

1. Test

Check if docker endpoint is working, you should see

Request				
HTTP Method	GET		Explanation	Note
Resource	http://localhost:4000/testhealth		Test endpoint health state	Happy Test
Body Payload	None			
Response				
Status	Body	Encoding	Explanation	
200	{ "message": "test route" }	JSON	Success.	

2. Create Account -Happy Test

Request				
HTTP Method	POST		Explanation	Note
Resource	http://localhost:4000/createuser			
Body Payload	{ "username": "Jenny@gmail.com", "password": "Aa12345678" }		Create an account	Happy Test
Response				
Status	Body	Encoding	Explanation	Note
201	[{ "message": "user created" }, { "success": true }, { "account_created": "Mon, 05 Aug 2024 17:47:58 GMT", "password": "\$2b\$12\$6C2dkMj3iZ92Aw9ivGgplOqmlm9UBHV7m1AHINccB9lCzluoAKUrC", "username": "jenny@gmail.com" }	JSON	Successfully create an account	Response contains password is for verification purpose, DO NOT use in production.

]			
--	---	--	--	--

3. Login an Account -Happy Test

Request				
HTTP Method	POST		Explanation	Note
Resource	http://localhost:4000/v1/login			
Body Payload	{ "username": "Jenny@gmail.com", "password": "Aa12345678" }		Login an account	Happy Test
Response				
Status	Body	Encoding	Explanation	Note
200	[{ "reason": "Login successful" }, { "success": true }]	JSON	Successfully login an account	

4. Create Account -Negative Test

Request				
HTTP Method	POST		Explanation	Note
Resource	http://localhost:4000/createuser			
Body Payload	{ "username": "Jenny@gmail.com", "password": "Aa12345678" }		Create an account	NegativeTest
Response				
Status	Body	Encoding	Explanation	Note

400	[{ "reason": "The username already exist. please choose a different one." }, { "success": false }]	JSON	Bad Request. Success: False. Reason: duplicated username.	
-----	------------------------------------------------------------------------------------------------------------------------------	------	--------------------------------------------------------------------	--

Request				
HTTP Method	POST		Explanation	Note
Resource	http://localhost:4000/createuser			
Body Payload	{ "username": "jo", "password": "1234" }		Create an account	NegativeTest
Response				
Status	Body	Encoding	Explanation	Note
400	[{ "reason": "username must have a minimum length of 3 characters and a maximum length of 32 characters" }, { "success": false }]	JSON	Bad Request. Success: False. Reason: minimum length required in username.	

Request				
HTTP Method	POST		Explanation	Note
Resource	http://localhost:4000/createuser			
Body Payload	{ "username2": "jo", "password": "1234" }		Create an account	NegativeTest
Response				

Status	Body	Encoding	Explanation	Note
400	[{ "reason": "Missing 'username' key or Missing 'password' key in JSON data" }, { "success": false }]	JSON	Bad Request. Success: False. Reason: No key in payload request.	

5. Login an Account - Negative Test

Request				
HTTP Method	POST		Explanation	Note
Resource	http://localhost:4000/v1/login			
Body Payload	{ "username": "jenny@gmail.com", "password": "Aa" }		Login an account	NegativeTest
Response				
Status	Body	Encoding	Explanation	Note
401	[{ "reason": "Invalid password" }, { "success": false }]	JSON	Unauthorized. Success: False. Reason: Not valid password.	

Request				
HTTP Method	POST		Explanation	Note
Resource	http://localhost:4000/v1/login			
Body Payload	{ "username": "jenn", "password": "Aa" }		Login an account	NegativeTest

Response				
Status	Body	Encoding	Explanation	Note
401	[{ "reason": "no such username" }, { "success": false }]	JSON	Unauthorized. Success: False. Reason: No such username.	

Request				
HTTP Method	POST		Explanation	Note
Resource	http://localhost:4000/v1/login			
Body Payload	{ "username": "jenny@gmail.com", "password": "Aa" }		Login an account. Click this endpoint 6 times .	NegativeTest
Response				
Status	Body	Encoding	Explanation	Note
429	[{ "reason": "Too many failed login attempts. Please wait a minute before trying again." }, { "success": false }]	JSON	Too Many Requests. Success: False. Reason: Not valid password.	

6. Optional, NOT required. If you want to check all the records in userTable.

Request				
HTTP Method	GET		Explanation	Note
Resource	http://localhost:4000/get/table/users		Get all records in User Table	Happy Test
Body Payload	None			

Response				
Status	Body	Encoding	Explanation	Note
200	[{ "all users": [{ "account_created": "Mon, 05 Aug 2024 21:32:36 GMT", "password": "\$2b\$12\$zDrr4.DaxiZBjqX9O6FgEuTH0le.Xp0X/AooSCJQ3i/9cZW0lbbiW", "username": "jenny@gmail.com" }] }, { "success": true }]	JSON	Success.	Response contains all the records in user table, NOT the best practice. DO NOT use in production.

check all the records in loginAttemptTable.

Request				
HTTP Method	GET		Explanation	Note
Resource	http://localhost:4000/get/table/loginAttemptTable		Get all records in LoginAttempt Table	Happy Test
Body Payload	None			
Response				
Status	Body	Encoding	Explanation	Note

200	<pre>[{ "all attempts": [{ "success_login": true, "timestamp": "Mon, 05 Aug 2024 21:40:37 GMT", "username": "jenny@gmail.com" }, { "success_login": false, "timestamp": "Mon, 05 Aug 2024 21:40:40 GMT", "username": "jenny@gmail.com" }] }, { "success": true }]</pre>	JSON	Success.	Response contains all the records in loginAttempt table, NOT the best practice. DO NOT use in production.
-----	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------	----------	-----------------------------------------------------------------------------------------------------------

7. Optional, NOT required. If ever needed to reset the database table

Request				
HTTP Method	DELETE		Explanation	Note
Resource	http://localhost:4000/delete/table/userTable		Drop UserTable	Optional, not required.
Body Payload	None			
Response				
Status	Body	Encoding	Explanation	Note
200	<pre>{ "message": "User table dropped!" }</pre>	JSON	Success.	This will drop the user table, DO NOT expose to frontend user. DO NOT use in production.

Request				
---------	--	--	--	--

HTTP Method	DELETE		Explanation	Note
Resource	http://localhost:4000/delete/table/loginAttemptTable		Drop LoginAttemptTable	Optional, not required.
Body Payload	None			
Response				
Status	Body	Encoding	Explanation	Note
200	{ "message": "LoginAttempt table dropped!" }	JSON	Success.	This will drop the loginAttempt table, DO NOT expose to frontend user. DO NOT use in production.