Api Contract for AuthService

1. api/auth/getaccesstoken

Description: Api should return the jwt token

POST : /api/authservice/getaccesstoken

Host: <Host\_Name>

Content-Type: application/json

Body : {

"userName": <encoded-username>,

"password": <encoded-password>

}

Response:

HTTP/1.1 200/401

Vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers

Cache-Control: no-store

Pragma: no-cache

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

X-Frame-Options: DENY

Transfer-Encoding: chunked, chunked

Date: Tue, 03 Aug 2021 12:10:16 GMT

Keep-Alive: timeout=60

Connection: keep-alive, close

Content-Type: application/json;charset=UTF-8

{

"access\_token": <access\_token>

"token\_type": "bearer",

"refresh\_token": <refresh\_token>,

"expires\_in": <expire-time>,

"scope": <scopes>,

"jti": <jti>

}

Request : Valid Credentials

POST /api/authservice/getaccesstoken

Host: localhost: 8084

Content-Type: application/json

{

"userName": "doug@123.com",

"password": "doug"

}

Response:

HTTP/1.1 200

Vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers

Cache-Control: no-store

Pragma: no-cache

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

X-Frame-Options: DENY

Transfer-Encoding: chunked, chunked

Date: Tue, 03 Aug 2021 12:10:16 GMT

Keep-Alive: timeout=60

Connection: keep-alive, close

Content-Type: application/json;charset=UTF-8

{

"access\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOlsiY291cG9uc2VydmljZSJdLCJleHAiOjE2MjgwMzU4MTYsInVzZXJfbmFtZSI6ImRvdWdAMTIzLmNvbSIsImp0aSI6ImNkMTM2MTdhLTE3MTYtNGY1OS04MWMxLTRhYzUwMzJjNTcwZSIsImNsaWVudF9pZCI6ImNvdXBvbmNsaWVudGFwcCIsInNjb3BlIjpbInJlYWQiLCJ3cml0ZSJdfQ.2GmWeWhfcAOu2NskNSrxrrgjoVNOaj6otMfKplpenyQ",

"token\_type": "bearer",

"refresh\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOlsiY291cG9uc2VydmljZSJdLCJ1c2VyX25hbWUiOiJkb3VnQDEyMy5jb20iLCJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiYXRpIjoiY2QxMzYxN2EtMTcxNi00ZjU5LTgxYzEtNGFjNTAzMmM1NzBlIiwiZXhwIjoxNjMwNTg0NjE2LCJqdGkiOiI4MGZlMWIwZC0wMjg2LTRmOGItOGQ0Yy1lNDJhNDlkYmYyYzEiLCJjbGllbnRfaWQiOiJjb3Vwb25jbGllbnRhcHAifQ.eqDSsOQVjpOXWx6R\_FWrOJ4pmXEjo2AEdi9sscg-O-g",

"expires\_in": 43199,

"scope": "read write",

"jti": "cd13617a-1716-4f59-81c1-4ac5032c570e"

}

When credentials are not valid

HTTP/1.1 400 Bad Request

{

"error\_description":"Bad credentials",

"Error":"invalid\_grant"

}

1. api/auth/getrefreshtoken

Description: Api should return the jwt token with given refresh token

POST : /api/authservice/getaccesstoken

Host: <Host\_Name>

Content-Type: application/json

Body : {

"refreshToken": <encoded-username>

}

Response:

HTTP/1.1 200/401/400

Vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers

Cache-Control: no-store

Pragma: no-cache

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

X-Frame-Options: DENY

Transfer-Encoding: chunked, chunked

Date: Tue, 03 Aug 2021 12:10:16 GMT

Keep-Alive: timeout=60

Connection: keep-alive, close

Content-Type: application/json;charset=UTF-8

{

"access\_token": <access\_token>

"token\_type": "bearer",

"refresh\_token": <refresh\_token>,

"expires\_in": <expire-time>,

"scope": <scopes>,

"jti": <jti>

}

Request : Valid refresh token

POST /api/authservice/getrefreshtoken

Host: localhost: 8084

Content-Type: application/json

{

"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOlsiY291cG9uc2VydmljZSJdLCJ1c2VyX25hbWUiOiJmYWl6QDEyMy5jb20iLCJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiYXRpIjoiNzk1YmU5OGUtNTcxOC00NzkyLTg5YTUtYjE2NDRlNmI4YmQ0IiwiZXhwIjoxNjI4NDMxNzEyLCJhdXRob3JpdGllcyI6WyJST0xFX0FETUlOIl0sImp0aSI6ImVjNzIzM2I1LWU0YzQtNGJiOS1hZDRmLTA5OGE5ZDYzNGI1ZSIsImNsaWVudF9pZCI6ImNvdXBvbmNsaWVudGFwcCJ9.tA1mEUE8Jyh508n81uAA3XJKK9KM90neec061-BebKk

"

}

{

"access\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOlsiY291cG9uc2VydmljZSJdLCJ1c2VyX25hbWUiOiJmYWl6QDEyMy5jb20iLCJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiZXhwIjoxNjI4NDMxNjExLCJhdXRob3JpdGllcyI6WyJST0xFX0FETUlOIl0sImp0aSI6ImM0NjdkYTAyLWU1ODYtNGE4ZS04OTQ4LTFjOGUwZWI3MzJjZCIsImNsaWVudF9pZCI6ImNvdXBvbmNsaWVudGFwcCJ9.juHwJLc8aFKD0bh-DE-6Ry8I8qgsiLerxgCZfIpbFLI",

"token\_type": "bearer",

"refresh\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOlsiY291cG9uc2VydmljZSJdLCJ1c2VyX25hbWUiOiJmYWl6QDEyMy5jb20iLCJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiYXRpIjoiYzQ2N2RhMDItZTU4Ni00YThlLTg5NDgtMWM4ZTBlYjczMmNkIiwiZXhwIjoxNjI4NDMxNzEyLCJhdXRob3JpdGllcyI6WyJST0xFX0FETUlOIl0sImp0aSI6ImVjNzIzM2I1LWU0YzQtNGJiOS1hZDRmLTA5OGE5ZDYzNGI1ZSIsImNsaWVudF9pZCI6ImNvdXBvbmNsaWVudGFwcCJ9.AK\_mWCrdiwlu0gjN2H2\_o4vmBSdeS9JkMJWlE3YQMkI",

"expires\_in": 119,

"scope": "read write",

"jti": "c467da02-e586-4a8e-8948-1c8e0eb732cd"

}

When Token is invalid or expired

HTTP/1.1 401 Unauthorized

{

"error\_description":"Invalid refresh token (expired)",

"Error":"invalid\_token"

}