Github Link

https://github.com/sudtanj/Auth-Micro-Services

Architecture Workflow

1. User login: The user must login to the system using their credentials (username and password) to gain access to the system.

2. Authentication: The system then verifies the user’s credentials and grants access if the credentials are valid.

3. User role: Depending on the user role, the system will allow different levels of access to the API.

4. Create: If the user has the privilege to create a new user, they can do so by providing the required data (username, password, etc.).

5. Read: If the user has the privilege to read user information, they can do so by providing the user ID.

6. Update: If the user has the privilege to update user information, they can do so by providing the required data (username, password, etc.).

7. Delete: If the user has the privilege to delete user information, they can do so by providing the user ID.

Admin Credential

```
{
   "username": "test",
   "password": "testtesttest",
}
```

.Env data

```
PORT=3000
DATABASE_URL=postgres://vdigaftt:xWvJcKGFA20wjvmXnRt0QH9QboSKmBTq@rosie.db.
elephantsql.com/vdigaftt
APP_SECRET=0dd8d1d7c673300e0e800e10e13eb6ee1414c140e046ebf7e2229010ab7ab7
9a10f06fddeebabfb428b6a380aa12654c
```

Applying Secret to Kubernetes namespace

secret.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
  namespace: auth-micro-services
type: Opaque
data:
  APP_SECRET:
MGRkOGQxZDdjNjczMzAwZTBlODAwZTEwZTEzZWI2ZWUxNDE0YzE0MGUwNDZlYmY3Z
TIyMjkwMTBhYjdhYjc5YTEwZjA2ZmRkZWViYWJmYjQyOGI2YTM4MGFhMTI2NTRj
  PORT: MzAwMA==
  DATABASE_URL:
cG9zdGdyZXM6Ly92ZGlnYWZ0dDp4V3ZKY0tHRkEyMHdqdm1YblJ0MFFIOVFib1NLbUJUc
UByb3NpZS5kYi5lbGVwaGFudHNxbC5jb20vdmRpZ2FmdHQ=
```

Apply the secret to kubernetes cluster using command

```
Kubectl apply -f secret.yaml
```

Documentation

Documentation can be access at http://localhost:3000/api (local deployment) or http://localhost/api (kubernetes deployment). You can also import manually the swagger json by using the following file.

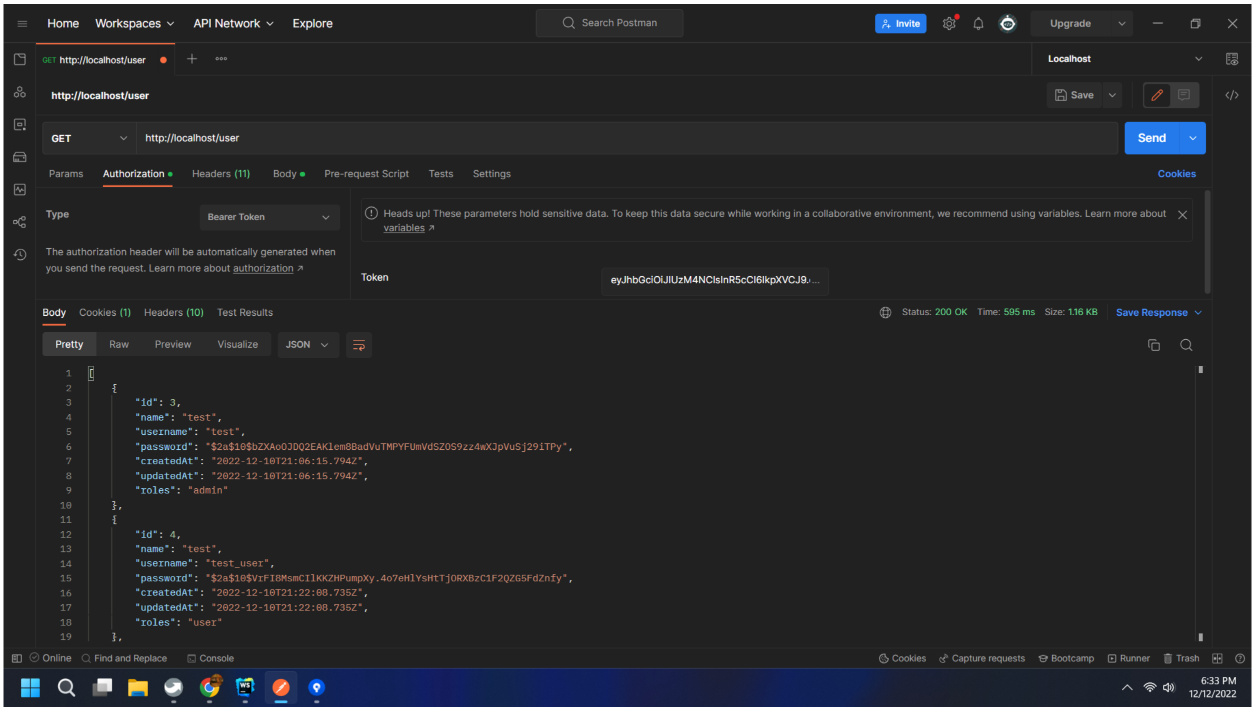
Swagger.json

```
{
  "openapi": "3.0.0",
  "paths": {
    "/": {
      "get": {
        "operationId": "AppController_getHello",
        "parameters": [],
        "responses": {
          "200": {
            "description": ""
          }
        }
      }
    },
    "/health": {
      "get": {
        "operationId": "HealthController_check",
        "summary": "For docker purposes! to get the health status of the container",
        "parameters": [],
        "responses": {
          "200": {
            "description": "The Health Check is successful",
            "content": {
              "application/json": {
                "schema": {
                  "type": "object",
                  "properties": {
                    "status": {
                      "type": "string",
                      "example": "ok"
                    },
                    "info": {
                      "type": "object",
                      "example": {
                        "database": {
                          "status": "up"
                        }
                      },
                      "additionalProperties": {
                        "type": "object",
                        "properties": {
                          "status": {
                            "type": "string"
                          }
                        },
                        "additionalProperties": {
                          "type": "string"
                        }
                      },
                      "nullable": true
                    },
                    "error": {
                      "type": "object",
                      "example": {},
                      "additionalProperties": {
                        "type": "object",
                        "properties": {
                          "status": {
                            "type": "string"
                          }
                        },
                        "additionalProperties": {
                          "type": "string"
                        }
                      },
                      "nullable": true
                    },
                    "details": {
                      "type": "object",
                      "example": {
                        "database": {
                          "status": "up"
                        }
                      },
                      "additionalProperties": {
                        "type": "object",
                        "properties": {
                          "status": {
                            "type": "string"
                          }
                        },
                        "additionalProperties": {
                          "type": "string"
                        }
                      }
                    }
                  }
                }
              }
            }
          },
          "503": {
            "description": "The Health Check is not successful",
            "content": {
              "application/json": {
                "schema": {
                  "type": "object",
                  "properties": {
                    "status": {
                      "type": "string",
                      "example": "error"
                    },
                    "info": {
                      "type": "object",
                      "example": {
                        "database": {
                          "status": "up"
                        }
                      },
                      "additionalProperties": {
                        "type": "object",
                        "properties": {
                          "status": {
                            "type": "string"
                          }
                        },
                        "additionalProperties": {
                          "type": "string"
                        }
                      },
                      "nullable": true
                    },
                    "error": {
                      "type": "object",
                      "example": {
                        "redis": {
                          "status": "down",
                          "message": "Could not connect"
                        }
                      },
                      "additionalProperties": {
                        "type": "object",
                        "properties": {
                          "status": {
                            "type": "string"
                          }
                        },
                        "additionalProperties": {
                          "type": "string"
                        }
                      },
                      "nullable": true
                    },
                    "details": {
                      "type": "object",
                      "example": {
                        "database": {
                          "status": "up"
                        },
                        "redis": {
                          "status": "down",
                          "message": "Could not connect"
                        }
                      },
                      "additionalProperties": {
```

```
                    "type": "object",
                    "properties": {
                      "status": {
                        "type": "string"
                      }
                    },
                    "additionalProperties": {
                      "type": "string"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
},
"/user": {
  "get": {
    "operationId": "UserController_list",
    "summary": "list all user available. admin and user can access!",
    "parameters": [],
    "responses": {
      "200": {
        "description": "Success.",
        "content": {
          "application/json": {
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/components/schemas/User"
              }
            }
          }
        }
      }
    },
    "tags": [
      "User"
    ],
    "security": [
      {
        "bearer": []
      }
    ]
  },
  "post": {
    "operationId": "UserController_register",
        "summary": "create new user. admin can access!",
        "parameters": [],
        "requestBody": {
          "required": true,
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/SignUp"
              }
            }
          }
        },
        "responses": {
          "201": {
            "description": "Success.",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/User"
                }
              }
            }
          }
        },
        "tags": [
          "User"
        ],
        "security": [
          {
            "bearer": []
          }
        ]
      }
    },
    "/user/{id}": {
      "get": {
        "operationId": "UserController_detail",
        "summary": "get detail of user. admin and user can access!",
        "parameters": [
          {
            "name": "id",
            "required": true,
            "in": "path",
            "schema": {
              "type": "string"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Success.",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/User"
                }
              }
            }
          }
        },
        "tags": [
          "User"
        ],
        "security": [
          {
            "bearer": []
          }
        ]
      },
      "patch": {
        "operationId": "UserController_update",
        "summary": "update existing user. admin can access!",
        "parameters": [
          {
            "name": "id",
            "required": true,
            "in": "path",
            "schema": {
              "type": "string"
            }
          }
        ],
        "requestBody": {
          "required": true,
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/User"
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "Success.",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/User"
                }
              }
            }
          }
        },
        "tags": [
          "User"
        ],
        "security": [
          {
            "bearer": []
          }
        ]
      },
      "delete": {
        "operationId": "UserController_delete",
        "summary": "delete existing user. admin can access!",
        "parameters": [
          {
            "name": "id",
            "required": true,
            "in": "path",
            "schema": {
              "type": "string"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Success.",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/User"
                }
              }
            }
          }
        },
        "tags": [
          "User"
        ],
        "security": [
          {
            "bearer": []
          }
        ]
      }
    },
    "/auth/register": {
      "post": {
        "operationId": "AuthController_register",
        "summary": "register new user. cannot register admin!",
        "parameters": [],
        "requestBody": {
          "required": true,
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/SignUp"
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "Success.",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/User"
                }
              }
            }
          }
        },
        "tags": [
          "Auth"
        ]
      }
    },
    "/auth/login": {
      "post": {
        "operationId": "AuthController_login",
        "summary": "login and get user token!",
        "parameters": [],
        "requestBody": {
          "required": true,
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/SignInDto"
              }
            }
          }
        },
        "responses": {
          "200": {
            "description": "Success.",
            "headers": {
              "Authorization": {
                "description": "token user will be store here at the header authorization response",
                "example": "Bearer
eyJhbGciOiJIUzM4NCIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0ZXN0X3VzZXIiLCJpYXQiOjE2NzA
3MzI1NTEsImV4cCI6MTY3MDgxODk1MX0.iw72t9fNF1_sAo_8Vpp7beGJZA46B7JD23Ey59
8jVni4jhtIjF0WD_akGjhVfgG5"
              }
            },
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/User"
                }
              }
            }
          }
        },
        "tags": [
          "Auth"
        ]
      }
    },
    "/auth/me": {
      "get": {
        "operationId": "AuthController_me",
        "summary": "get current token profile!",
        "parameters": [],
        "responses": {
          "200": {
            "description": "Success.",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/User"
                }
              }
            }
          }
        },
        "tags": [
          "Auth"
        ]
      }
    }
  },
  "info": {
    "title": "Auth Micro Services Documentation ",
    "description": "The API is use for micro services architecture deployment for
authentication",
    "version": "1.0",
    "contact": {}
```

```
},
"tags": [
  {
    "name": "auth",
    "description": ""
  }
],
"servers": [],
"components": {
  "schemas": {
    "User": {
      "type": "object",
      "properties": {
        "id": {
          "type": "number",
          "example": 1,
          "description": "Postgres id"
        },
        "name": {
          "type": "string",
          "example": "John",
          "description": "name of the user"
        },
        "username": {
          "type": "string",
          "example": "john",
          "description": "username for login"
        },
        "roles": {
          "type": "string",
          "example": "admin",
          "description": "User role"
        }
      },
      "required": [
        "id",
        "name",
        "username",
        "roles"
      ]
    },
    "SignUp": {
      "type": "object",
      "properties": {
        "name": {
          "type": "string",
          "example": "John",
          "description": "name of the user"
        },
        "username": {
            "type": "string",
            "example": "john",
            "description": "username for login"
          },
          "password": {
            "type": "string",
            "example": "password",
            "description": "user password for login"
          },
          "roles": {
            "type": "string",
            "example": "admin",
            "description": "User role"
          }
        },
        "required": [
          "name",
          "username",
          "password",
          "roles"
        ]
      },
      "SignInDto": {
        "type": "object",
        "properties": {
          "username": {
            "type": "string",
            "example": "john",
            "description": "username for login"
          },
          "password": {
            "type": "string",
            "example": "password",
            "description": "user password for login"
          }
        },
        "required": [
          "username",
          "password"
        ]
      }
    }
  }
}
```

Screenshot



http://localhost/user

GET http://localhost/user

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

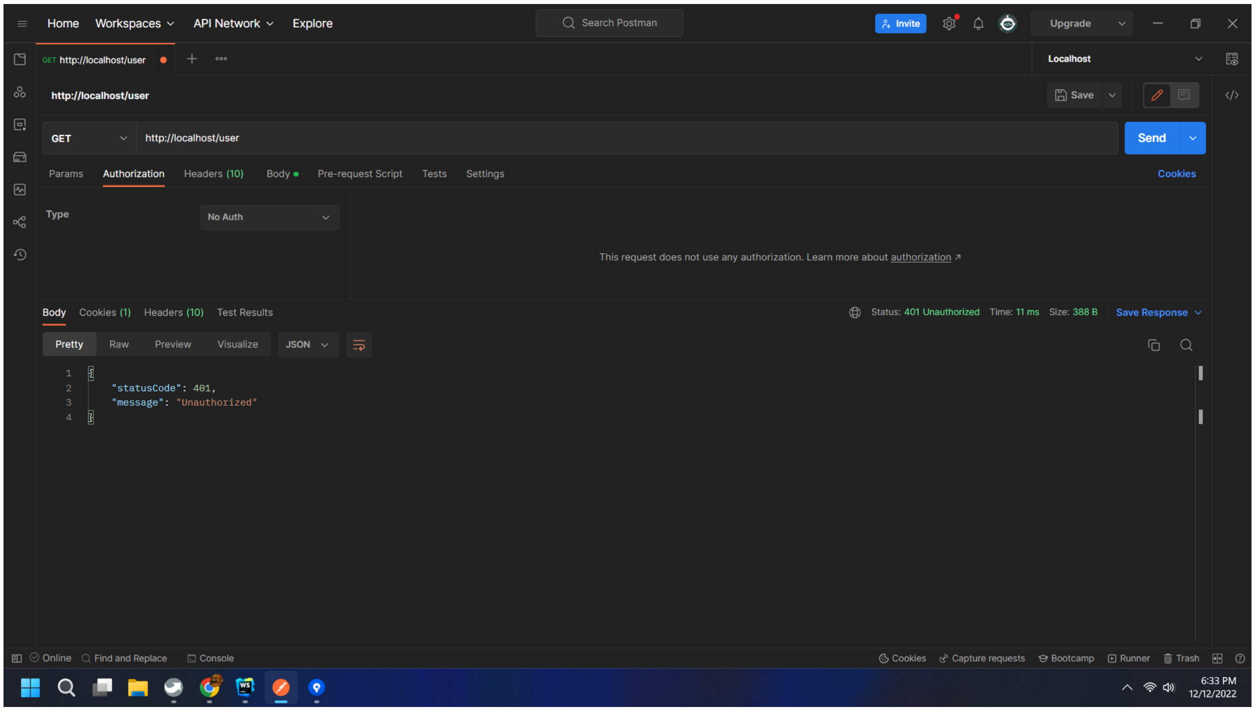
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables ↗

Token: eyJhbGciOiJIUzM4NCIsInR5cCI6IkpXVCJ9.…

Body Cookies (1) Headers (10) Test Results

Status: 200 OK Time: 595 ms Size: 1.16 KB Save Response

```
[
    {
        "id": 3,
        "name": "test",
        "username": "test",
        "password": "$2a$10$bZXAoOJDQ2EAKlem8BadVuTMPYFUmVdSZOS9zz4wXJpVuSj29iTPy",
        "createdAt": "2022-12-10T21:06:15.794Z",
        "updatedAt": "2022-12-10T21:06:15.794Z",
        "roles": "admin"
    },
    {
        "id": 4,
        "name": "test",
        "username": "test_user",
        "password": "$2a$10$VrFI8MsmCIlKKZHPumpXy.4o7eHlYsHtTjORXBzC1F2QZG5FdZnfy",
        "createdAt": "2022-12-10T21:22:08.735Z",
        "updatedAt": "2022-12-10T21:22:08.735Z",
        "roles": "user"
    },
```



http://localhost/user

GET http://localhost/user

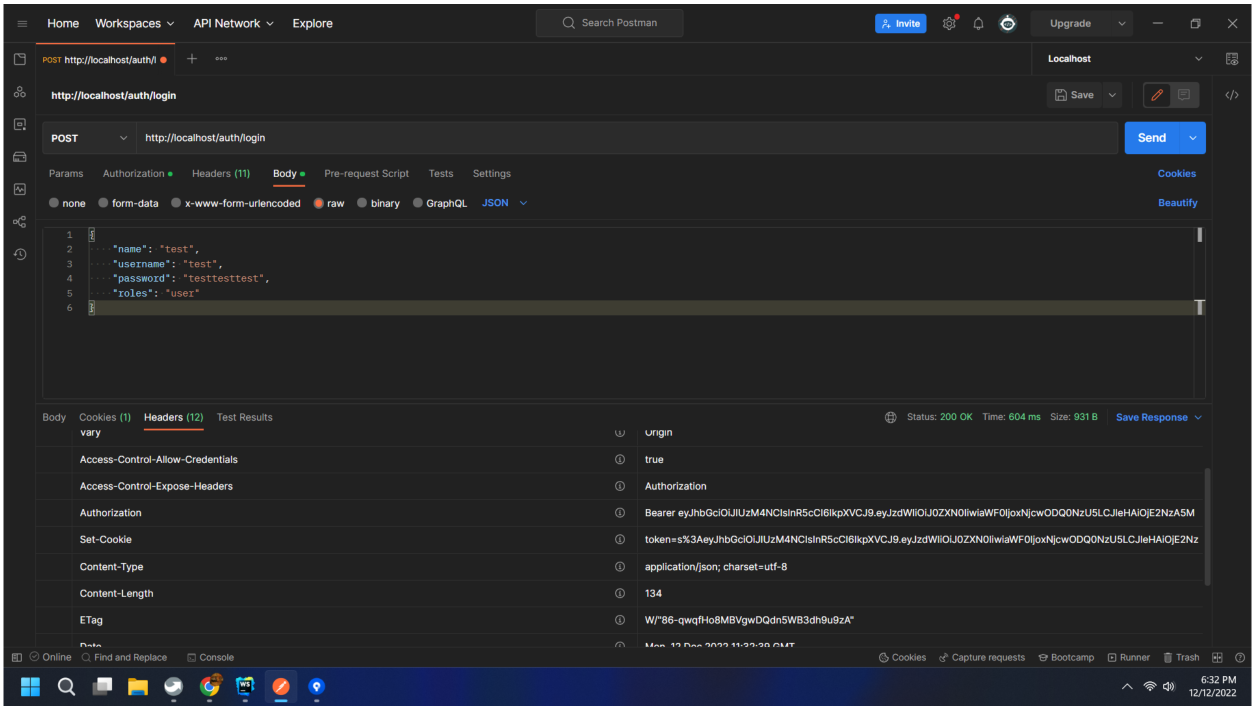
Type: No Auth

This request does not use any authorization. Learn more about authorization ↗

Body Cookies (1) Headers (10) Test Results

Status: 401 Unauthorized Time: 11 ms Size: 388 B Save Response

```
{
    "statusCode": 401,
    "message": "Unauthorized"
}
```



POST http://localhost/auth/login

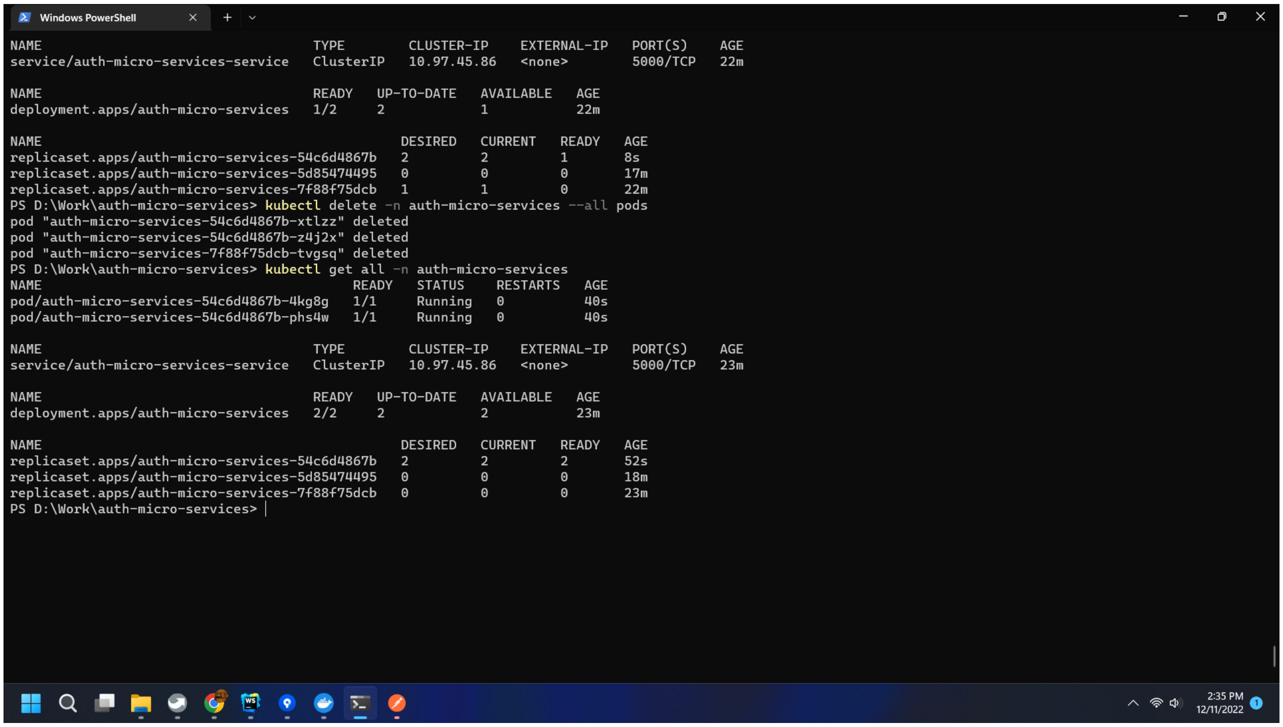
Params Authorization Headers (11) **Body** Pre-request Script Tests Settings

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

```
{
    "name": "test",
    "username": "test",
    "password": "testtesttest",
    "roles": "user"
}
```

Body Cookies (1) **Headers (12)** Test Results — Status: 200 OK Time: 604 ms Size: 931 B Save Response

vary	Origin
Access-Control-Allow-Credentials	true
Access-Control-Expose-Headers	Authorization
Authorization	Bearer eyJhbGciOiJIUzM4NCIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0ZXN0IiwiaWF0IjoxNjcwODQ0NzU5LCJleHAiOjE2NzA5M
Set-Cookie	token=s%3AeyJhbGciOiJIUzM4NCIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0ZXN0IiwiaWF0IjoxNjcwODQ0NzU5LCJleHAiOjE2Nz
Content-Type	application/json; charset=utf-8
Content-Length	134
ETag	W/"86-qwqfHo8MBVgwDQdn5WB3dh9u9zA"
Date	Mon, 12 Dec 2022 11:32:39 GMT



```
NAME                                   TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE
service/auth-micro-services-service   ClusterIP   10.97.45.86   <none>        5000/TCP   22m

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/auth-micro-services   1/2     2            1           22m

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/auth-micro-services-54c6d4867b   2         2         1       8s
replicaset.apps/auth-micro-services-5d85474495   0         0         0       17m
replicaset.apps/auth-micro-services-7f88f75dcb   1         1         0       22m
PS D:\Work\auth-micro-services> kubectl delete -n auth-micro-services --all pods
pod "auth-micro-services-54c6d4867b-xtlzz" deleted
pod "auth-micro-services-54c6d4867b-z4j2x" deleted
pod "auth-micro-services-7f88f75dcb-tvgsq" deleted
PS D:\Work\auth-micro-services> kubectl get all -n auth-micro-services
NAME                                      READY   STATUS    RESTARTS   AGE
pod/auth-micro-services-54c6d4867b-4kg8g   1/1     Running   0          40s
pod/auth-micro-services-54c6d4867b-phs4w   1/1     Running   0          40s

NAME                                   TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE
service/auth-micro-services-service   ClusterIP   10.97.45.86   <none>        5000/TCP   23m

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/auth-micro-services   2/2     2            2           23m

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/auth-micro-services-54c6d4867b   2         2         2       52s
replicaset.apps/auth-micro-services-5d85474495   0         0         0       18m
replicaset.apps/auth-micro-services-7f88f75dcb   0         0         0       23m
PS D:\Work\auth-micro-services>
```