**Playground - Features**

Feature No. Page

1 – 3

2 – 5

3 – 7

4 – 9

5 – 11

6 – 14

7 – 17

8 – 21

9 – 23

10 – 26

11 – 33

Teammates: Aviv, Liran, Tal, Shay

**Feature 0: system initialization**

Scenario: Test Server Is Booting Correctly

**Given** nothing

**When** the Server starts up

**Then** no error occurs

**Feature 1: Create user**

Scenario 1: successful creation of new player - **PASS**

**Given** the server is up,

**When** I POST /playground/users

And request body is:

{“email”:" demo@gmail ", “username”:”demo”, "avatar”:”avatar.url”, “role”:”player”}

**Then** the response is:

{

UserTO

{

"email”: " demo@gmail ",

"playground":”playground\_lazar”,

“username”:”demo”,

“avatar”:”avatar.url”,

“role”:”player”

"points":"0"

}

}

Scenario 2: create a new User dorm when the user (which means his email) exists in the playground. – **Suceess**

**Given** the server is up,

And database contains:

[

UserEntity

{

"email”: " demo@gmail ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: any valid role,

"points": any valid number >=0

}

]

**When** I POST /playground/users

And request body is: {“email”:" demo@gmail ", “username”:any name, "avatar”:any avatar,

“role”:any valid role}

}

**Then** the response is: status <> 2xx

Scenario 3: create a new user with an invalid role. - **Sucess**

**Given** the server is up,

**When** I POST /playground/users

And request body is: {“email”:" demo@gmail ", “username”:any name, "avatar”:any avatar,

“role”:"servant"}

**Then** the response is: status <> 2xx

**Feature 2: Confirm User**

Scenario 2: player entered invalid confirmation number - **PASSED**

**Given** the server is up,

And database contains:

[

NewUserForm

{

“email”:"demo@gmail.com",

“username”:any name,

"avatar”:any avatar,

“role”:"player"

}

"code":"1234"

]

**When** I GET /playground/users/confirm/playground\_lazar/ demo@gmail.com/12345

**Then** the response is: status <> 2xx

Scenario 3: the database empty (which means user did not created new user form in system) - **PASSED**

**Given** the server is up,

**When** I GET /playground/users/confirm/playground\_lazar/demo@gmail /1234

**Then** the response is: status <> 2xx

**Feature 3: Login**

Scenario 2: user tries to login with invalid playground - **PASS**

**Given** the server is up,

And database contains:

[

UserEntity

{

"email”: " demo@gmail.com",

"playground":”playground\_lazar”,

“username”:”demo”,

“avatar”:”avatar.url”,

“role”:”player”

"points":"0"

}

]

**When** I GET /playground/users/login/playground\_tomy/demo@gmail.com

with headers:

Content-Type: application/json

**Then** the response is: status <> 2xx

Scenario 3: user tries to login with invalid email - **PASS**

**Given** the server is up,

And database contains:

[

UserEntity

{

"email”: " demo@gmail.com",

"playground":”playground\_lazar”,

“username”:”demo”,

“avatar”:”avatar.url”,

“role”:”player”

"points":"0"

}

]

**When** I GET /playground/users/login/playground\_lazar/gogo@gmail.com

with headers:

Content-Type: application/json

**Then** the response is: status <> 2xx

**Feature 4: Update user’ details**

Scenario 2: Update a user when the database is empty - **PASSED**

**Given** the server is up,

**When** I PUT /playground/users/playground\_lazar/demo@gmail.com

with headers:

Accept: application/json

And request body:

{"email”: any email, "playground":any playground,

“username”: any name, “avatar”: any avatar, “role”: any role,

"points":any valid number}

**Then** the response is: status <> 2xx

Scenario 3: Update a user when the user is not exist in databse - **NOT IMPLEMENTED**

**Given** the server is up,

And database contains:

[

UserEntity

{

"email”: " demo@gmail ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: any valid role

"points": any valid number >0

}

]

**When** I PUT /playground/users/playground\_lazar/otherDemo@gmail.com

with headers:

Accept: application/json

And request body:

{"email”: any email, "playground":any playground,

“username”: any name, “avatar”: any avatar, “role”: any role,

"points":any valid number}

**Then** the response is: status <> 2xx

**Feature 5: Add New Element**

Scenario 1: Addition success By Manager - **PASSED**

**Given** the server is up,

And database contain:

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

**When** I POST /playground/elements/playground\_lazar/[demoManager@gmail.com](mailto:demoManager@gmail.com)

with headers:

Accept: application/json

Content-Type: application/json

And body request:

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

**Then** the response is: status 2xx

Scenario 2: Addition failed By Player - **PASSED**

**Given** the server is up,

And database contain:

UserEntity

{

"email”: " demoPlayer@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Player",

"points": any valid number >=0

}

**When** I POST /playground/elements/playground\_lazar/[demoPlayer@gmail.com](mailto:demoPlayer@gmail.com)

with headers:

Accept: application/json

Content-Type: application/json

And body request:

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

**Then** the response is: status <> 2xx

Scenario 3: Add element with mail that does not exist- **PASSED**

**Given** the server is up,

**When** I POST /playground/elements/playground\_lazar/badEmail@gmail.com

with headers:

Accept: application/json:

Content-Type: application/json

And the request body:

Any elementTO

**Then** the response is: status <> 2xx

Scenario 4: Add element when element already exist - **PASSED**

**Given** the server is up,

And the database contains:

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

**When** I POST /playground/elements/playground\_lazar/ demoManager@gmail.com

with headers:

Accept: application/json

Content-Type: application/json

And the request body:

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

**Then** the response is: status <> 2xx

**Feature 6: Update Element**

Scenario 1: Update creator playground success By Manager - **PASSED**

**Given** the server is up,

and the data base contains:

[

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any creator playground

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

]

**When** I PUT /playground/elements/playground\_lazar/demoManager@gmail.com /playground\_tomy/1

with headers:

Accept: application/json

And body request:

"playground":" playground\_lazar",

"id": "1",

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:"lazar\_2019",

“creatorEmail”: "demo@gmail.com"

**Then** the response is: status 2xx

Scenario 2: Update creator playground failed By Player - **PASSED**

**Given** the server is up,

and the data base contains:

[

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any creator playground

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoPlayer@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Player",

"points": any valid number >=0

}

]

**When** I PUT /playground/elements/playground\_lazar/demoPlayer@gmail.com/playground\_tomy/1

with headers:

Accept: application/json

And body request:

"playground":" playground\_lazar",

"id": "1",

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:"lazar\_2019",

“creatorEmail”: "demo@gmail.com"

**Then** the response is: status <> 2xx

Scenario 3: Update element that does not exist - **PASSED**

**Given** the server is up,

**When** I PUT /playground/elements/playground\_lazar/demo@gmail.com/playground\_tomy/1 with:

with headers:

Accept: application/json

And body request:

"playground": any valid playground,

"id": "1",

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

**Then** the response is: status <> 2xx

**Feature 7: Get Specific Element**

Scenario 1: Get specific element success By Player **- PASSED**

**Given** the server is up,

and the data base contains:

[

ElementEntity

{

“playground”:”playground\_lazar”,

“id”:1

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:null,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoPlayer@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Player",

"points": any valid number >=0

}

]

**When** I GET /playground/elements/playground\_lazar/demoPlayer@gmail.com/playground\_lazar/1 with:

with headers:

Content-Type: application/json

**Then** the response is:

{

ElementTO[]

{

“playground”:”playground\_lazar”,

“id”:”1”

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:null,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

}

Scenario 2: Get expired element failed By Player **- PASSED**

**Given** the server is up,

and the data base contains:

[

ElementEntity

{

“playground”:”playground\_lazar”,

“id”:1

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”: any expired date before the test,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoPlayer@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Player",

"points": any valid number >=0

}

]

**When** I GET /playground/elements/playground\_lazar/demoPlayer@gmail.com/playground\_lazar/1 with:

with headers:

Content-Type: application/json

**Then** the response is: status <> 2xx

Scenario 3: Get expired element Successfully By Manager **- PASSED**

**Given** the server is up,

and the data base contains:

[

ElementEntity

{

“playground”:”playground\_lazar”,

“id”:1

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”: any expired date before the test,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

]

**When** I GET /playground/elements/playground\_lazar/demoManager@gmail.com/playground\_lazar/1 with:

with headers:

Content-Type: application/json

**Then** the response is:

{

ElementTO[]

{

“playground”:”playground\_lazar”,

“id”:”1”

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”: any expired date before the test,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

}

Scenario 4: Get an element when the database is empty - **PASSED**

**Given** the server is up,

and the data base contains:

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

**When** I GET /playground/elements/playground\_lazar/ demoManager@gmail.com/lazar/1 with:

with headers:

Content-Type: application/json

**Then** the response is: status <> 2xx

**Feature 8: Get all user' elements**

Scenario 1: get all elements success - **PASSED**

**Given** the server is up,

and the data base contains:

[

ElementEntity[]

{

“playground”:”playground\_lazar”,

“id”:”1”

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

]

**When** I GET /playground/elements/playground\_lazar/ demoManager@gmail.com/all with:

with headers:

Content-Type: application/json

**Then** the response is:

{

ElementTO[]

{

“playground”:”playground\_lazar”,

“id”:”1”

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

}

Scenario 2: database is empty - **PASSED**

**Given** the server is up,

And database contains:

{

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

}

**When** I GET /playground/elements/playground\_lazar/ demoManager@gmail.com /all with:

with headers:

Content-Type: application/json

**Then** the response is: an empty array of ElementTO

Scenario 3: Player get successfully 10 not-expired elements from 20 (include pagination) - **PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

19 element which not expired,

1 element expired

UserEntity

{

"email”: " demoPlayer@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Player",

"points": any valid number >=0

}

**When** I GET /playground/elements/playground\_lazar/demoPlayer@gmail.com/all

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array of size 10, which are not expired

**Feature 9: Get all elements in near to point**

Scenario 1: get the near elements success - **PASSED**

**Given** the server is up,

and the data base contains:

[

ElementEntity[]

{

“playground”:”playground\_lazar”,

“id”: a valid number >0 [= id\_result]

location:

“x”:”0”,

“y”:”1”

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

]

**When** I GET /playground/elements/playground\_lazar/demoManager@gmail.com/near/1.0/1.0/1.0 with:

with headers:

Content-Type: application/json

**Then** the response is: an array of ElementTO which its length is 1, and his “playground” is ”playground\_lazar” and "id" is id\_result.

Scenario 2: user put invalid distance - **PASSED**

**Given** the server is up,

and the data base contains:

an ElementEntity array which its size >=0

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/near/1.0/1.0/-1.0 with:

with headers:

Content-Type: application/json

**Then** the response is: status <> 2xx

Scenario 3: there is no element near user - **PASSED**

**Given** the server is up,

and the data base contains:

[

ElementEntity[]

{

“playground”:”playground\_lazar”,

“id”:”1”

location:

“x”:”0”,

“y”:”1”

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

]

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/near/1.0/1.0/0.0 with:

with headers:

Content-Type: application/json

**Then** the response is: an empty array of ElementTO

Scenario 4: getting first near elements (pagination) - **PASSED**

**Given** the server is up,

and the database contains an array of ElementEntity in size >10 in distance <=1 from the location: "x":0, "y":0

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/near/0.0/0.0/1.0 with:

with headers:

Content-Type: application/json

**Then** the response is: an array of ElementTO in size ==10

Scenario 5: getting 10 first near elements which is not expired by Player – **PASSED**

**Given** the server is up,

and the database contains:

An ElementEntity array, which its size is 20:

19 element which not expired,

1 element expired

UserEntity

{

"email”: " demoPlayer@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Player",

"points": any valid number >=0

}

**When** I GET /playground/elements/playground\_lazar/ demoPlayer@gmail.com/near/0.0/0.0/10.0

with headers:

Content-Type: application/json

**Then** the response is an ElementTO array of size 10, which are not expired

**Feature 10: Search Element**

Scenario 1 - Name: Search by attribute' name "name" success - **PASSED**

Given the server is up,

and the data base :

ElementEntity[]

{

“playground”:”playground\_lazar”,

“id”:”1”

location:

“x”:”0”,

“y”:”1”

"name":"demo"

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

When I GET /playground/elements/playground\_lazar/demoManager@gmail.com/search/name/demo

with headers:

Content-Type: application/json

**Then** the response is:

ElementTO[]

{

“playground”:”playground\_lazar”,

“id”:”1”

location:

“x”:”0”,

“y”:”1”

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

Scenario 1 - Type: Search by attribute' name "type" success - **PASSED**

**Given** the server is up,

and the database :

ElementEntity[]

{

“playground”:”playground\_lazar”,

“id”:”1”

location:

“x”:”0”,

“y”:”1”

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: "demo",

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoManager@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/search/type/demo with:

with headers:

Content-Type: application/json

**Then** the response is:

ElementTO[]

{

“playground”:”playground\_lazar”,

“id”:”1”

location:

“x”:”0”,

“y”:”1”

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: "demo",

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

Scenario 1 - Type: Search expired element by attribute' name "type" by Player - failed - **PASSED**

**Given** the server is up,

and the database :

ElementEntity[]

{

“playground”:”playground\_lazar”,

“id”:”1”

location:

“x”:”0”,

“y”:”1”

"name":any name

“creationDate”:any date,

“exirationDate”:any expired date til this test,

“Type”: "demo",

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

UserEntity

{

"email”: " demoPlayer@gmail.com ",

"playground":”playground\_lazar”,

“username”: any name,

“avatar”: any avatar,

“role”: "Manager",

"points": any valid number >=0

}

**When** I GET /playground/elements/playground\_lazar/demoPlayer@gmail.com/search/type/demo

with headers:

Content-Type: application/json

**Then** the response is: status <> 2xx

Scenario 2 - Name: find one element among twenty elements by attribute' name "name" success - **PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

19 element with attribute “name” as ”demo”,

1 element with attribute “name” as ”demo\_target”

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/search/name/demo\_target with:

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array of size 1, and the element' attribute "name" is ”demo\_target”

Scenario 2 - Type: find one element among twenty elements by attribute' name "type" success - **PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

19 element with attribute “type” as ”demo”,

1 element with attribute “type” as ”demo\_target”

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/search/type/demo\_target with:

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array of size 1, and the element' attribute "type" is ”demo\_target”

Scenario 3 - Name: find five elements among ten elements by attribute' name "name" success - **PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

15 element with attribute “name” as ”demo”,

5 element with attribute “name” as ”demo\_target”

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/search/name/demo\_target with:

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array of size 5, and each element' attribute "name" is ”demo\_target”

Scenario 3 - Type: find five elements among ten elements by attribute' name "name" success - **PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

15 element with attribute “name” as ”demo”,

5 element with attribute “name” as ”demo\_target”

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/search/name/demo\_target with:

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array of size 5, and each element' attribute "name" is ”demo\_target”

Scenario 4: Search for element by invalid attribute' name - **PASSED**

**Given** the server is up,

and the data base contains: An ElementEntity array which its size >=1

**When** I GET /playground/elements/playground\_lazar/aviv@gmail.com/search/attack/1

with headers:

Content-Type: application/json

**Then** the response is: status <> 2xx

Scenario 5 - Type: find no element by attribute' name "type" in array with 20 elements - **PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

20 element with attribute “type” as ”demo type”

**When** I GET [/playground/elements/playground\_lazar/aviv@gmail.com/search/type/no](mailto:/playground/elements/playground_lazar/aviv@gmail.com/search/type/no) demo type

with headers:

Content-Type: application/json

**Then** the response is: an empty ElementTO array

Scenario 5 - Name: find no element by attribute' name "name" in array with 20 elements **- PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

20 element with attribute “name” as ”demo name”

**When** I GET [/playground/elements/playground\_lazar/aviv@gmail.com/search/name/no](mailto:/playground/elements/playground_lazar/aviv@gmail.com/search/name/no) demo name

with headers:

Content-Type: application/json

**Then** the response is: an empty ElementTO array

Scenario 6 - Name: check pagination – get 10 first elements by attribute' name "name" in array with 20 elements **- PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

20 element with attribute “name” as ”demo”

**When** I GET [/playground/elements/playground\_lazar/aviv@gmail.com/search/name/demo](mailto:/playground/elements/playground_lazar/aviv@gmail.com/search/name/demo)

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array which its size is 10

Scenario 6 - Type: check pagination – get 10 first elements by attribute' name "type" in array with 20 elements **- PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

20 element with attribute “type” as ”demo type”

**When** I GET [/playground/elements/playground\_lazar/aviv@gmail.com/search/type/demo](mailto:/playground/elements/playground_lazar/aviv@gmail.com/search/type/demo) type

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array which its size is 10

Scenario 7 - Name: check pagination – get 10 first elements by attribute' name "name" in array with 20 elements **- PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

20 element with attribute “name” as ”demo”

**When** I GET [/playground/elements/playground\_lazar/aviv@gmail.com/search/name/demo](mailto:/playground/elements/playground_lazar/aviv@gmail.com/search/name/demo)?size=7&page=1

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array which its size is 7, and their attribute name "id" is 8 til 14

Scenario 7 - Type: check pagination – get 10 first elements by attribute' name "type" in array with 20 elements **- PASSED**

**Given** the server is up,

and the data base contains:

An ElementEntity array, which its size is 20:

20 element with attribute “type” as ”demo type”

**When** I GET [/playground/elements/playground\_lazar/aviv@gmail.com/search/type/demo](mailto:/playground/elements/playground_lazar/aviv@gmail.com/search/type/demo) type?size=7&page=1

with headers:

Content-Type: application/json

**Then** the response is: an ElementTO array which its size is 7, and their attribute name "id" is 8 til 14

**Feature 11: Use Activity**

Scenario1: Echo activity success - **PASSED**

**Given** the server is up,

**When** I POST /playground/activities/playground\_lazar/demo@gmail.com with:

with headers:

Accept: application/json

Content-Type: application/json

And request body is:

Empty ActivityTO

**Then** the response is:  new Object which is cast to ActivityTO, which its "id" is 1

Scenario2: Add post on board success - **PASSED**

**Given** the server is up,

And database contains:

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

**When** I POST /playground/activities/playground\_lazar/demo@gmail.com with:

with headers:

Accept: application/json

Content-Type: application/json

And request body is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type":"BordPost",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"poster":"tal"

"message":"This is a test"

}

**Then** the response is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type":"BordPost",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"poster":"tal"

"message":"This is a test"

}

Scenario 3: Read two messages from board success - **PASSED**

**Given** the server is up,

and database contains:

ActivityEntity

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type":"BordPost",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"poster":"tal"

"message":"This is a test"

}

ActivityEntity

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type":"BordPost",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"poster":"Human"

"message":"Generic Message"

}

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

**When** I POST /playground/activities/playground\_lazar/demo@gmail.com with:

with headers:

Accept: application/json

Content-Type: application/json

And request body is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type":"ReadPost",

"playPlayground": "playground\_lazar",

"playerEmail":null

}

**Then** the response is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type":"BordPost",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"poster":"tal"

"message":"This is a test"

}

ActivityEntity

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type":"BordPost",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"poster":"Human"

"message":"Generic Message"

}

Scenario 4: Post a message with invalid "type" failed - **PASSED**

**Given** the server is up,

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

**When** I POST /playground/activities/playground\_lazar/demo@gmail.com with:

with headers:

Accept: application/json

Content-Type: application/json

And request body is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type": any invalid type,

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"poster":"tal"

"message":"This is a test"

}

**Then** the response is:  status <> 2xx

Scenario 5: Post a message with invalid "attributes" failed - **PASSED**

**Given** the server is up,

And database contains:

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

**When** I POST /playground/activities/playground\_lazar/demo@gmail.com with:

with headers:

Accept: application/json

Content-Type: application/json

And request body is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type": "ReadPost",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes": null

}

**Then** the response is:  status <> 2xx

Scenario 6: Read empty board - **PASSED**

**Given** the server is up,

And database contains:

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

**When** I POST /playground/activities/playground\_lazar/demo@gmail.com with:

with headers:

Accept: application/json

Content-Type: application/json

And request body is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type": "ReadPost",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"attribute1":"tal"

"attribute2":"This is a test"

}

**Then** the response is:  {}

Scenario 7: Cook an omelette success - **PASSED**

**Given** the server is up,

And database contains:

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

**When** I POST /playground/activities/playground\_lazar/demo@gmail.com with:

with headers:

Accept: application/json

Content-Type: application/json

And request body is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type": "CookOmelette",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"eggSize":"medium"

}

**Then** the response is:  status 2xx

Scenario 8: Cook 4 omelettes with 4 different sizes success - **PASSED**

**Given** the server is up,

And database contains:

ElementEntity

{

"playground":"playground\_lazar",

"id": "1"

"location":

"x":any number,

"y":any number

"name":any name

“creationDate”:any date,

“exirationDate”:any date,

“Type”: any type,

“attributes”: any valid map – include null,

“creatorPlayground”:any valid playground,

“creatorEmail”: any valid email address

}

**When** I POST /playground/activities/playground\_lazar/demo@gmail.com with:

with headers:

Accept: application/json

Content-Type: application/json

And request body is:

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type": "CookOmelette",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"eggSize":"small"

}

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type": "CookOmelette",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"eggSize":"medium"

}

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type": "CookOmelette",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"eggSize":"large"

}

ActivityTO

{

"playground":"playground\_lazar",

"elementPlayground":"playground\_lazar",

"elementId":"1",

"type": "CookOmelette",

"playPlayground": "playground\_lazar",

"playerEmail": any string,

"attributes":

"eggSize":"extraLarge"

}

**Then** the response is:  status 2xx