Abstract

This is a microservice created as coursework for GoSchool’s Go Microservices 1

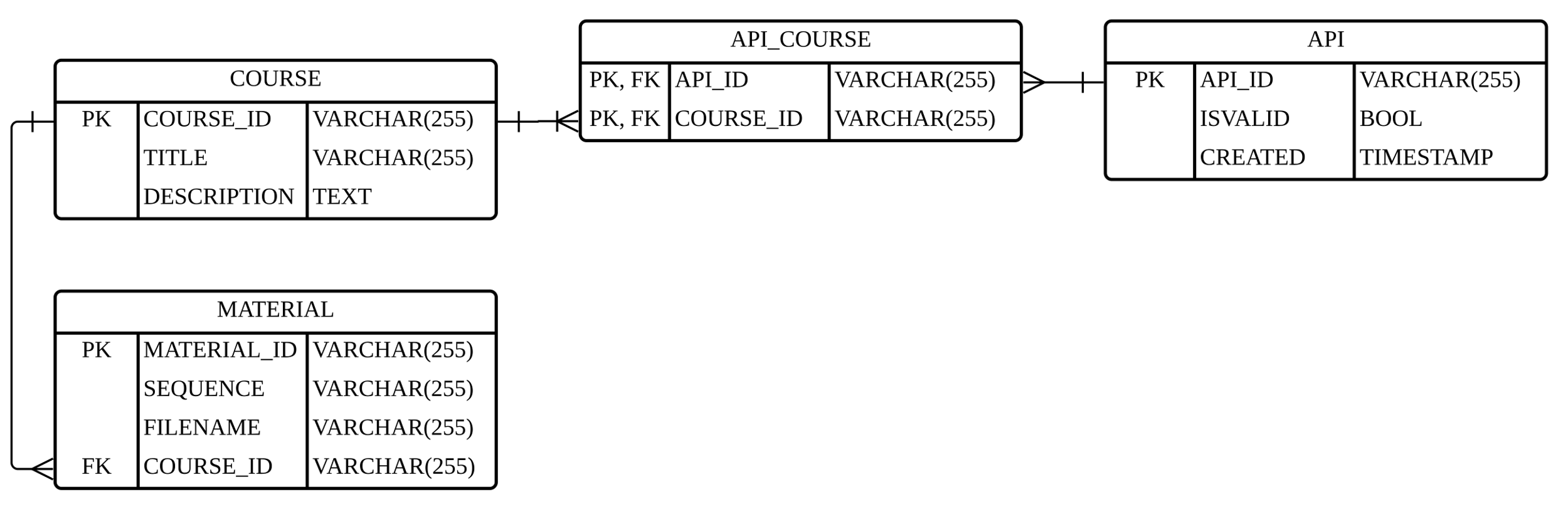
COurse Information microservice

By Calvin Chew Ing Hao

# System Components

## Database

The system uses a containerized MySQL database. Below is the ER diagram to show the database table structure.



The Foreign Keys referencing COURSE(COURSE\_ID ) are set such that on delete of COURSE(COURSE\_ID), the operation will cascade down to delete both API\_COURSE(COURSE\_ID) and MATERIAL(COURSE\_ID).

Similarly, deletion of API(API\_ID) will cascade down to delete API\_COURSE(COURSE\_ID).

Updates of the FK are set to restrict.

The relevant SQL Lines are:

CREATE TABLE API\_COURSE (

FOREIGN KEY (COURSE\_ID) REFERENCES COURSE(COURSE\_ID) ON DELETE CASCADE ON UPDATE RESTRICT,

FOREIGN KEY (API\_ID) REFERENCES API(API\_ID) ON DELETE CASCADE ON UPDATE RESTRICT

…

CREATE TABLE MATERIAL (

FOREIGN KEY (COURSE\_ID) REFERENCES COURSE(COURSE\_ID) ON DELETE CASCADE ON UPDATE RESTRICT

…

A minimal sample of 3 lines of data are added to the tables for demo purpose:

1. A default API key is added to API table and set to ISVALID to true.
2. A sample course information is added to COURSE table.
3. The API Key is associated with the sample course in the API\_COURSE table.

MATERIAL table only contains the file metadata. The file binary itself is stored in the server filesystem itself (download folder).

## Server

The Server container serves the following routes using http.ListenAndServeTLS:

* GET / - Return this page.
* GET /v1 - Return this page.
* GET /v1/courses – Return a list of all courses.
* POST /v1/courses – Add a new course.
* GET /v1/courses/{code} – Return the specified course.
* PUT /v1/courses/{code} – Update the specified course.
* DELETE /v1/courses/{code} – Delete the specified course.
* POST /v1/courses/{code}/materials – Add a new material to the specified course.
* GET /v1/courses/{code}/materials/metadata – Return metadata of all materials of the specified course.
* GET /v1/courses/{code}/materials/files – Download the specified material.
* PUT /v1/courses/{code}/materials/{id} – Update the specified material.
* DELETE /v1/courses/{code}/materials/{id} – Delete the specified material.
* GET /v1/courses/{code}/materials/{id}/metadata – Return the specified material.
* GET /v1/courses/{code}/materials/{id}/files – Download the specified material.

An authenticationMiddleware function checks for incoming API-keys and does a general check against the API table to see if the API-key is authorized to use the APIs. If the user attempts to access a specific course or material associated with a course, a more granular check is done against the API\_COURSE table on a per course basis.

The SSL certificates are self-signed and generated using a Docker Golang container image using command:

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ssl/key.pem -out ssl/cert.pem -subj "/C=SG/ST=Singapore/L=Singapore/O=CalvinChew/CN=server/" -addext "subjectAltName=DNS:server"

Thereafter they are placed in the ssl folder for the program to use and recopied when new containers are spun.

SQL calls are protected from SQL injection by the use of “? “placeholder parameters.

Upon receiving material in a POST or PUT request, after writing the metadata to the MATERIAL table, the file binary is stored in filesystem using the path:

download/{code} /{sequence}/filename

Thereafter the download/{code} folder is zipped up and the zip file is later used by http.ServeFile for download GETs.

Sequence is an integer that represents the order of the materials.

## Client

The Client container provides following options to call the Server container in order to demonstrate REST operations.

Courses Microservice Client:

1. Add a new course.

2. Add a new material to the specified course.

3. Update a specified course.

4. Update a specified material of a specified course.

5. Delete a specified course.

6. Delete a specified material.

7. Return a list of all courses.

8. Return a specified course.

9. Return metadata of all materials of the specified course.

10. Return metadata of a specified material.

11. Download files of all materials of the specified course.

12. Download file of a specified material.

Select Option: \_

To add or update a new material, the files have to be stored in the clients’ upload folder.

For demo purposes, there are two files currently in the upload folder:

1. sample1.txt
2. sample2.txt

Hence, when prompted for Material Filename, you should enter “sample.txt”.

Files that are downloaded are stored in the Client’s download folder.

The Server container’s SSL certificate is copied to the Client container’s ssl folder and appended to x509.NewCertPool(), with TLSClientConfig configured such that the client is able to accept the Server’s self-signed certificate and make https calls.

The Client will send the API-Key in the X-API-KEY header for the Server to authenticate against.

# Dockerfiles and docker-compose.yml

The docker-compose.yml contains the account names and passwords for the database as well as the API-Key for the client, so keep this file in a safe place.

The docker-compose.yml sets the environment variable XAPIKEY so that the client container can use it to send to the server for it to authenticate against.

The docker-compose.yml also sets the environment variable PLACE for the server and client so that depending on whether the program is run in a docker container or the developer’s machine, the program is able to switch to the correct URI to use.

Dockerfile.client and Dockerfile.server are referenced in the docker-compose.yml and used to set up the client and server containers. The server folder is copied to the server container WORKDIR and the client folder is copied to the client container WORKDIR.

This means that the server container is only able to see the files in the server and not the other files such as docker-compose.yml.

The db folder is attached to the db container as a volume in /docker-entrypoint-initdb.d. The db folder contains db.sql.gz, which is an export of the MySQL database. This means that upon spinning up the db container, MySQL will automatically load the db.sql.gz file and the database is now ready.

# Setup Guide

A prerequisite is to have Docker and docker-compose installed.

The containers will use the ports 3306, 5000, and 8080, so make sure those ports are not used by other apps.

To deploy the system, after unzipping the project zipped file, navigate inside the project folder and run the following command:

docker-compose up -d --build

The command will use the docker-compose.yml to spin up 4 containers – adminer (GUI to manage the database), server, db, and client.

Text

Description automatically generated

Once it is done, run the command:

docker exec -it goms1\_client\_1 ./main

This will run the main program inside the client container.

# Testing Guide

Text

Description automatically generated

Outside of the client main program, navigate your web browser to:

http://localhost:8080/

And use the user credentials stored in the docker-compose.yml file to login.

Graphical user interface, application

Description automatically generated

You can use this Adminer GUI interface to check the data stored in MySQL database when operating the client program.

## Option 1, Add a new course.

In the client main program, press “1”, press enter, and then key in as follows:

Text

Description automatically generated

In the Adminer GUI, click on the select next to COURSE (highlighted in red box)

Graphical user interface

Description automatically generated

Notice that a new row is added successfully:

Graphical user interface, application

Description automatically generated

Because you are the one to add a new course, a new row is added to API\_COURSE, indicating your API-Key is now able to access course TIC1002 as well.

Graphical user interface, application

Description automatically generated

## Option 2. Add a new material to the specified course.

In the client main program, press “2”, press enter, and then key in as follows:

Text

Description automatically generated

Notice that the material metadata has been successfully added. We will verify the file binary later in the test.

Graphical user interface, application

Description automatically generated

## Option 3. Update a specified course.

In the client main program, press “3”, press enter, and then key in as follows:

Text

Description automatically generated

Notice the course details have been updated.

Graphical user interface, application

Description automatically generated

## Option 4. Update a specified material of a specified course.

In the client main program, press “4”, press enter, and then key in as follows:

Text

Description automatically generated

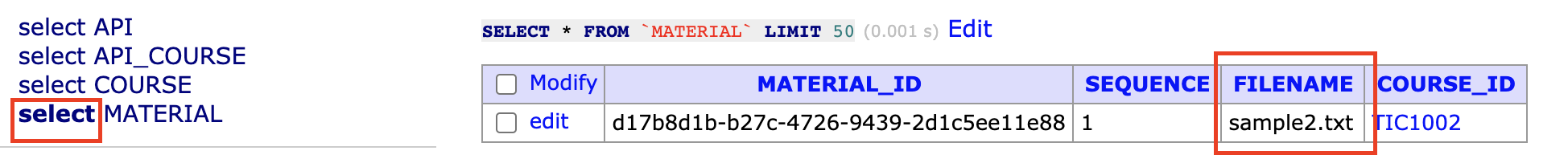
You will need to enter the Material ID. For now, get it from the Adminer GUI, we will test another way later. Graphical user interface, application

Description automatically generated

Text

Description automatically generated

Notice that FILENAME has changed to sample2.txt just like we specified.



## Option 5. Delete a specified course.

In the client main program, press “5”, press enter, and then key in as follows:

Text

Description automatically generated

Note that the course TIC1001 is now gone.

Table

Description automatically generated with low confidence

Because the course is deleted, the delete cascades to API\_COURSE. Notice that API\_COURSE no longer contains the entry for COURSE\_ID TIC1001. It is the same for the associated material in MATERIAL table (although it does not contain an associated material in the first place)

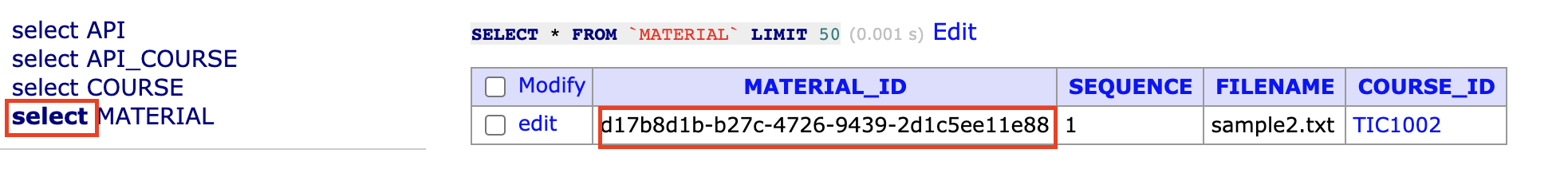
A picture containing graphical user interface

Description automatically generated

## Option 6. Delete a specified material.

In the client main program, press “6”, press enter, and then key in as follows:

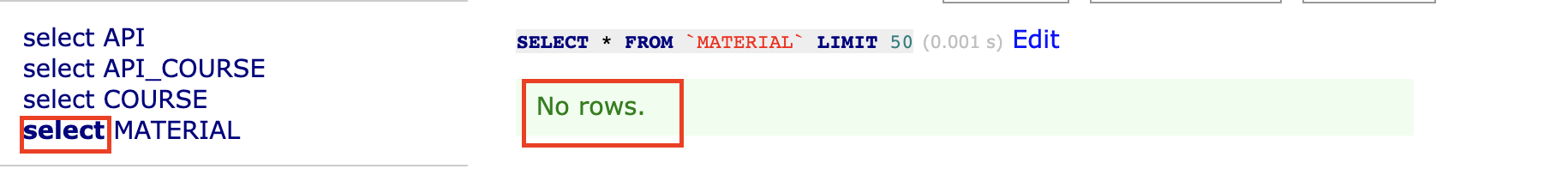
Unfortunately we have to get the material id from Adminer GUI again.



Text

Description automatically generated

But you can notice that the table is empty now.



**Let us redo press 1. Add a new course first before we continue.**

Text

Description automatically generated

## Option 7. Return a list of all courses.

In the client main program, press “7”, press enter.

You can now see all courses, include the new one we just added.

Text

Description automatically generated

It matches with Adminer GUI

Table

Description automatically generated with medium confidence

## Option 8. Return a specified course.

In the client main program, press “8”, press enter, and then key in as follows:

It only returns one specified course as expected.

Text

Description automatically generated

**Let us redo press 2. Add a new material to the specified course before we continue.**

Text

Description automatically generated

## Option 9. Return metadata of all materials of the specified course.

In the client main program, press “9”, press enter, and then key in as follows:

Now you can rely on this option to find out the material id instead of going to Adminer GUI.

Text

Description automatically generated

## Option 10. Return metadata of a specified material.

In the client main program, press “10”, press enter, and then key in as follows:

If you try to find a material id that does not exist, you will get a 404 not found.

Text

Description automatically generated

## Option 11. Download files of all materials of the specified course.

In the client main program, press “11”, press enter, and then key in as follows:

You will find a zip file containing all materials associated with TIC1003 in the download folder. More instructions on how to view it will follow.

Text, email

Description automatically generated

## Option 12. Download file of a specified material.

In the client main program, press “12”, press enter, and then key in as follows:

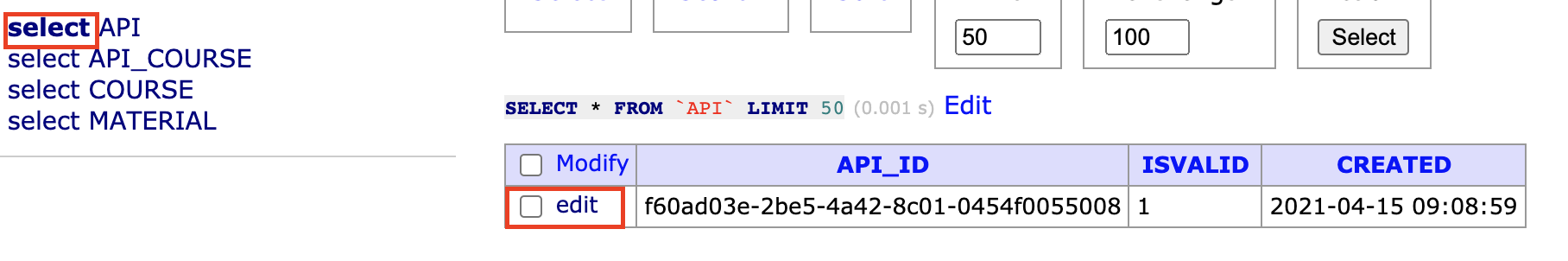
You can redo Option 9 - return metadata of all materials of the specified course, in order to get a valid material id. Then try option 12 to download it.

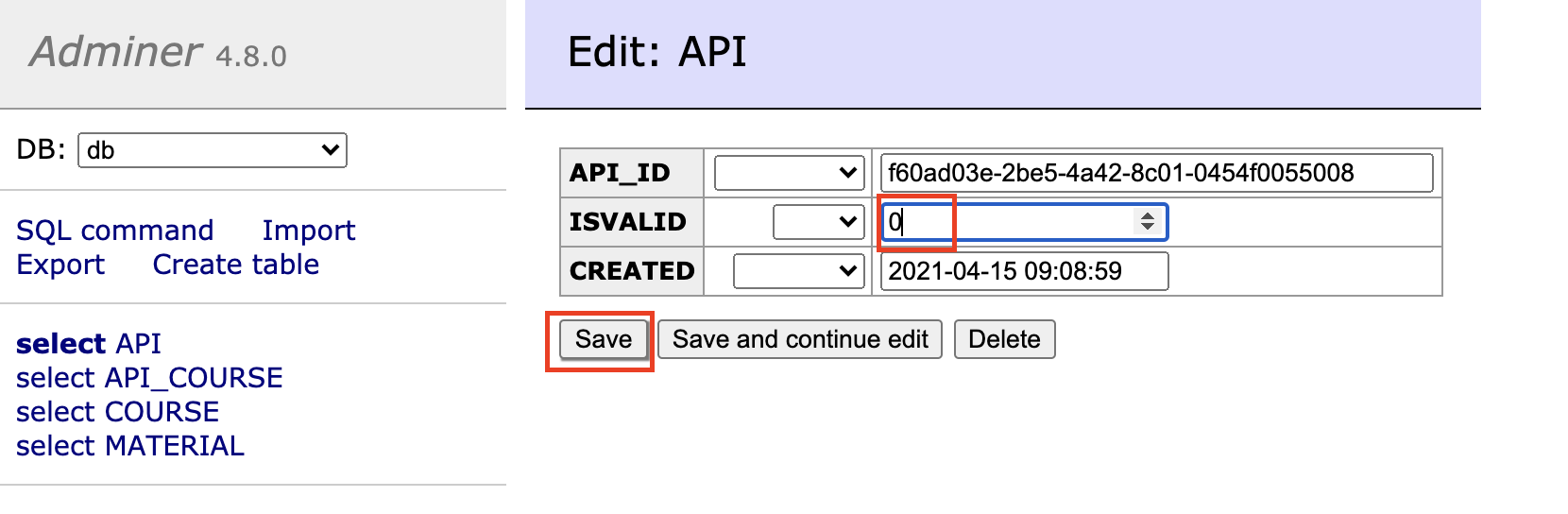
Text

Description automatically generated

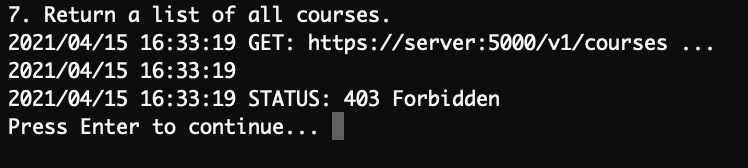
## Testing API-Key

You can test the API-Key security by using Adminer GUI and changing the ISVALID to 0.





After saving, you will not be able to perform options:



If you still wish to continue testing this program, remember to set ISVALID back to 1.

## Viewing the download folder

Press Ctrl-C to break out of the client main program.

Then enter these commands:

docker exec -it goms1\_client\_1 bash

cd download

ls

You can now see that TIC1003.zip and sample1.txt is there.

Text

Description automatically generated