

1st Assignment hy452

AWS SERVICES

Irodotos Apostolou
csd4437

A1: Step Function

1: First the table is empty

The screenshot shows the AWS DynamoDB console interface. On the left is a navigation sidebar with options like Dashboard, Tables, Update settings, Explore items, PartIQ editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main area is titled 'tests' and shows a 'Tables (2)' dropdown menu. Below it, there's a search bar 'Find tables by table name' and a list of tables with 'tests' selected. The 'tests' table is shown as empty with 'Items returned (0)'. A message at the bottom states 'The query did not return any results.' The top of the browser shows the URL 'us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#item-explorer?initialTagKey=&maximize=true&table=tests'.

2: I am gona run the step function to add in table

The screenshot shows the AWS Step Functions console. The top section displays 'Execution Status' as 'Succeeded' with an 'Execution ARN' of 'arn:aws:states-us-east-1:535490399091:execution:MyStateMachine:b40cf1a5-f8c4-59a7-900b-04940516da49'. It also shows 'State transitions' as 7. The 'Graph view' shows a state machine diagram with a 'Start' node, an 'Action' node, a 'Validation URL' node, a 'Status Code' node, a 'Generate UUID' node, a 'Store Test' node, a 'Bad Request' node, a 'Get Test' node, an 'Exist' node, an 'Item' node, an 'Empty item' node, and an 'End' node. The 'Action' node is highlighted, and its 'Input and output' are shown. The input is a JSON object with 'request' containing 'key': 'post', 'url': 'https://tfz5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?n=1&b=1', and 'expected_result': '2'. The output is a similar JSON object with 'expected_result': '2'.

The screenshot shows the AWS DynamoDB console interface. On the left is a navigation menu with options like Dashboard, Tables, and DAX. The main area displays the 'tests' table. A 'Scan/Query items' section shows a successful scan of the 'tests' table. Below this, the 'Items returned (1)' section shows a single item with the following details:

id	expected_result	url
48ea5add-bb69-44f9...	2	https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=1

The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 11:53 PM on 18/10/22.

I gave valid url so everything did well and stored in table. Now with random url I will get error and execution finished

The screenshot shows the AWS Step Functions console. At the top, a red banner indicates a 'Fail state executed in step: Bad Request'. Below this, the 'Graph view' shows a workflow diagram with steps: Start, Action, Validation URL, Status Code, Generate UUID, Store Test, Bad Request, Item, Empty Item, and End. The 'Bad Request' step is highlighted in red. To the right, the 'Action' section shows the input and output for the failed step. The input is a JSON object:

```
1 {
2   "request": {
3     "key": "post",
4     "url": "https://tfr5qh5091.execute-api",
5     "expected_result": "2"
6   }
7 }
```

The output is a JSON object:

```
1 {
2   "request": {
```

Below the graph, the 'Events (12)' section shows a list of events. The first event is 'ExecutionStarted' at 18 Oct 2022, 12:01:32.507 (GMT+3).

ID	Type	Step	Resource	Started After	Timestamp
1	ExecutionStarted			0	18 Oct 2022, 12:01:32.507 (GMT+3)

The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 12:01 PM on 18/10/22.

3: All okay I am gona try to get it from table

The screenshot displays the AWS Step Functions console for a state machine execution. The top right shows the execution details, including the ARN and state transitions. The main area is divided into a 'Graph view' and an 'Action' panel. The 'Graph view' shows a flowchart starting with 'Start', followed by 'Validation URL', 'Status Code', 'Generate UUID', 'Store Test', 'Bad Request', 'Item', 'Empty Item', and finally 'End'. The 'Action' panel shows the 'Input and output' for the 'Item' state, with the input being a request object and the output being a response object. The 'Events' section at the bottom shows a list of 13 events.

```
graph TD
    Start([Start]) --> Action[Action]
    Action --> ValidationURL[Validation URL]
    ValidationURL --> StatusCode[Status Code]
    StatusCode --> GenerateUUID[Generate UUID]
    GenerateUUID --> StoreTest[Store Test]
    StoreTest --> BadRequest[Bad Request]
    BadRequest --> Item[Item]
    Item --> EmptyItem[Empty Item]
    EmptyItem --> End([End])
```

Input and output for the 'Item' state:

```
1 {
2   "request": {
3     "key": "get",
4     "id": "48ea5add-bb69-44f9-a773-f61fb1164294"
5   }
6 }
```

```
1 {
2   "request": {
3     "key": "get",
4     "id": "48ea5add-bb69-44f9-a773-f61fb1164294"
5   }
6 }
```

I gave correct id and I get the item back. Now I am gona give random id and I will get empty item

The screenshot displays the AWS Step Functions console for a state machine execution. The top right shows the execution details, including the ARN and state transitions. The main area is divided into a 'Graph view' and an 'Action' panel. The 'Graph view' shows a flowchart starting with 'Start', followed by 'Validation URL', 'Status Code', 'Generate UUID', 'Store Test', 'Bad Request', 'Item', 'Empty Item', and finally 'End'. The 'Action' panel shows the 'Input and output' for the 'Item' state, with the input being a request object and the output being a response object. The 'Events' section at the bottom shows a list of 13 events.

```
graph TD
    Start([Start]) --> Action[Action]
    Action --> ValidationURL[Validation URL]
    ValidationURL --> StatusCode[Status Code]
    StatusCode --> GenerateUUID[Generate UUID]
    GenerateUUID --> StoreTest[Store Test]
    StoreTest --> BadRequest[Bad Request]
    BadRequest --> Item[Item]
    Item --> EmptyItem[Empty Item]
    EmptyItem --> End([End])
```

Input and output for the 'Item' state:

```
1 {
2   "request": {
3     "key": "get",
4     "id": "aa"
5   }
6 }
```

```
1 {
2   "request": {
3     "key": "get",
4     "id": "aa"
5   }
6 }
```

A2: API GETWAY

1: I will run post request from API to store 1 more in table

The screenshot shows the AWS API Gateway console for a resource named `/test`. The left sidebar contains navigation links for Custom Domain Names, VPC Links, API: testAPI, Resources, Stages, Authorizers, Gateway Responses, Models, Resource Policy, Documentation, Dashboard, Settings, Usage Plans, API Keys, Client Certificates, and Settings. The main content area displays details for the `/test` resource, including its path, query strings, headers, stage variables, client certificate, and request body. A test call is shown with a status of 502 and a latency of 429 ms. The response body is `{"message": "Internal server error"}`. The response headers include `"x-amzn-ErrorType": "InternalServerErrorException"`. The logs show the execution log for the request, including the request URI, method, path, headers, and body.

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method.

Path

No path parameters exist for this resource. You can define path parameters by using the syntax **(myPathParam)** in a resource path.

Query Strings

`(test)`

`param1=value1¶m2=value2`

Headers

`(test)`

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. `Accept: application/json.`

Stage Variables

No [stage variables](#) exist for this method.

Client Certificate

No client certificates have been generated.

Request Body

```
1 {
2   "request": {
3     "key": "post",
4     "url": "https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=2",
5     "expected_result": "3"
6   }
7 }
```

Request: /test

Status: 502

Latency: 429 ms

Response Body

```
{
  "message": "Internal server error"
}
```

Response Headers

```
{
  "x-amzn-ErrorType": "InternalServerErrorException"
}
```

Logs

Execution log for request 875574ae-012a-49a5-83ea-953841eb0f8a

```
Tue Oct 18 09:07:32 UTC 2022 : Starting execution for request: 875574ae-012a-49a5-83ea-953841eb0f8a
Tue Oct 18 09:07:32 UTC 2022 : HTTP Method: POST, Resource Path: /test
Tue Oct 18 09:07:32 UTC 2022 : Method request path: {}
Tue Oct 18 09:07:32 UTC 2022 : Method request query string: {}
Tue Oct 18 09:07:32 UTC 2022 : Method request headers: {}
Tue Oct 18 09:07:32 UTC 2022 : Method request body before transformations: {
  "request": {
    "key": "post",
    "url": "https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=2",
    "expected_result": "3"
  }
}
Tue Oct 18 09:07:32 UTC 2022 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-31/functions/arn:aws:lambda:us-east-1:535490399091:function:add_necessary_field_for_stateMachine/invocations
Tue Oct 18 09:07:32 UTC 2022 : Endpoint request headers: (X-Amz-Date=20221018090732Z, x-amzn-apigateway-api-id=rmkvkjoea, Accept=application/json, User-Agent=AmazonAPIGateway_tmnkvkjoea, Host=lambda.us-east-1.amazonaws.com, X-Amz-Content-Sha256=7e6d0be47765feb64d5ed75f52f5ab550621daa8f5db5403f4777ebacde921, X-Amzn-Trace-Id=Root=1-634e6c04-0c7abeb9e55229a5ea9a0be, x-amzn-lambda-integration-tag=875574ae-012a-49a5-83ea-953841eb0f8a, Authorization=*****d1ad26, X-Amz-Source-Arn*****)
```

The screenshot shows the AWS DynamoDB console for a table named `tests`. The left sidebar contains navigation links for Dashboard, Tables, Update settings, Explore Items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main content area displays the `tests` table details, including a scan/query table or index, filters, and a table of items returned.

Tables (2)

Any table tag

Find tables by table name

results

tests

tests

Autopreview View table details

Scan/Query items

Scan/query a table or index

Scan Query tests

Filters

Run Reset

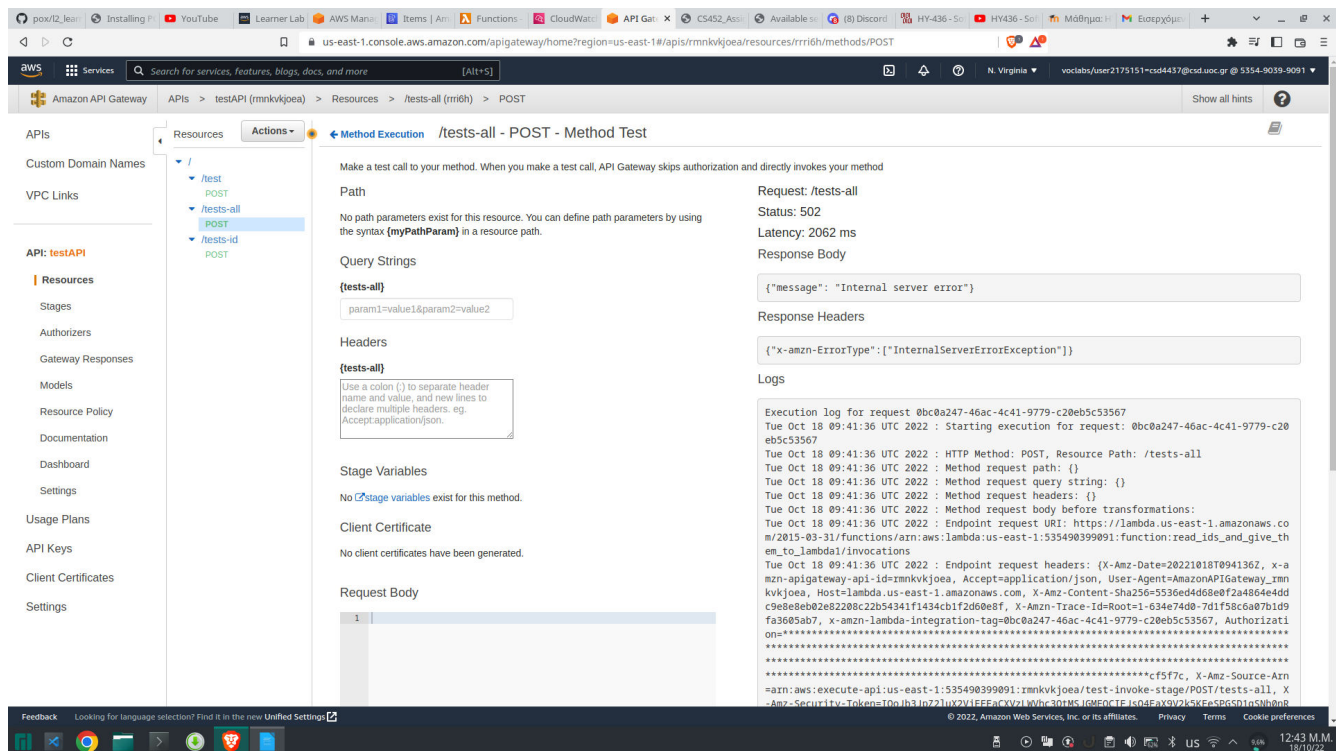
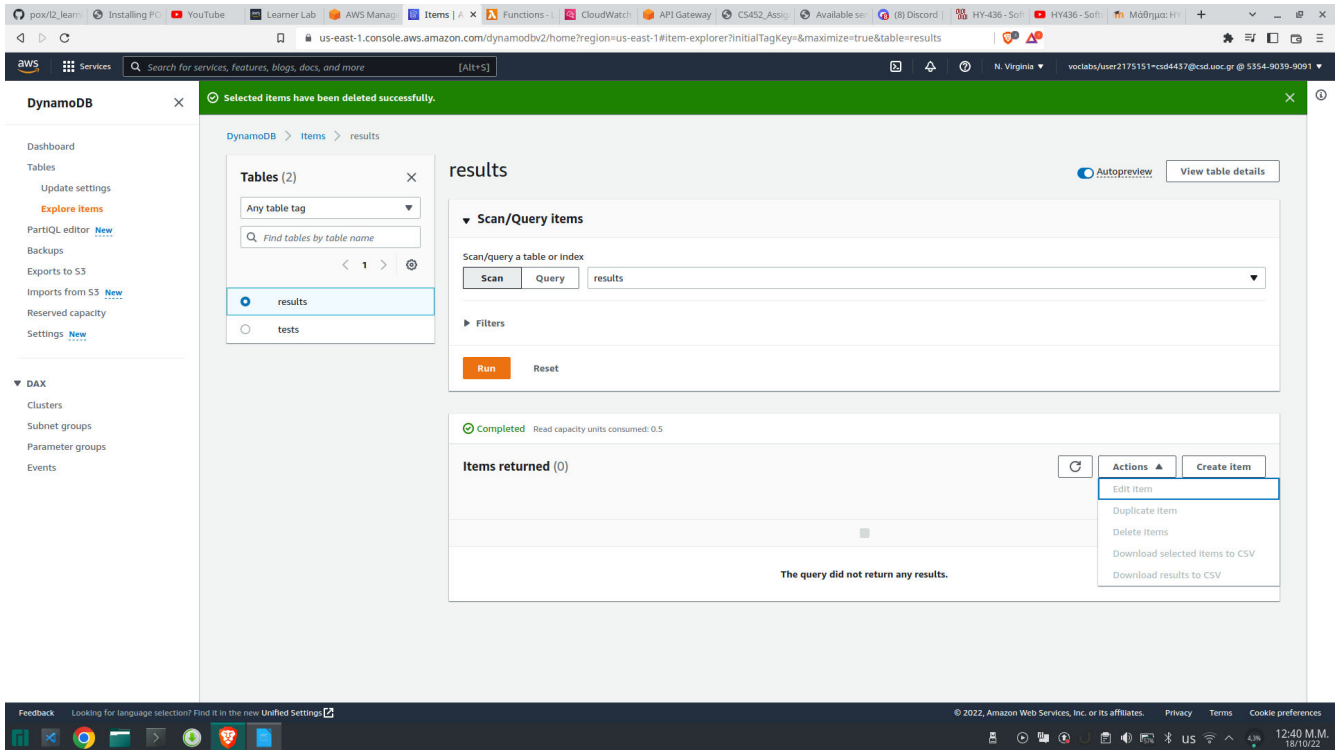
Completed Read capacity units consumed: 0.5

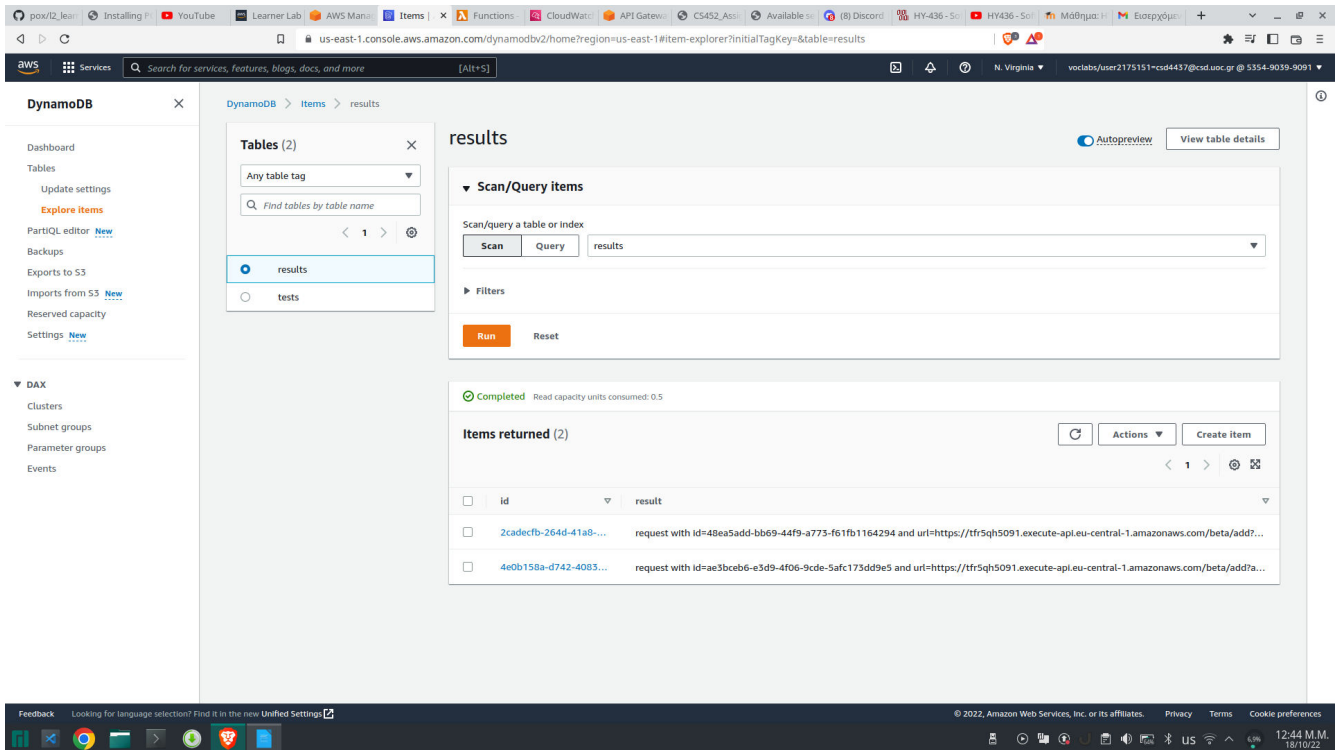
Items returned (2)

id	expected_result	url
48ea5add-bb69-44f9-...	2	https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=1
ae3cbcb6-e3d9-4f06-...	3	https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=2

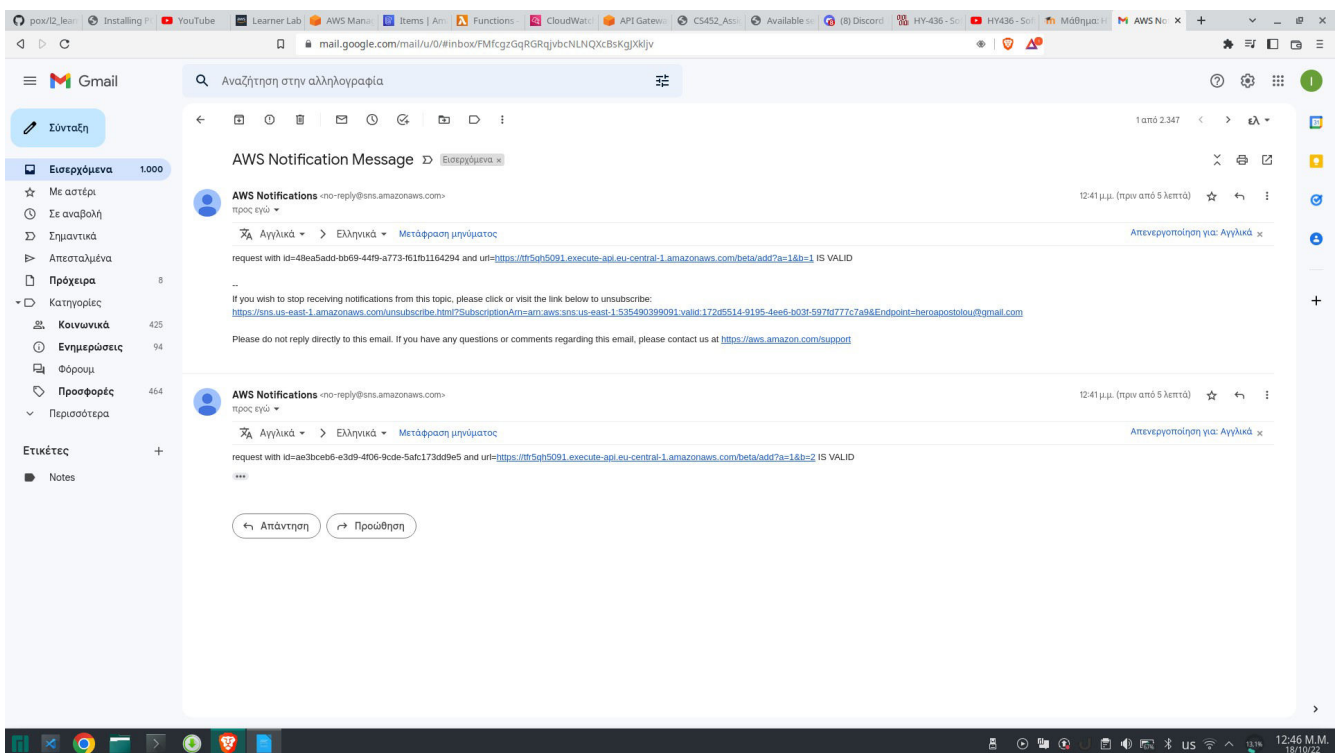
The post request works.

2: Now the post request to get all test from table and store them in result database. First the tests are 2 (we see above) and the result database is empty



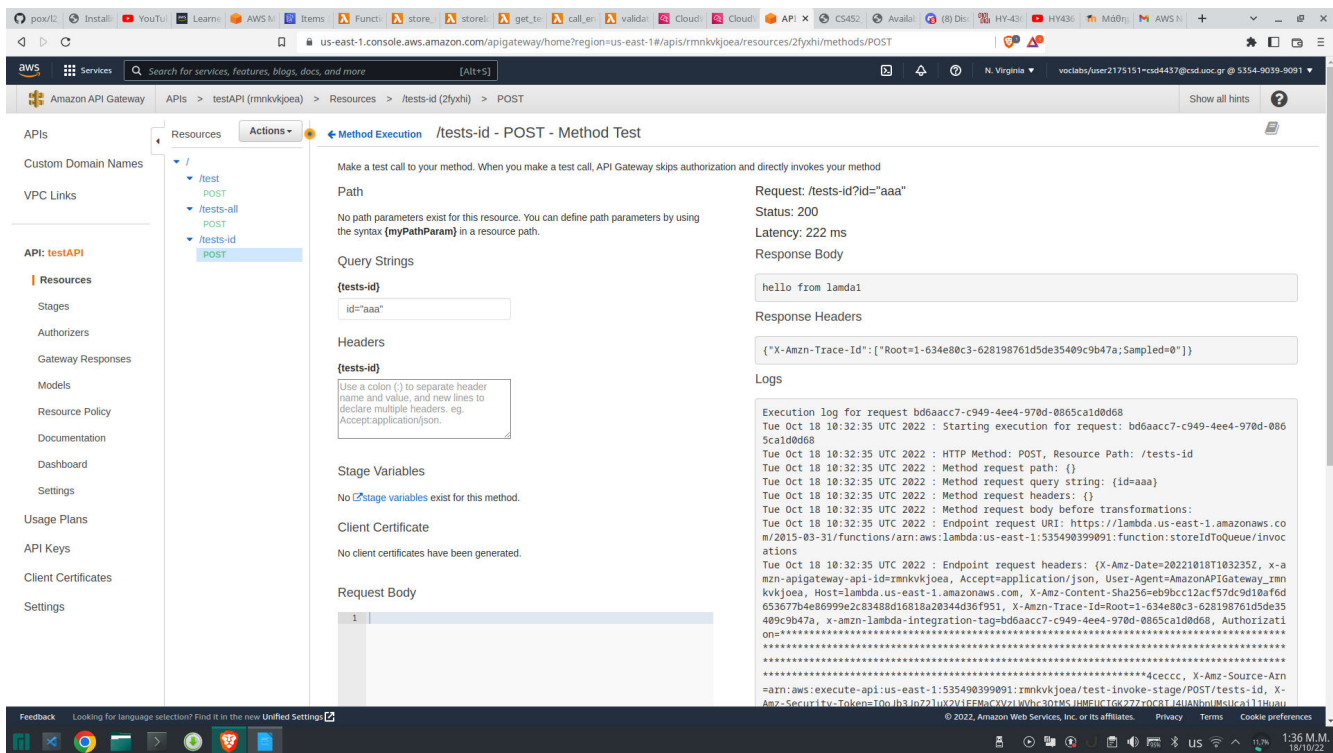


After the tests/all we can see that now we have 2 items in the result table.



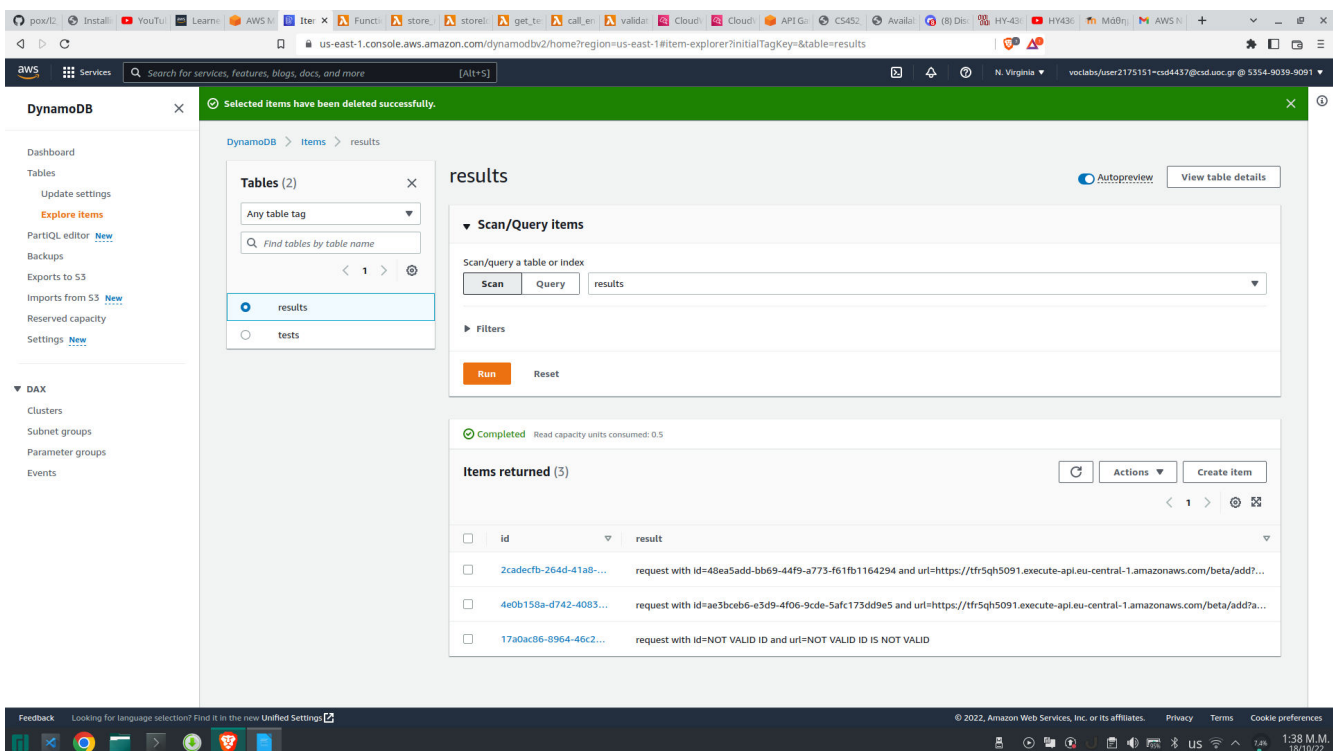
Also I get 2 notification emails from Amazon the result

3) I will post/tests/id BUT THE ID IS NOT EXIST IN TABLE



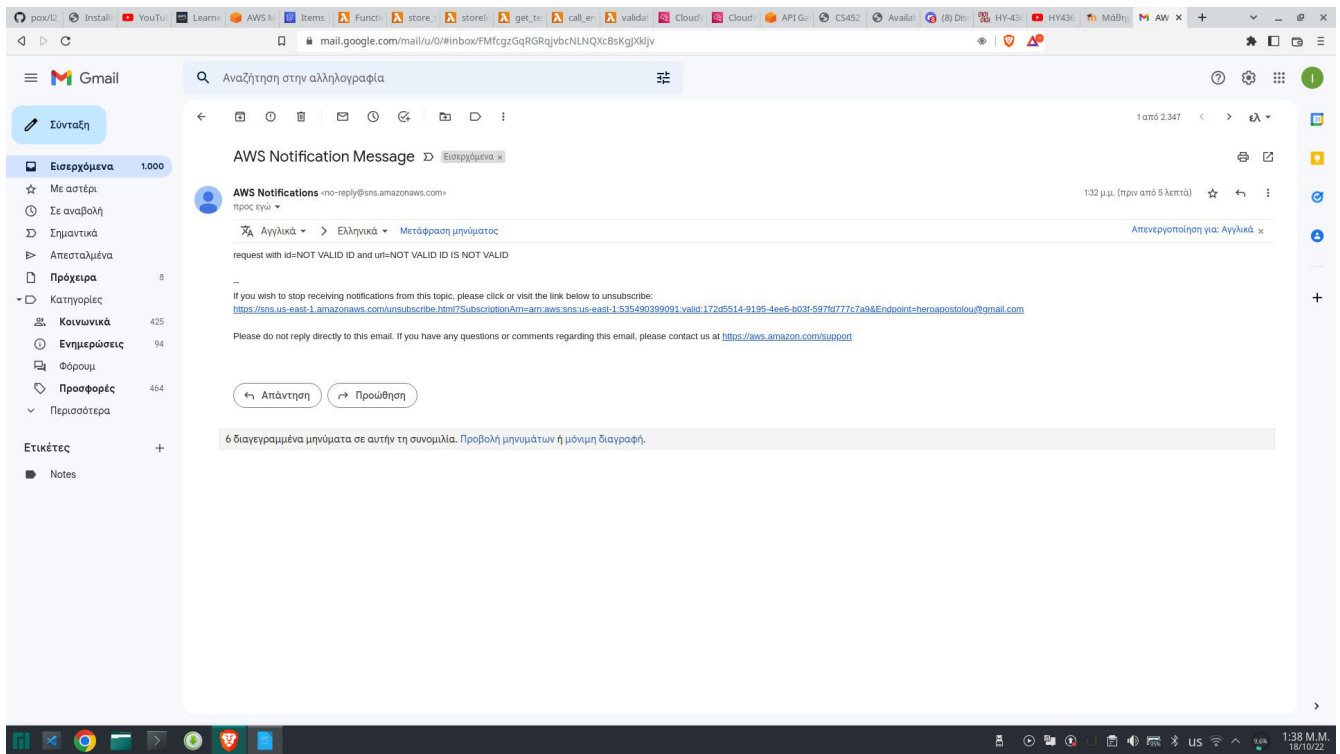
The screenshot shows the Amazon API Gateway console. The left sidebar contains navigation links for APIs, Custom Domain Names, VPC Links, Resources, Stages, Authorizers, Gateway Responses, Models, Resource Policy, Documentation, Dashboard, Settings, Usage Plans, API Keys, Client Certificates, and Settings. The main panel displays the 'Method Execution' for a POST resource named '/tests-id'. The 'Path' field is empty, and the 'Query Strings' field contains '{tests-id}' with a value of 'id=aaa'. The 'Headers' field is empty. The 'Stage Variables' field shows 'No stage variables exist for this method.' The 'Client Certificate' field shows 'No client certificates have been generated.' The 'Request Body' field contains a JSON object: { 'id': 'aaa' }. The 'Response' section shows a status of 200, a latency of 222 ms, and a response body of 'hello from lambda!'. The 'Logs' section shows the execution log for the request, including the request path, headers, and the response body.

Because ID not exist we store in result table this: request with id=NOT VALID ID and url=NOT VALID ID IS NOT VALID

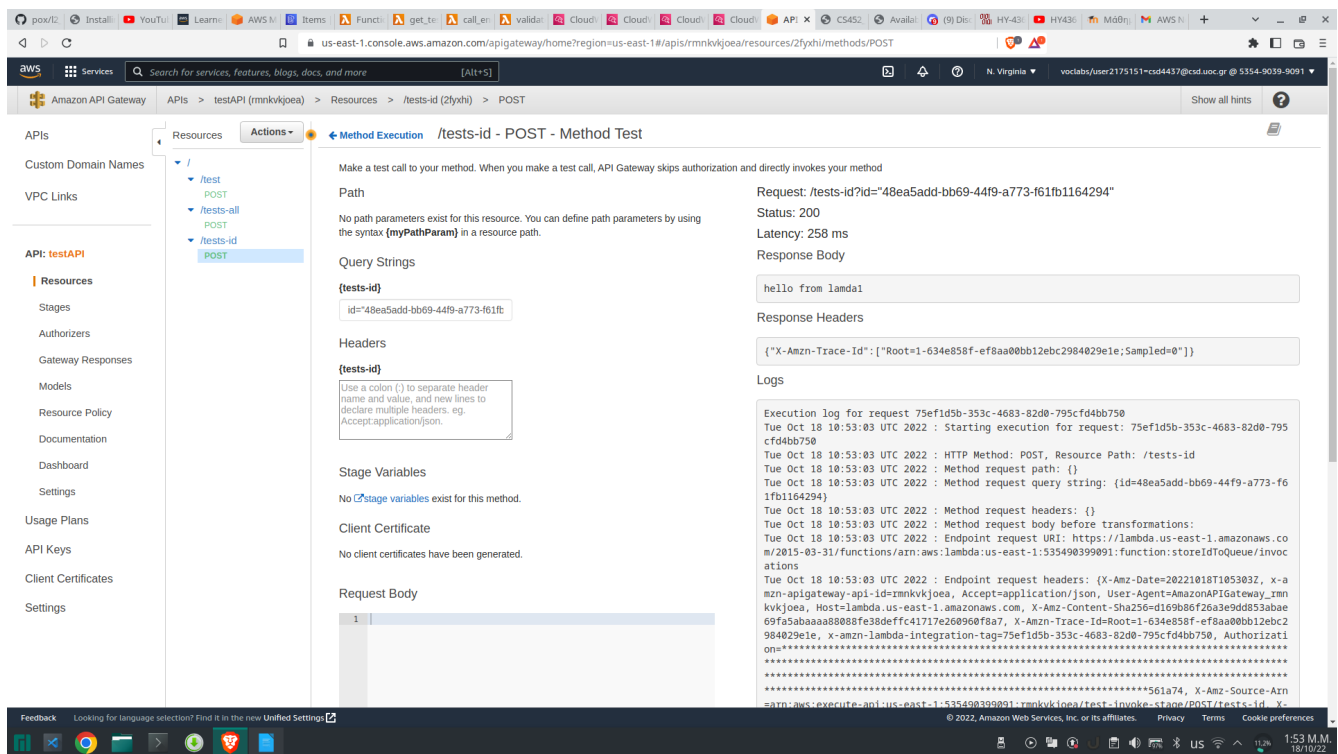


The screenshot shows the Amazon DynamoDB console. The left sidebar contains navigation links for Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main panel displays the 'results' table. The 'Tables (2)' section shows a list of tables: 'Any table tag', 'Find tables by table name', 'results', and 'tests'. The 'results' table is selected. The 'Scan/Query items' section shows a scan query executed on the 'results' table. The results show three items, including one with an invalid ID and URL.

id	result
2cadebf-264d-41a8-...	request with id=48ea5add-bb69-44f9-a773-f61fb1164294 and url=https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add7...
4e0b158a-d742-4083-...	request with id=ae3bceb6-e3d9-4f06-9cde-5afc173dd9e5 and url=https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add7a...
17a0ac86-8964-46c2-...	request with id=NOT VALID ID and url=NOT VALID ID IS NOT VALID



4) Now posts/tests/id and id is correct



us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#item-explorer?initialTagKey=&maximize=true&table=results

DynamoDB

Dashboard
Tables
Update settings
Explore items
 PartiQL editor
Backups
Exports to S3
Imports from S3
Reserved capacity
Settings

▼ DAX
Clusters
Subnet groups
Parameter groups
Events

results

Autopreview View table details

▼ Scan/Query items

Scan/Query a table or index

Scan Query results

Filters

Run Reset

Completed Read capacity units consumed: 0.5

Items returned (4)

Actions Create item

id	result
e6ced70a-ee14-4329...	request with id=48ea5add-bb69-44f9-a773-f61fb1164294 and url=https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=1 IS VALID
cc7f8a8a-095b-4ac1...	request with id=48ea5add-bb69-44f9-a773-f61fb1164294 and url=https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=1 IS VALID
74eaf09d-c955-477e...	request with id=NOT VALID ID and url=NOT VALID ID IS NOT VALID
04edb77b-085f-4a32...	request with id=ae3bceb6-e3d5-4f06-9cde-5afc173dd9e5 and url=https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=2 IS VALID

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

1:58 M.M. 18/10/22

mail.google.com/mail/u/0/#inbox/FMfcgzGqRGRqjVbcNLNQXcBskgJkXjlv

Gmail

Αναζήτηση στην αλληλογραφία

Σύνταξη

Εισερχόμενα 1.000

Με αστέρι
Σε αναβολή
Σημαντικά
Απεσταλμένα
Πρόχειρα 8
Κατηγορίες
Κοινωνικά 425
Ενημερώσεις 94
Φόρουμ
Προσφορές 464
Περισσότερα

Ετικέτες
Notes

1 από 2.347

AWS Notification Message

Εισερχόμενα

AWS Notifications
request with id=NOT VALID ID and url=NOT VALID ID IS NOT VALID -- If you wish to stop receiving notifications from this topic, please click or visit the link be

1:49 μ.μ. (πριν από 8 λεπτά)

AWS Notifications <no-reply@sns.amazonaws.com>
προς εγώ

1:53 μ.μ. (πριν από 4 λεπτά)

Απεργολογή για: Αγγλικά

Αγγλικά Ελληνικά Μετάφραση μηνύματος

request with id=48ea5add-bb69-44f9-a773-f61fb1164294 and url=https://tfr5qh5091.execute-api.eu-central-1.amazonaws.com/beta/add?a=1&b=1 IS VALID

Απάντηση Προώθηση

12 διαγεγραμμένα μηνύματα σε αυτήν τη συνομιλία. Προβολή μηνυμάτων ή μόνιμη διαγραφή.

1:58 M.M. 18/10/22

So post/tests/id works fine both id is correct and not correct