**1.**

Here Postman is the client sent a GET request to the [Postman Echo API](https://www.postman.com/postman/workspace/published-postman-templates/documentation/631643-f695cab7-6878-eb55-7943-ad88e1ccfd65?ctx=documentation) server located at postman-echo.com.

The API server received the request, processed it, and returned a response to Postman.

Postman received the response and displayed it in the Response pane.

Graphical user interface, text, application, email

Description automatically generated

**2.** Created 1st collection named api-lab and saved 1st request in that collection.

Graphical user interface, text, application, email

Description automatically generated

**API CALLS**

**1.**

Status: 200 ok, Response body shows authenticated: true

Graphical user interface, text, application, email

Description automatically generated

**2.**

There is only 1 list of 6 objects in the response body.

Graphical user interface, text, application, email

Description automatically generated

**3.**

No list in the response body

2 object type keys data and support.

Inside data : 5 keys – “id”, “email”, “first name”, “last name” and “avatar”.

Inside support : 2 keys – “url” and “text”.

Graphical user interface, text, application, email

Description automatically generated

**4**

Response status: 204 No content. Response body shows 1.

Again checked user list with get call, found 6 users in the list.

Graphical user interface, text, application, email

Description automatically generated

Checked record of user with id 9. Got response below

{

    "data": {

        "id": 9,

        "email": "tobias.funke@reqres.in",

        "first\_name": "Tobias",

        "last\_name": "Funke",

        "avatar": "https://reqres.in/img/faces/9-image.jpg"

    },

    "support": {

        "url": "https://reqres.in/#support-heading",

        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"

    }

}

**5.**

Send GET request to <https://reqres.in/api/users/40> for user information with ID 40.

Response Status: 404 Not Found.

No list in the response body.

No property in the response body.

**6.**

Sent POST request to create new user. Got Response Status:201 Created.

Available property key: “name”, “job”, “id” and “createdAt”

Header Etag: W/"57-zFQ5TPtYFrYK7Evkkl4OQrnWrpY"

Graphical user interface, text, application, email

Description automatically generated

**7.**

Sent a POST request to login with email: “peter@klaven”

Response code : 400 Bad Request.

Response body : "error": "Missing password"

**8.**

Sent a POST request to login with email: “eve.holt@reqres.in” and password: “cityslicka”

Response Header Etag : W/"1d-lGCrvD6B7Qzk11+2C98+nGhhuec"

Response code : 200 ok Request successful.

Response body: "token": "QpwL5tke4Pnpja7X4"

Graphical user interface, text, application, email

Description automatically generated

**9.**

There is 1 list of objects named “results”

Inside results there are

* 10 nested Lists named “residents”
* 10 nested lists named “films”

**10.**

There are 14 properties in the response body.

Graphical user interface, text, application, email

Description automatically generated

**11.**

1.

“count” : number

“next” : String

“previous” : null

“results” : object

“name” : String

“model” : String

“manufacturer” : String

"cost\_in\_credits" : String

"length": String

"max\_atmosphering\_speed": String

"crew": String

"passengers": String

"cargo\_capacity": String

"consumables": String

"hyperdrive\_rating": String

"MGLT": String

"starship\_class": String

"pilots": array

"films": array

"created": String

"edited": String

"url" : String

2. There is 1 list of objects named “results”

Inside results there are

* 10 nested Lists named “pilots”
* 10 nested lists named “films”

**12.** there are 2 lists in the response body

Graphical user interface, text, application, email

Description automatically generated

**13.**

"count": number

"next": null,

"previous": null,

"results": array

"title": String

"episode\_id": number

"opening\_crawl": String

"director": String

"producer": String

"release\_date": String

"characters": array

“planets”: array

“starships”: array

“vehicles”: array

“species”: array

“created”: String

“edited”: String

“url”: String

There is 1 list of objects named “results”

Inside results there are

* 3 nested Lists named “character”
* 3 nested lists named “planets”
* 3 nested lists named “starships”
* 3 nested lists named “vehicles”
* 3 nested lists named “species”

**14.**

There is 1 list of objects named “results”

Inside results there are

* 10 nested Lists named “people”
* 10 nested lists named “films”

**15.**

Only 1 list in the response body.

Graphical user interface, text, application, email

Description automatically generated

**16.**

Response code 404 Not Found.

The requested booking id not found.

Graphical user interface, text, application, email

Description automatically generated

**17.**

Response code 404 Not Found.

The requested booking id not found.

Graphical user interface, text, application, email

Description automatically generated

**18.**

Response successful. Code : 200 ok.

There is 1 list of objects named “results”

Inside results there are

* 10 nested Lists named “residents”
* 10 nested lists named “films”

**19.**

Response successful. Code : 200 ok.

There is 1 list of objects named “results”

Inside results there are

* 10 nested Lists named “people”
* 10 nested lists named “films”

"count": number

"next": String

"previous": **null**,

"results": array

{

"name": String

"classification": String

"designation": String

"average\_height": String

"skin\_colors": String

"hair\_colors": String

"eye\_colors": String

"average\_lifespan": String

"homeworld": String

"language": String

"people": array

"films": array

"created": String

"edited": String

"url": String

**20.**

a. To retrieve all direct properties of the studio object

JSON file.studio

b. To find out the music’s song

JSON file.studio.music.song

c. To find the rating of all items in the studio

JSON file.studio.movie[\* ].rating

d. To retrieve information on all movies

JSON file.studio.movie[\*]

e. To find out the titles of all movies

JSON file.studio.movie[\*].title

f. To retrieve the titles of all movies by Jack

JSON file.studio.[?(@.director==’jack’)].title

g. To retrieve the category of the last movie

JSON file.studio.[@.length-1].catagory

i. To retrieve all movies that have the isbn property

JSON file.studio[?@.isbn]

**21.**

There is only 1 list in the response body.

Response is successful, code : 200 ok.

“status”: String

"data”: array

“id": number

“employee\_name": String

"employee\_salary": String

"employee\_age": number

"profile\_image": String

"message": String

**22.**

Only 1 data in the response body

Response status : 200 Ok.

**23.**

Response success. Response status : 200 Ok.

Response body asf:

{

    "status": "success",

    "data": {

        "name": "Reza",

        "salary": "123",

        "age": "23",

        "id": 7693

    },

    "message": "Successfully! Record has been added."

}

Id: 7693

**24.**

Response status : 200 Ok.

Response body:

{

    "status": "success",

    "data": "2",

    "message": "Successfully! Record has been deleted"

}

Sent get request to <http://dummy.restapiexample.com/api/v1/employees>

And found 24 employees in the list.

Employee with id: 2 also in the list.

    "status": "success",

    "data": [

        {

            "id": 1,

            "employee\_name": "Tiger Nixon",

            "employee\_salary": 320800,

            "employee\_age": 61,

            "profile\_image": ""

        },

        {

            "id": 2,

            "employee\_name": "Garrett Winters",

            "employee\_salary": 170750,

            "employee\_age": 63,

            "profile\_image": ""

        },

…………………………………………………………….

{

            "id": 24,

            "employee\_name": "Doris Wilder",

            "employee\_salary": 85600,

            "employee\_age": 23,

            "profile\_image": ""

        }

    ],

    "message": "Successfully! All records has been fetched.

**25.**

Response status : 404 Bad request.

{

    "error": "Note: Only defined users succeed registration"

}

Graphical user interface, text, application, email

Description automatically generated

**26.**

Response successful, status : 200 ok

{

    "data": {

        "id": 2,

        "name": "fuchsia rose",

        "year": 2001,

        "color": "#C74375",

        "pantone\_value": "17-2031"

    },

    "support": {

        "url": "https://reqres.in/#support-heading",

        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"

    }

}

**27.**

Response successful, the request has been fulfilled,, status : 201 created

Response body : 1 created

**28.**

Response successful, status : 200 ok Time 3.38 s

**29.**

Response successful, status : 200 ok Time 3.38 s

**29.**

Response status 404 Not FOUND.

This is an error from client side. The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.

{

    "detail": "Not found"

}

**30.**

Response status 404 Not FOUND.

This is an error from client side. The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.

{

    "detail": "Not found"

}

Only 1 property key “detail” in the response body.

**2. Training API using AWS Cognito**

**Overview**

**1By Postman**

Made a post request to URL: <https://izaan-test.auth.us-east-1.amazoncognito.com/oauth2/token>

Verb : POST, In the Request Headers : Added content type application/x-www-form-urlencoded,

in the Request Body: Selected x-www-form-urlencoded and Copied from assignment page key value pairs and pested here.

(Key : scope, value: izaan\_test/post\_info) and (key: grant\_type , value: client\_credentials)

Then in Authorization : Selected Basic Auth.

Copied encoded username and password from assignment and decoded from website : <https://www.base64decode.org/>

Encoded: MXU1aW80dmE5c3I0NW43OWZjZWcyZGFtamY6MXFia3RodnA3bGJjN2FhdnVoaG1mZzhmMmNyZWtvcjloMmg3YWJ1Mm9ydTFubHBqNzFmZQ==

Decoded result : 1u5io4va9sr45n79fceg2damjf:1qbkthvp7lbc7aavuhhmfg8f2crekor9h2h7abu2oru1nlpj71fe

Username: 1u5io4va9sr45n79fceg2damjf

Password: 1qbkthvp7lbc7aavuhhmfg8f2crekor9h2h7abu2oru1nlpj71fe

Result: Token received

Token : eyJraWQiOiJtNlhhM3FkczlxY2o2ODZYOVBRb0JqTEJSQU5yMzBPZ21ZQzdvdThZc1N3PSIsImFsZyI6IlJTMjU2In0.eyJzdWIiOiIxdTVpbzR2YTlzcjQ1bjc5ZmNlZzJkYW1qZiIsInRva2VuX3VzZSI6ImFjY2VzcyIsInNjb3BlIjoiaXphYW5fdGVzdFwvcG9zdF9pbmZvIiwiYXV0aF90aW1lIjoxNjY2NTM1NDQ0LCJpc3MiOiJodHRwczpcL1wvY29nbml0by1pZHAudXMtZWFzdC0xLmFtYXpvbmF3cy5jb21cL3VzLWVhc3QtMV9ZN3k4cTVDMVoiLCJleHAiOjE2NjY1MzkwNDQsImlhdCI6MTY2NjUzNTQ0NCwidmVyc2lvbiI6MiwianRpIjoiZTA3ZjBmZDMtMDM3OS00NzliLWFkN2ItNmNiMGU3OTA3NTUwIiwiY2xpZW50X2lkIjoiMXU1aW80dmE5c3I0NW43OWZjZWcyZGFtamYifQ.G1Sw4UOp7Q5cZfyY3DGPa2HHur6PCZRoUXoWoflQ8018Zc3lSTb27izYc7uSCVwoipMTG\_kFi0hca54DFS2YhvTiX4zLpL-kDuPs-YP46twOY9WivWTdLyIFNTBYM0VLlRqyt7OD24g-UyYiBamquHDRgP8kEud7q9\_Ih2JzeR\_Uq-ERbQWUOqK0y0Etl1MYluofMDQ9h1GW91pr621GxS54vgcdT6cIETzvirml1n1OzK9ttRA24dHpjMSxp3fch1jJFl5LLQPaagkarT4aNattoIwHVImAAwbe3fDkNIwH8Y5cjndF13FsL9CxW12G1N---v1uN7iO5sM1In8-7A

**By Gitbash**

Received Token

Text

Description automatically generated

Token: eyJraWQiOiJtNlhhM3FkczlxY2o2ODZYOVBRb0JqTEJSQU5yMzBPZ21ZQzdvdThZc1N3PSIsImFsZyI6IlJTMjU2In0.eyJzdWIiOiIxdTVpbzR2YTlzcjQ1bjc5ZmNlZzJkYW1qZiIsInRva2VuX3VzZSI6ImFjY2VzcyIsInNjb3BlIjoiaXphYW5fdGVzdFwvcG9zdF9pbmZvIiwiYXV0aF90aW1lIjoxNjY2NjI5MzE3LCJpc3MiOiJodHRwczpcL1wvY29nbml0by1pZHAudXMtZWFzdC0xLmFtYXpvbmF3cy5jb21cL3VzLWVhc3QtMV9ZN3k4cTVDMVoiLCJleHAiOjE2NjY2MzI5MTcsImlhdCI6MTY2NjYyOTMxNywidmVyc2lvbiI6MiwianRpIjoiOTY0MjZhNDAtODQ2OC00ZTE5LTlkMzMtZjNlZDgxMDAwYjI2IiwiY2xpZW50X2lkIjoiMXU1aW80dmE5c3I0NW43OWZjZWcyZGFtamYifQ.LuPmmhDJUAdJFpmmANVOQQyUJ3ZpEBhqso-zP2UXKKV-2dPZZjd2OKXRTVYV0gDFhRB7IG5xQGbSCmiZfh19bHUVH0bN6TxZG4F8Wpq\_3K3BQON7Mn8gD1-r9j-zhupxyhmn8AkSgAyPCznmatYKuzDY2TBDRW9z2RIzv1EgqWYwb-y5JX3j16EpLYodFN-y5hstyPfPMsdEwJv3h05\_DPbRXqKRgGJg4zmwehWUcp\_4nDMQNWEuq0W4SAOpkVBTzjJ5BV9qbZRKmFjWJCeGFnhYuOf9DltWff7dT2viuf\_abW73-iQoHUrDAGFt\_QsBtrNTe3nhKlf46jHUFBnH9g

**2.**

**By Postman**

Sent a post request using the token received in the last call with name and age data.

Response received successfully. Status: 200 ok.

Graphical user interface, text, application, email

Description automatically generated

**By Gitbash**

Text

Description automatically generated

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 62 100 31 100 31 38 38 --:--:-- --:--:-- --:--:-- 77{"message": "You are an Adult"}

## GitHub API

1.Sent a post request to <https://api.github.com/user/repos> with a github personal access code to create a public Github repository name “PostmanTest”

Successful response received. Status code: 201 Created.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**2.** Sent a get request to <https://api.github.com/repos/sreza30th/WebApp> with personal access code to get the existing repository name “WebAPP”.

Response successful, Status code 200 ok

Graphical user interface, text, application, email

Description automatically generated

**3.** Sent a DELETE request to https://api.github.com/repos/sreza30th/PostmanTest/automated-security-fixes with personal access code to Delete the existing repository name “PostmanTest”.

Response successful, Status code 204 No Content.

Graphical user interface, text, application, email

Description automatically generated

Note: before sending the DELETE request in the Github under the repository PostmanTest(Setting> Code Security and analysis, disable Dependboat alerts and Dependboat security updates)

**4.** Sent a get request to <https://api.github.com/repos/sreza30th/GitAndGithubReza/git/refs/heads> with bearer token to get the sha code. Successful response received. Status : 200 ok.

Graphical user interface, text, application, email

Description automatically generated

Sent a POST request to <https://api.github.com/repos/sreza30th/GitAndGithubReza/git/refs> with bearer token and Json body {

      "ref": "refs/heads/<BranchfromPostman>",

      "sha": "69781c1d71a893002e22bab42c7b9fc79689ab2c"

  }

to create a new branch name “BranchfromPostman” under main branch of git repository “GitAndGithubReza”

Successful response received. Status: 201 Created.

Graphical user interface, text, application, email

Description automatically generated

