9.4 HTTP, URL, JSON, API: Homework

Part 1. Consider the GET request: /api/v0/aircrafts/export

- Read the documentation in Swagger http://***.***.***.***:****/swagger-ui/index.html#/Aircraft/exportAllAsCsvFile and describe what this method does.
- How to run this method from curl? What happened?
- How to redirect the result to a file instead of a terminal?
- Say, we need to see an empty result to check how the method works in this case. What preparation needs to be done in the system?

Part 2. Explain the components of URI:

- https://john.doe@www.example.com:8080/forum/questions/?tag=networking&order=newest#top
- telnet://192.0.5.105:23/

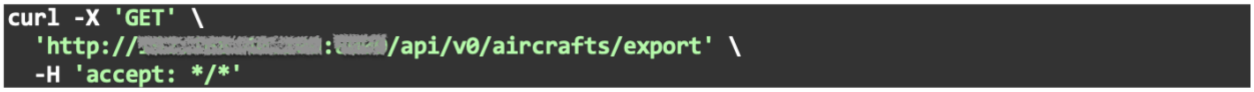
Part 3* (optional for 100 points): Explain each component of the following command:

curl -i -H 'X-USER-IDENTITY-DOMAIN-NAME: OAuthTestTenant125' -H 'Authorization: Basic MzAzYTI0OTItZDY0Zi00ZTA0LWI3OGYtYjQzMzAwNDczMTJiOll5Sk5NSkdFc0ZqUkxWZVZsdVMz' -H 'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' --request POST https://<idm-domain>.identity.<data-center>.oraclecloud.com/oauth/tokens -d 'grant_type=password &username=tenantAdminUser &password=Fusionapps1&scope=http://www.example.com'

1. **Part 1. Consider the GET request: /api/v0/aircrafts/export**
   - **Read the documentation in Swagger http://***.***.***.***:****/swagger-ui/index.html#/Aircraft/exportAllAsCsvFile and describe what this method does.**

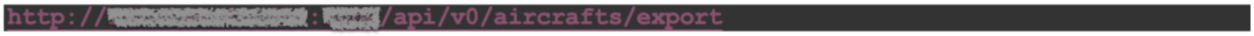
     The **GET** request: **/api/v0/aircrafts/export** Attempt to export all aircraft records to CSV file.
     For this method there is no parameters.
     Responses could be:
     200 – OK,
     400 – Bad Request.

     I tried to execute this method:
     In Responses get:
     Curl:



```
curl -X 'GET' \
  'http://[redacted]:[redacted]/api/v0/aircrafts/export' \
  -H 'accept: */*'
```

- Request URL



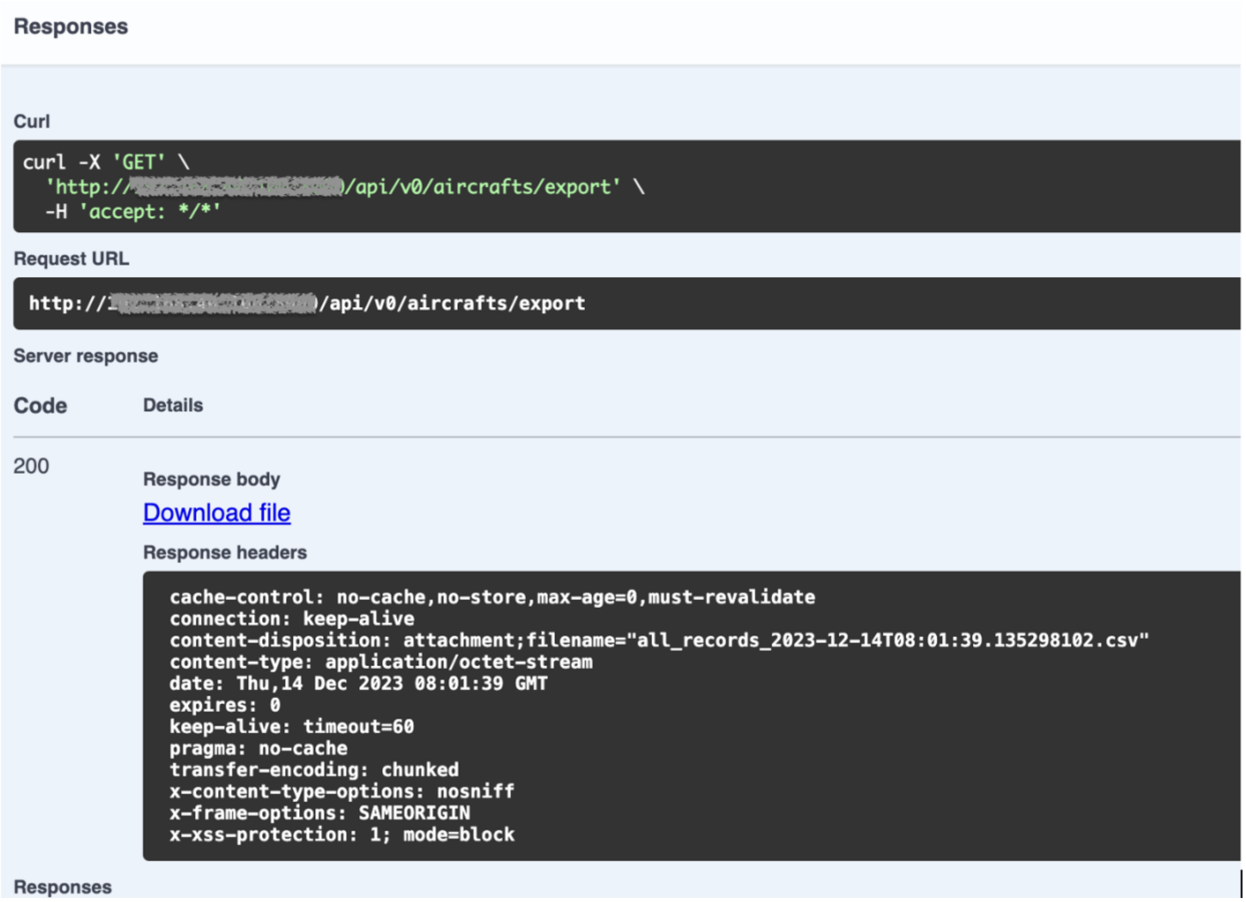
```
http://[redacted]:[redacted]/api/v0/aircrafts/export
```

- Server response

The Result was 200 – OK.
I got a file:
all_records_2023-12-14T08_01_39.135298102.csv

Response headers:

```
cache-control: no-cache,no-store,max-age=0,must-revalidate  connection: keep-alive
content-disposition: attachment;filename="all_records_2023-12-
14T08:01:39.135298102.csv"  content-type: application/octet-stream  date: Thu,14 Dec
2023 08:01:39 GMT  expires: 0  keep-alive: timeout=60  pragma: no-cache  transfer-
encoding: chunked  x-content-type-options: nosniff  x-frame-options: SAMEORIGIN  x-
xss-protection: 1; mode=block
```



Responses

Curl

```
curl -X 'GET' \
  'http://[redacted]/api/v0/aircrafts/export' \
  -H 'accept: */*'
```

Request URL

```
http://[redacted]/api/v0/aircrafts/export
```

Server response

Code	Details
200	Response body Download file Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-disposition: attachment;filename="all_records_2023-12-14T08:01:39.135298102.csv"
content-type: application/octet-stream
date: Thu,14 Dec 2023 08:01:39 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
```

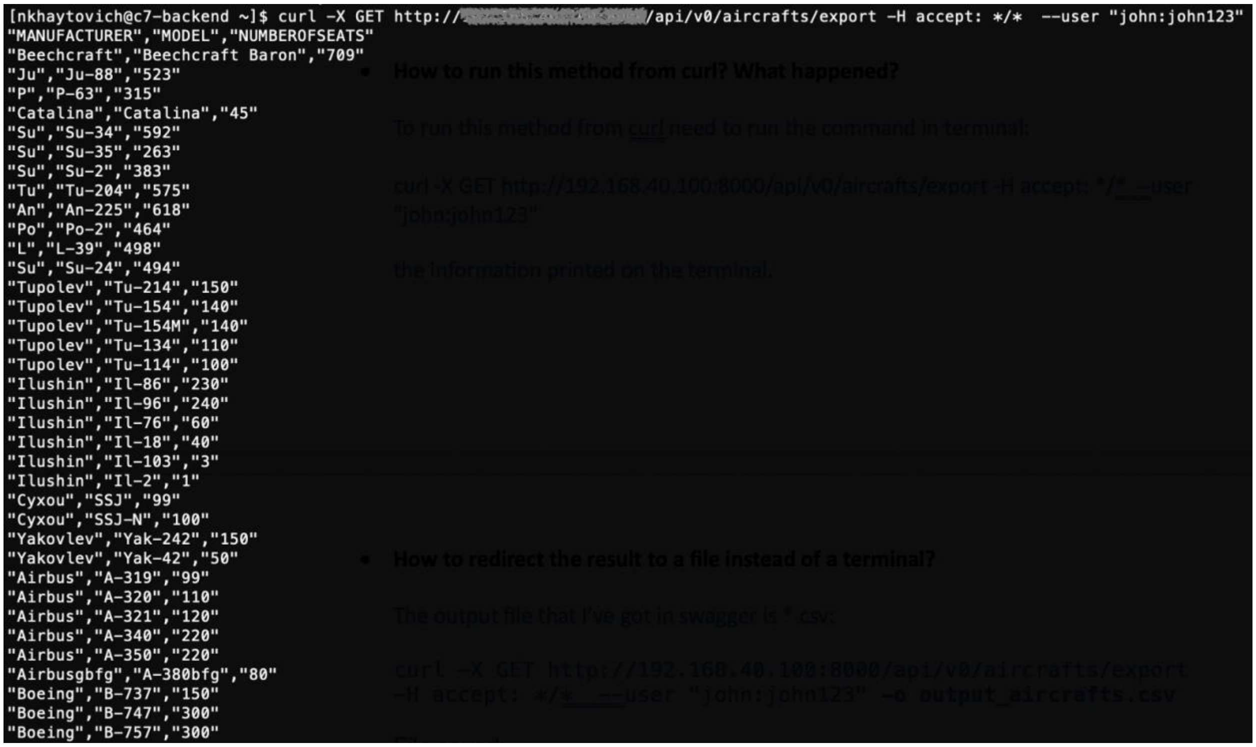
Responses

- **How to run this method from curl? What happened?**

To run this method from curl need to run the command in terminal:

curl -X GET http://***.***.***.***:****/api/v0/aircrafts/export -H accept: */* --user "john:john123"

the information printed on the terminal:



```
[nkhaytovich@c7-backend ~]$ curl -X GET http://[redacted]/api/v0/aircrafts/export -H accept: */*  --user "john:john123"
"MANUFACTURER","MODEL","NUMBEROFSEATS"
"Beechcraft","Beechcraft Baron","709"
"Ju","Ju-88","523"
"P","P-63","315"
"Catalina","Catalina","45"
"Su","Su-34","592"
"Su","Su-35","263"
"Su","Su-2","383"
"Tu","Tu-204","575"
"An","An-225","618"
"Po","Po-2","464"
"L","L-39","498"
"Su","Su-24","494"
"Tupolev","Tu-214","150"
"Tupolev","Tu-154","140"
"Tupolev","Tu-154M","140"
"Tupolev","Tu-134","110"
"Tupolev","Tu-114","100"
"Ilushin","Il-86","230"
"Ilushin","Il-96","240"
"Ilushin","Il-76","60"
"Ilushin","Il-18","40"
"Ilushin","Il-103","3"
"Ilushin","Il-2","1"
"Cyxou","SSJ","99"
"Cyxou","SSJ-N","100"
"Yakovlev","Yak-242","150"
"Yakovlev","Yak-42","50"
"Airbus","A-319","99"
"Airbus","A-320","110"
"Airbus","A-321","120"
"Airbus","A-340","220"
"Airbus","A-350","220"
"Airbusgbfg","A-380bfg","80"
"Boeing","B-737","150"
"Boeing","B-747","300"
"Boeing","B-757","300"
```

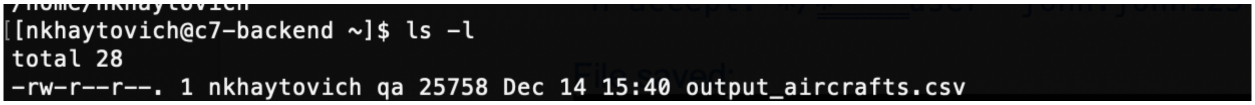
- **How to redirect the result to a file instead of a terminal?**

  The output file that I've got in swagger is *.csv:

  curl -X GET http:// ***.***.***.***:****/api/v0/aircrafts/export -H accept: */* --user

  "john:john123" **-o output_aircrafts.csv**

  File saved:



```
/home/nkhaytovich
[nkhaytovich@c7-backend ~]$ ls -l
total 28
-rw-r--r--. 1 nkhaytovich qa 25758 Dec 14 15:40 output_aircrafts.csv
```

- **Say, we need to see an empty result to check how the method works in this case. What preparation needs to be done in the system?**
  To see an empty result, we need to delete all the records about aircrafts from the system.

Part 2. Explain the components of URI:

- **https://john.doe@www.example.com:8080/forum/questions/?tag=networking&order=newest#top**

  https:// - protocol
  john.doe – I think it is the username or username and password
  www.example.com - host
  8080 - port
  /forum/questions/ - path
  tag=networking&order=newest#top - parameters

- **telnet://192.0.5.105:23/**
  telnet:// - protocol
  192.0.5.105 - host
  23 - port

Part 3* (optional for 100 points): Explain each component of the following command:
curl -i -H 'X-USER-IDENTITY-DOMAIN-NAME: OAuthTestTenant125' -H 'Authorization: Basic MzAzYTI0OTItZDY0Zi00ZTA0LWI3OGYtYjQzMzAwNDczMTJiOll5Sk5NSkdFc0ZqUkxWZVZsdVMz' -H 'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' --request POST https://<idm-domain>.identity.<data-center>.oraclecloud.com/oauth/tokens -d 'grant_type=password &username=tenantAdminUser &password=Fusionapps1&scope=http://www.example.com'

curl - command line tool for data transferring across the Internet.

-i - Include protocol headers in the output (H/F).

-H 'X-USER-IDENTITY-DOMAIN-NAME: OAuthTestTenant125' – Header. Here **X-USER-IDENTITY-DOMAIN-NAME** is parameter which used in the access token request. The name of the identity domain is **OAuthTestTenant125.**

-H 'Authorization: Basic MzAzYTI0OTItZDY0Zi00ZTA0LWI3OGYtYjQzMzAwNDczMTJiOll5Sk5NSkdFc0ZqUkxWZVZsdVMz' – Authorization Header. Basic authentication to pass to server. Some token.

-H 'Content-Type: application/x-www-form-urlencoded;charset=UTF-8' – Header. The Content-Type representation header indicates the original media type which is **application**, the application name is **x-www-form-urlencoded**, and it's template is *https://www.iana.org/assignments/media-types/application/x-www-form-urlencoded*
charset=UTF-8 - Tells the client the server's preferred encoding scheme when submitting a username and password. The only allowed value is the case-insensitive string "UTF-8".

--request POST – the HTTP method POST is sent. It shows that need to create something on the server side.

https://<idm-domain>.identity.<data-center>.oraclecloud.com/oauth/tokens - for this URL method POST is sent. I think here a server which provides tokens.

-d 'grant_type=password &username=tenantAdminUser &password=Fusionapps1&scope=http://www.example.com' – some data for authorization. "Grant type" refers to the way an application gets an access token, here grant type is password. Here **username=tenantAdminUser,**
password=Fusionapps1.
We see also **scope** which means that application http://www.example.com is requesting a token with limited scope, it provides the requested scopes here.