

Test for the User management REST API

For the user management REST API we used Postman to check the functional correctness of the program

Test Users in the database:

Username	Password	User Type
test@test.com	123456	admin
store@test.com	123456	Store manager
division@test.com	123456	Division manager
loss@test.com	123456	Loss prevention manager

● Login

1. Login with a wrong password for test@test.com

Expect: returns message “fail” with status code 200

Result:

A screenshot of the Postman application showing a POST request to `http://localhost:8080/loss_prevention_war/rest/login`. The request body is a JSON object: `{ "email": "test@test.com", "password": "wrongpassword" }`. The response status is `200 OK` with a time of `2.65 s` and size of `159 B`. The response body is `1 fail`, which is highlighted with a red box.

2. Login with an unregistered user unregistered@test.com

Expect: returns message “fail” with status code 200

Result:

A screenshot of the Postman application showing a POST request to `http://localhost:8080/loss_prevention_war/rest/login`. The request body is a JSON object: `{ "email": "unregistered@test.com", "password": "123456" }`. The response status is `200 OK` with a time of `408 ms` and size of `159 B`. The response body is `1 fail`, which is highlighted with a red box.

3. Login with correct incorrect syntax email (sdfunsif.com)(check the input santinization)

Expect: returns message “unknown user” with status code 200

Result:

POST http://localhost:8080/loss_prevention_war/rest/login

Body

```
1 {
2   "email": "sdfunsif.com",
3   "password": "123456"
4 }
```

Status: 200 OK Time: 10 ms Size: 168 B

1 unknown user

4. Login with correct email and password

Expect: returns a JWT token with status code 200

Result:

POST http://localhost:8080/loss_prevention_war/rest/login

Body

```
1 {
2   "email": "test@test.com",
3   "password": "123456"
4 }
```

Status: 200 OK Time: 1189 ms Size: 360 B

1 eyJhbGciOiJIUzUxMiJ9.eyJzdWUiOiJ0ZXN0QHRlc3QuY29tIiwiaXNkIjoieWRTaW4iLCJpYXQ1OjE2MjQ1NzY4NmZmImV4cCI6MTYyNDU3ODY3M300.4n_rQfz1mpsP2snLvKS8NsAIWwH7k_y1FX347hYpAmNbMA703SoSA9sVjgRJ1a_yr-FB7YsyhHYoiu0mcCQ1g

● Register

We’ve logged in 2 types of users (admin and store manager) so that we can test if admin is the only user type that can register users. JWT token is stored on the header.

POST http://localhost:8080/loss_prevention_war/rest/register

Headers (12)

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWUiOiJ0ZXN0QHRlc3QuY29tIiwiaXNkIjoieWRTaW4iLCJpYXQ1OjE2MjQ1NzY4NmZmImV4cCI6MTYyNDU3ODY3M300.4n_rQfz1mpsP2snLvKS8NsAIWwH7k_y1FX347hYpAmNbMA703SoSA9sVjgRJ1a_yr-FB7YsyhHYoiu0mcCQ1g	Admin
<input type="checkbox"/> Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWUiOiJ0ZXN0QHRlc3QuY29tIiwiaXNkIjoieWRTaW4iLCJpYXQ1OjE2MjQ1NzY4NmZmImV4cCI6MTYyNDU3ODY3M300.4n_rQfz1mpsP2snLvKS8NsAIWwH7k_y1FX347hYpAmNbMA703SoSA9sVjgRJ1a_yr-FB7YsyhHYoiu0mcCQ1g	Store Manager

The ticked one is the one that will be used to send a request. For example, if the admin one is ticked, then this request will be sent with admin’s account

1. Register user with a store manager account

Expect: returns status code “401” unauthorized

Result:

POST http://localhost:8080/loss_prevention_war/rest/register

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRIc3Qu...	Admin			
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJkdG9yZUB0ZXN...	Store Manager			

Store manager's account is used to send out this request

POST http://localhost:8080/loss_prevention_war/rest/register

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies Beautify

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

```
1 {
2   ... "email": "newUser@test.com",
3   ... "password": "123456",
4   ... "first_name": "new",
5   ... "last_name": "user",
6   ... "type": "store manager"
7 }
```

Body Cookies Headers (6) Test Results Status: 401 Unauthorized Time: 258 ms Size: 899 B Save Response

Pretty Raw Preview Visualize HTML

```
3
4 <head>
5 <title>HTTP Status 401 - Unauthorized</title>
```

For the following tests, the admin user will be used to send requests

2. Register with an invalid email (not right syntax) notvalidemail.com

Expect: returns message "invalid email" with status code "400"

Result:

POST http://localhost:8080/loss_prevention_war/rest/register

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies Beautify

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

```
1 {
2   ... "email": "notvalidemail.com",
3   ... "password": "123456",
4   ... "first_name": "new",
5   ... "last_name": "user",
6   ... "type": "store manager"
7 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 349 ms Size: 143 B Save Response

Pretty Raw Preview Visualize Text

```
1 invalid email
```

3. Register with a first_name / last_name that does not pass the user input sanitization

Expect: returns "invalid first name" / "invalid last name" with status code "400"

Result:

POST http://localhost:8080/loss_prevention_war/rest/register

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

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

```
1 {
2   ... "email": "newuser@test.com",
3   ... "password": "123456",
4   ... "first_name": "new",
5   ... "last_name": "<fbf>fbf<sfb>32=343",
6   ... "type": "store manager"
7 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 282 ms Size: 147 B Save Response

Pretty Raw Preview Visualize Text 1 invalid last name

POST http://localhost:8080/loss_prevention_war/rest/register

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

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

```
1 {
2   ... "email": "newuser@test.com",
3   ... "password": "123456",
4   ... "first_name": "<script>",
5   ... "last_name": "user",
6   ... "type": "store manager"
7 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 567 ms Size: 148 B Save Response

Pretty Raw Preview Visualize Text 1 invalid first name

4. Register user with other type rather than “admin / store manager / division manager / loss prevention manager

Expect: return message “invalid type” with status code 400

Result:

POST http://localhost:8080/loss_prevention_war/rest/register

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

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

```
1 {
2   ... "email": "newuser@test.com",
3   ... "password": "123456",
4   ... "first_name": "new",
5   ... "last_name": "user",
6   ... "type": "universe manager"
7 }
```

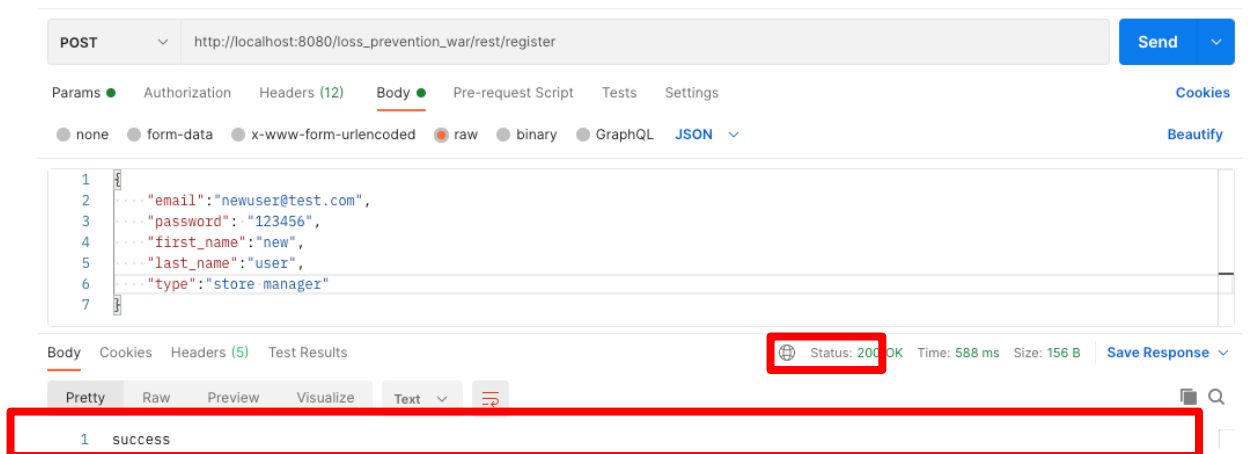
Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 313 ms Size: 142 B Save Response

Pretty Raw Preview Visualize Text 1 invalid type

5. Register with everything correct

Expect: return message “success” with status code 200

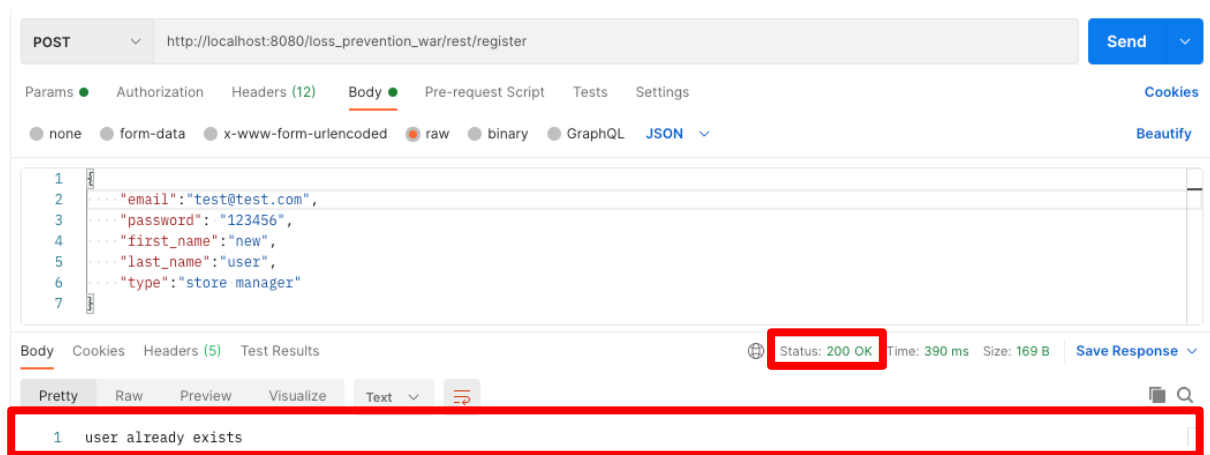
Result:



6. Register with already existing account email

Expect: return message “user already exists” with status code “200”

Result:



• Get/Modify/Delete Account

We’ve logged in 3 different types of users for this sections test. JWT is stored and used the same way as the register section.

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRlc3Qu...	Admin			
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzdG9yZUB0ZXN...	Store Manager			
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJkaXZpc2lvbkB0Z...	Division Manager			

For getting user profile (first name, last name, email, user type), all the users that is logged in is allowed to do that, users that is not logged in will be unauthorized to do this.

For modifying user profile, user itself is only allowed to change basic information about themselves (first name, last name), any other information (user type, password) can only be changed by admin.

Admin has the right to modify first name and last name as well.

For deleting an account, only admin is allowed to do it.

All the test cases below will be based on these assumptions

1. Get a user profile without log in (no Authorization section in the header)

Expect: returns status code “401”

Result:

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRlc3Qu...	Admin			
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzdG9yZUB0ZXN...	Store Manager			
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJkaXZpc2lvbkB0Z...	Division Manager			

No header was selected

GET http://localhost:8080/loss_prevention_war/rest/account/test@test.com

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

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

This request does not have a body

Body Cookies Headers (6) Test Results Status: 401 Unauthorized Time: 454 ms Size: 899 B Save Response

Pretty Raw Preview Visualize HTML

```
4 <head>
5 <title>HTTP Status 401 - Unauthorized</title>
6 <style type="text/css">
```

2. Get existing user profile with a logged in user

Expect: return JSON format about the user which includes first name, last name, email, and type with status code "200"

Result:

GET http://localhost:8080/loss_prevention_war/rest/account/test@test.com

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Authorization Authorization

Body Cookies Headers (5) Test Results Status: 200 OK Time: 251 ms Size: 235 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "last_name": "test",
3   "type": "admin",
4   "first_name": "test",
5   "email": "test@test.com"
6 }
```

3. Get non-existing user profile with a logged in user

Expect: return message "user not found" with status code "404"

Result:

GET http://localhost:8080/loss_prevention_war/rest/account/unkown@test.com

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Authorization Authorization Authorization

Body Cookies Headers (5) Test Results Status: 404 Not Found Time: 153 ms Size: 177 B Save Response

Pretty Raw Preview Visualize JSON

```
1 user not found
```

4. User itself (not admin) trying to modify account including user type and password

Expect: returns status code “401”

Result:

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRlc3QuY29tliwi...	Admin			
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRlc3QuY29tliwi...	Store Manager			
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRlc3QuY29tliwi...	Division Manager			

To show that the correct user is chosen

PUT http://localhost:8080/loss_prevention_war/rest/account/store@test.com

Params Authorization Headers (12) Body Pre-request Script Tests Settings

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

```
1 {
2   ... "password": "changePassword",
3   ... "first_name": "new",
4   ... "last_name": "user",
5   ... "type": "admin"
6 }
```

Body Cookies Headers (6) Test Results Status: 401 Unauthorized Time: 238 ms Size: 899 B Save Response

Pretty Raw Preview Visualize HTML

```
4 <head>
5   <title>HTTP Status 401 - Unauthorized</title>
6   <style tvpe="text/css">
```

5. Another user (neither user itself nor admin) tries to modify other user

Expect: returns status code “401”

Result:

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRlc3QuY29tliwi...	admin@test.com			
<input type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRlc3QuY29tliwi...	store@test.com			
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRlc3QuY29tliwi...	division@test.com			

Shows that the division@test.com is trying to modify store@test.com

PUT http://localhost:8080/loss_prevention_war/rest/account/store@test.com

Params Authorization Headers (12) Body Pre-request Script Tests Settings

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

```
1 {
2   ... "password": "",
3   ... "first_name": "new",
4   ... "last_name": "user",
5   ... "type": ""
6 }
```

Body Cookies Headers (6) Test Results Status: 401 Unauthorized Time: 311 ms Size: 899 B Save Response

Pretty Raw Preview Visualize HTML

```
4 <head>
5   <title>HTTP Status 401 - Unauthorized</title>
6   <style tvpe="text/css">
```

6. User itself tries to modify all the file that is allowed to be modified (first_name, last_name) with inputs that does not pass user input sanitization

Expect: returns message “invalid first name” / “invalid last name” with status code “400”

Result:

PUT `http://localhost:8080/loss_prevention_war/rest/account/store@test.com` Send

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies Beautify

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

```
1 {
2   ... "password": "",
3   ... "first_name": "<script>hello</script>",
4   ... "last_name": "",
5   ... "type": ""
6 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 199 ms Size: 148 B Save Response

Pretty Raw Preview Visualize Text

```
1 invalid first name
```

PUT `http://localhost:8080/loss_prevention_war/rest/account/store@test.com` Send

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies Beautify

Monitors form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "password": "",
3   ... "first_name": "",
4   ... "last_name": "<script>hello</script>",
5   ... "type": ""
6 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 216 ms Size: 147 B Save Response

Pretty Raw Preview Visualize Text

```
1 invalid last name
```

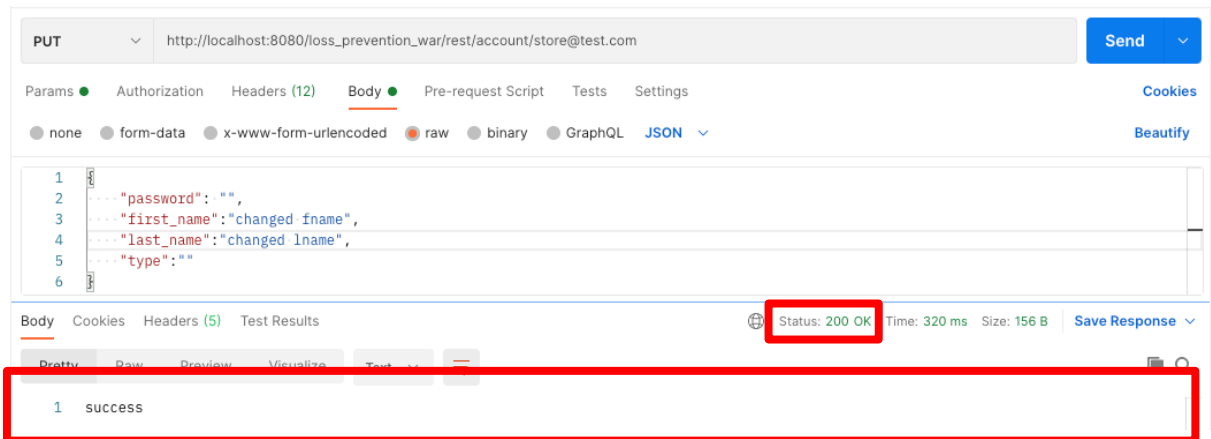
7. User itself tries to modify all the file that is allowed to be modified (first_name, last_name)

Expect: returns message “success” with status code 200

Result:

Before modifying

Edit	Delete	store@test.com	tfWZ8OSdiGBrR2fvWuCa2g==	store	manager	store manager	hgy9l>hF7MxhQ68U
------	--------	----------------	--------------------------	-------	---------	---------------	------------------



After modifying:

Edit	Delete	store@test.com	tfWZ8OSdiGBr2fvWuCa2g==	changed fname	changed lname	store manager	hgy9l>hF7MxhQ68L
------	--------	----------------	-------------------------	---------------	---------------	---------------	------------------

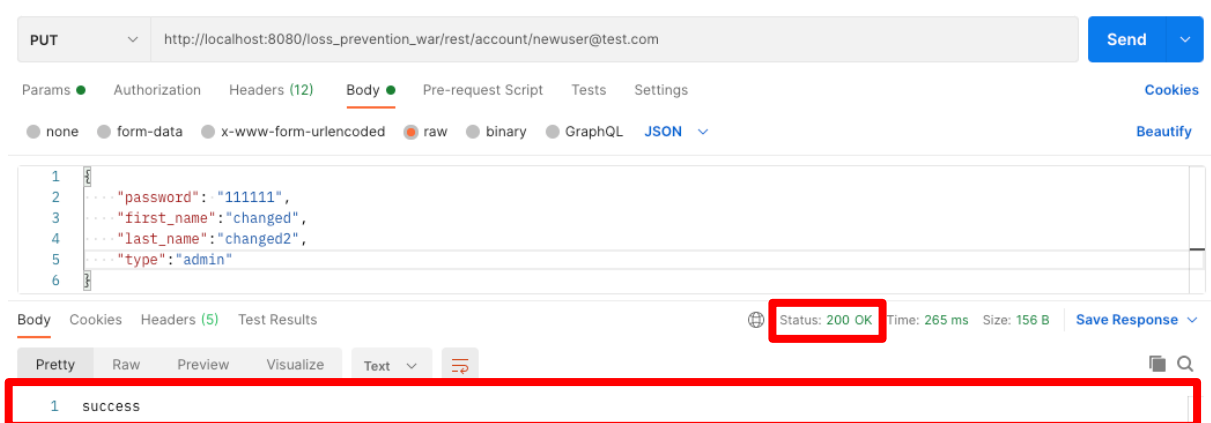
8. Admin modifies the [newuser@test.com](#) (first name, last name, type, password)

Expect: returns message “success” with status code “200”

Result:

Before modifying:

Edit	Delete	newuser@test.com	ZcYHhhEcpewuVJhErD7K5Q==	new	user	store manager	YfbzG8ilM>)Uc(ge
------	--------	------------------	--------------------------	-----	------	---------------	------------------



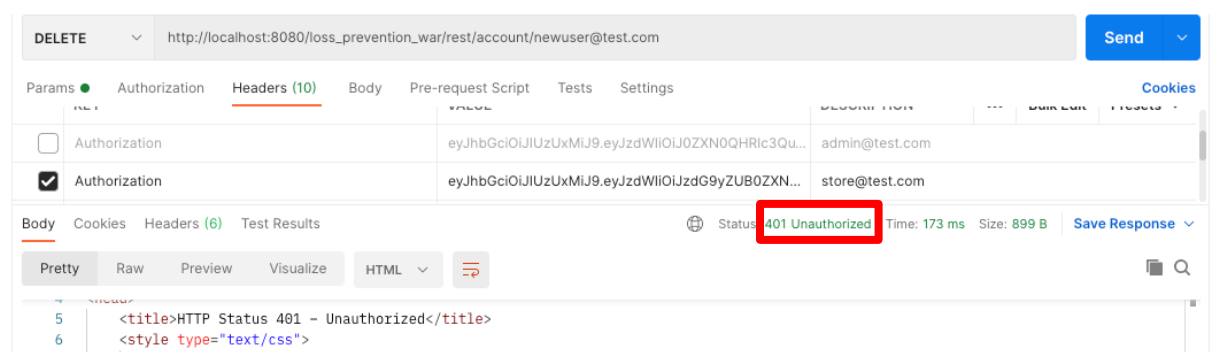
After modifying:

Edit	Delete	newuser@test.com	eKUMm5zRlrf4Yi93UICGIA==	changed	changed2	admin	YfbzG8ilM>)Uc(ge
------	--------	------------------	--------------------------	---------	----------	-------	------------------

9. Not admin tries to delete the [newuser@test.com](#)

Expect: returns status code “401”

Result:



10. Admin tried to delete an account

Expect: returns message “success” with status code “200”

Result:

Before deleting:

Actions	email	hashed_pass	first_name	last_name	type	salt
Edit Delete	newuser@test.com	eKUMm5zRlrf4Yi93UICGIA==	changed	changed2	admin	YfbzG8ilm>)Uc(ge
Edit Delete	division@test.com	VoudgseG725HprAcF9HiXQ==	division	manager	division manager	cv"0*&mmLhC\F_)}
Edit Delete	loss@test.com	/uYtbXUdlGoSedzvXIgvQQ==	loss prevention	manager	loss prevention manager	a#M94%~-.Qy@<LiA
Edit Delete	test@test.com	BIX1dwEKdPB6ECnhGR8rlw==	changed fname	changed lname	admin	?mileC=H"*xi";Zw
Edit Delete	store@test.com	tfWZ8OSdiGBrR2fvWuCa2g==	store	manager	store manager	hgy9[>hF7MxhQ68U

DELETE

http://localhost:8080/loss_prevention_war/rest/account/newuser@test.com

Send

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies

☒ Authorization

eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0QHRic3Qu...

admin@test.com

☐ Authorization

eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0ZUB0ZXN...

store@test.com

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 263 ms

Size: 156 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

success

After deleting:

Actions	email	hashed_pass	first_name	last_name	type	salt
Edit Delete	division@test.com	VoudgseG725HprAcF9HiXQ==	division	manager	division manager	cv"0*&mmLhC\F_)}
Edit Delete	loss@test.com	/uYtbXUdlGoSedzvXIgvQQ==	loss prevention	manager	loss prevention manager	a#M94%~-.Qy@<LiA
Edit Delete	test@test.com	BIX1dwEKdPB6ECnhGR8rlw==	changed fname	changed lname	admin	?mileC=H"*xi";Zw
Edit Delete	store@test.com	tfWZ8OSdiGBrR2fvWuCa2g==	store	manager	store manager	hgy9[>hF7MxhQ68U