": "Springfest21",
4   "duration": 16,
5   "date": "2022-05-13T11:44:39.833+00:00",
6   "events": null,
7   "organizers": null,
8   "festival": null
9 }
```

1 • **SELECT * FROM festivaldb.festival_run;**

Result Grid Filter Rows: Edit: Export/Import

	festivalrunid	date	duration	name	festival_festivalid
▶	1	2022-05-13 14:44:40	16	Springfest21	NULL
	2	2022-05-13 14:44:40	16	Springfest21	NULL
*	NULL	NULL	NULL	NULL	NULL

/addconcert :

POST http://localhost:8080/addconcert Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Nafiz Dölek Konseri",
3   "description": "Live Concert",
4   "date": "2022-05-14T11:44:39.833+00:00",
5   "performer": "Nafiz Dölek"
6 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 20 ms Size: 321 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "eventid": 1,
3   "name": "Nafiz Dölek Konseri",
4   "description": "Live Concert",
5   "date": "2022-05-14T11:44:39.833+00:00",
6   "festivalrun": null,
7   "performer": "Nafiz Dölek"
8 }
```

1 • SELECT * FROM festivaldb.concert;

	performer	eventid
▶	Nafiz Dölek	1
*	NULL	NULL

/addshow :

POST http://localhost:8080/addshow Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "name": "Cem Yilmaz event",
3   "description": "Standup",
4   "date": "2022-05-14T11:44:39.833+00:00",
5   "performers": "Cem Yilmaz"
6 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 9 ms Size: 313 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "eventId": 3,
3   "name": "Cem Yilmaz event",
4   "description": "Standup",
5   "date": "2022-05-14T11:44:39.833+00:00",
6   "festivalrun": null,
7   "performers": "Cem Yilmaz"
8 }
```

1 • SELECT * FROM festivaldb.shows;

Result Grid Filter Rows: Edit:

	performers	eventId
▶	Cem Yilmaz	3
*	NULL	NULL

/addotheractivity :

POST http://localhost:8080/addotheractivity

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Strawberry Gathering",
3   "description": "Strawberry gathering from village of Yesilirmak",
4   "date": "2022-05-14T11:44:39.833+08:00"
5 }
```

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 13 ms Size: 330 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "eventId": 4,
3   "name": "Strawberry Gathering",
4   "description": "Strawberry gathering from village of Yesilirmak",
5   "date": "2022-05-14T11:44:39.833+08:00",
6   "festivalrun": null
7 }
```

Result Grid						Filter Rows:	Edit:	Export/Import:	Wrap Cell Content
	eventid	date	description	name	festivalrun				
▶	1	2022-05-14 14:44:40	Live Concert	Nafiz Dölek Konseri	NULL				
	2	2022-05-14 14:44:40	Standup	Cem Yılmaz event	NULL				
	3	2022-05-14 14:44:40	Standup	Cem Yılmaz event	NULL				
	4	2022-05-14 14:44:40	Strawberry gathering from village of Yesilirmak	Strawberry Gathering	NULL				
✱	NULL	NULL	NULL	NULL	NULL				

Result Grid	
▶	eventid
▶	4
✱	NULL

/addorganiser :

POST http://localhost:8080/addorganiser

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "email": "alikoc@metu.edu.tr",
3   "fname": "Ali",
4   "lname": "Koc",
5   "address": "Kadıköy, Ülker Arena",
6   "phone": 2011007,
7   "FestivalRun": []
8 }
9 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "email": "alikoc@metu.edu.tr",
3   "fname": "Ali",
4   "lname": "Koc",
5   "address": "Kadıköy, Ülker Arena",
6   "phone": 2011007,
7   "festivals": null
8 }
```

Status: 200 OK Time: 8 ms Size: 291 B Save Response

Limit to 1000 rows

1 • SELECT * FROM festivaldb.organizer;

Result Grid Filter Rows: Edit: Expo

	email	address	fname	lname	phone
▶	alikoc@metu.edu.tr	Kadıköy, Ülker Arena	Ali	Koc	2011007
	kamil@metu.edu.tr	cokuzaklarr	kamil	can	533
*	NULL	NULL	NULL	NULL	NULL

/getfestival/{festivalid} :

GET http://localhost:8080/getfestival/1

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 [JSON]
2
3 "email": "alikoc@metu.edu.tr",
4 "fname": "Ali",
5 "lname": "Koc",
6 "address": "Kadıköy, Ülker Arena",
7 "phone": 2818887,
8 "FestivalRun": []
9 [JSON]

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 205 ms Size: 243 B Save Response

Pretty Raw Preview Visualize JSON

1 [JSON]
2 {
3 "festivalid": 1,
4 "festivalname": "SpringFest",
5 "city": "Nicosia",
6 "festivalruns": []
7 }
8 [JSON]
9

/getallfestivals :

GET http://localhost:8080/getallfestivals

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 [JSON]
2
3 "email": "alikoc@metu.edu.tr",
4 "fname": "Ali",
5 "lname": "Koc",
6 "address": "Kadıköy, Ülker Arena",
7 "phone": 2818887,
8 "FestivalRun": []
9 [JSON]

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 20 ms Size: 325 B Save Response

Pretty Raw Preview Visualize JSON

1 [JSON]
2 {
3 "festivalid": 1,
4 "festivalname": "SpringFest",
5 "city": "Nicosia",
6 "festivalruns": []
7 },
8 {
9 "festivalid": 2,
10 "festivalname": "SpringFest",
11 "city": "Nicosia",
12 "festivalruns": []
13 }
14 [JSON]

/getfestivalrun/{festivalrunid} :

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/getfestivalrun/1`. The request body is a JSON object with the following fields:

```
{  "email": "alikoc@metu.edu.tr",  "fname": "Ali",  "lname": "Koc",  "address": "Kadıköy, Ülker Arena",  "phone": "2011007",  "FestivalRun": []}
```

The response is a JSON object with the following fields:

```
{  "festivalrunid": 1,  "name": "Springfest21",  "duration": 16,  "date": "2022-05-13T11:44:40.000+00:00",  "events": [],  "organizers": [],  "festival": null}
```

The status is 200 OK, Time: 208 ms, Size: 302 B.

/getallfestivalruns/{festivalid} :

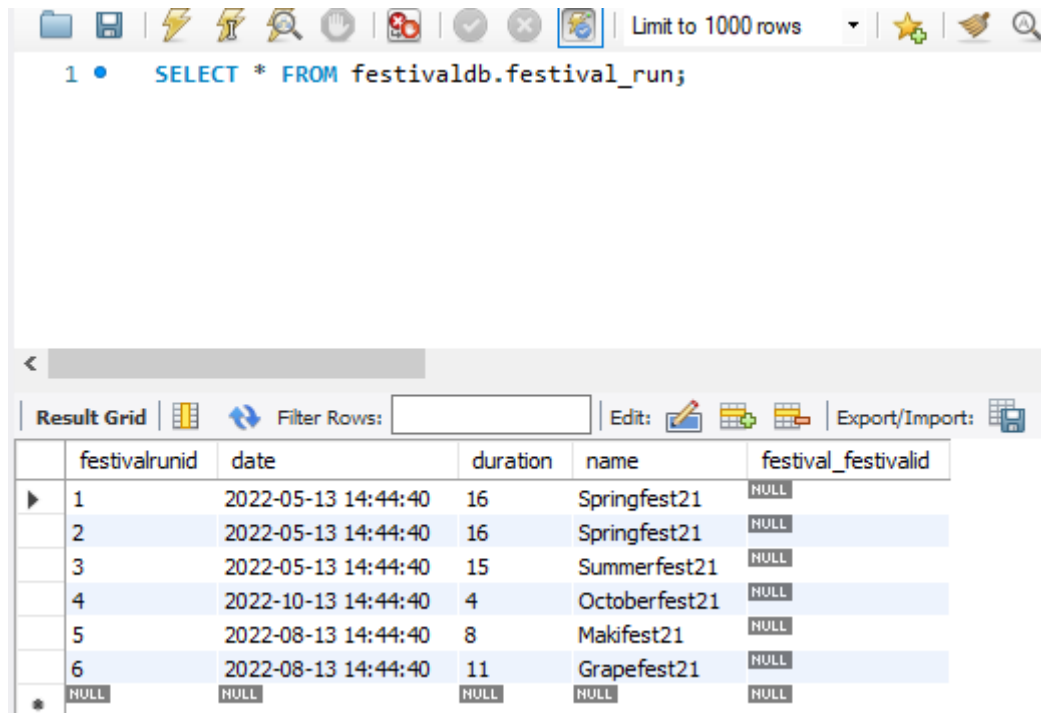
The screenshot shows a REST client interface with a GET request to `http://localhost:8080/getallfestivalruns/1`. The response is an empty JSON array:

```
[]
```

The status is 200 OK, Time: 8 ms, Size: 166 B.

Since we have not add any festivalruns to given festival, empty list is our expected answer.

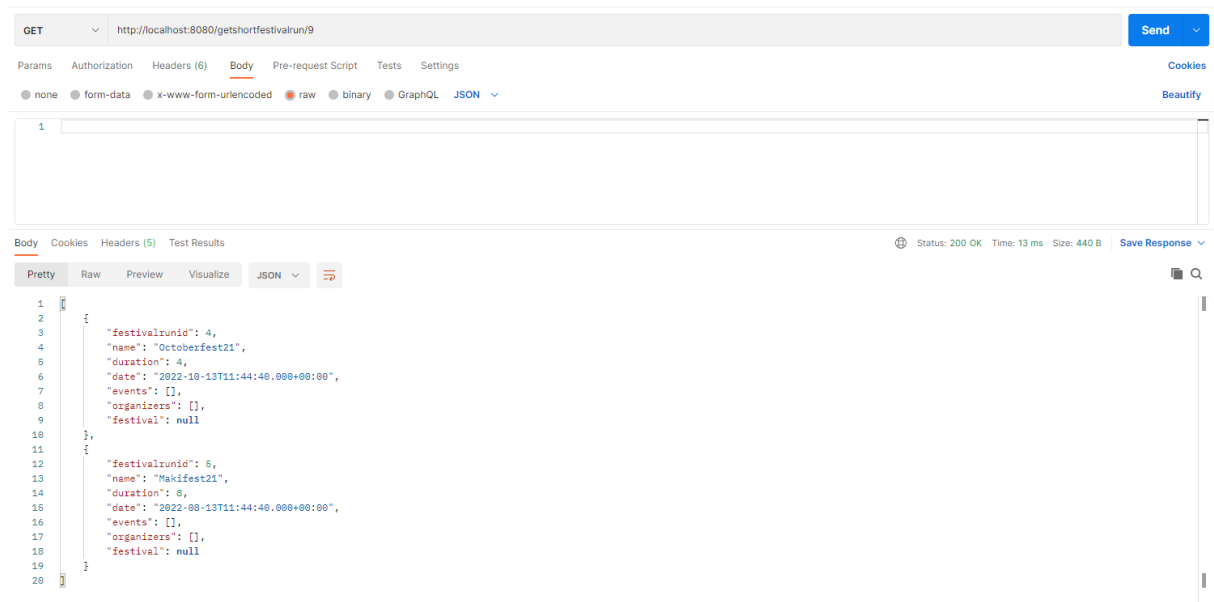
/getshortfestivalrun/{duration} :



The screenshot shows a database client interface. At the top, there's a toolbar with various icons and a text input field containing the SQL query: `1 • SELECT * FROM festivaldb.festival_run;`. Below the query, there's a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and an "Export/Import:" button. The main area displays a table with 6 rows and 6 columns. The columns are: festivalrunid, date, duration, name, and festival_festivalid. The data is as follows:

	festivalrunid	date	duration	name	festival_festivalid
▶	1	2022-05-13 14:44:40	16	Springfest21	NULL
	2	2022-05-13 14:44:40	16	Springfest21	NULL
	3	2022-05-13 14:44:40	15	Summerfest21	NULL
	4	2022-10-13 14:44:40	4	Octoberfest21	NULL
	5	2022-08-13 14:44:40	8	Makifest21	NULL
	6	2022-08-13 14:44:40	11	Grapefest21	NULL
✱	NULL	NULL	NULL	NULL	NULL

This is all our festivalruns, and we tested for duration less than 9.



The screenshot shows a REST client interface. At the top, there's a "GET" method selected and a URL input field containing `http://localhost:8080/getshortfestivalrun/9`. Below the URL, there's a "Send" button. The main area displays the response body, which is a JSON array. The response is as follows:

```
1 [
2   {
3     "festivalrunid": 4,
4     "name": "Octoberfest21",
5     "duration": 4,
6     "date": "2022-10-13T11:44:40.000+00:00",
7     "events": [],
8     "organizers": [],
9     "festival": null
10  },
11  {
12    "festivalrunid": 6,
13    "name": "Makifest21",
14    "duration": 8,
15    "date": "2022-08-13T11:44:40.000+00:00",
16    "events": [],
17    "organizers": [],
18    "festival": null
19  }
20 ]
```

/getfestivalsforacity/{city} :

GET

http://localhost:8080/getfestivalsforacity/Nicosia

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 212 ms

Size: 325 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "festivalid": 1,
3   "festivalname": "SpringFest",
4   "city": "Nicosia",
5   "festivalruns": []
6 },
7 {
8   "festivalid": 2,
9   "festivalname": "SpringFest",
10  "city": "Nicosia",
11  "festivalruns": []
12 }
13
14
```