Distributed Systems Programming

Distributed Systems API

|  |
| --- |
|  |
| Word Count: 1098 |
| April 2020  Student Number: 2017071831  Authored by: Onayo Moro |



Contents

[API’s 3](#_Toc38577999)

[What is an API 3](#_Toc38578000)

[Stateless v Stateful 3](#_Toc38578001)

[Route Mapping and Web APIs 3](#_Toc38578002)

[Requests 4](#_Toc38578003)

[Client and Server ApiKey Use 7](#_Toc38578004)

[RSA 7](#_Toc38578005)

[AES 7](#_Toc38578006)

[Entity Framework 7](#_Toc38578007)

[Conclusion 8](#_Toc38578008)

# API’s

## What is an API

An Application Programming Interface (API) is a software interface that offers other software components methods of utilising parts of a computer system. This could be databases, applications or other devices. Essentially, APIs take requests, notify the system of the request and return a response. They define what calls and requests can be made and what functions and components of a computer system can be interfaced with.

## Stateless v Stateful

The server designed for this project uses what is known as a stateless API. This is due to the server forgetting the client after the client has sent a request. Furthermore, an open connection between the client and the server is not maintained. Every time the client makes a new request to the server, it sends over identification which the server validates if it exists within its database. If validated, the server sends back a response and promptly closes the connection and forgets the client. The reason for using a stateless system is this server will only be dealing with a small number of known clients. The server also remembers and stores the details of its client’s requests in a database.

The difference between stateful and stateless is that a stateful system maintains the connection between the client and the server. The server may also remember the client’s information such as their identity and request. Stateful systems may also be less computationally heavy as there is no constant database verification between requests.

# Route Mapping and Web APIs

Route mapping refers to the structure of the URI sent to the server. The Web API requires the URI to define which particular controller and action it wants to access within its structure. Therefore, the client must format its resource identifiers in the URI before sending it over.

Figure 1: Web API URI format

Figure 1 shows how the URI format accepted by the Web API.

# Requests

In this project, GET, POST and DELETE protocols were used to communicate with the server.

A GET request is used when the client is requesting a resource from the server, for example, data from the server’s database.

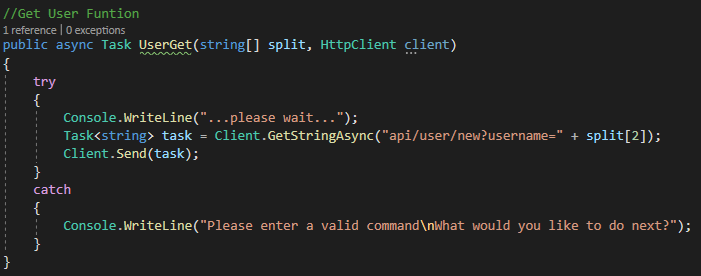


Figure 2: User Get Client Function

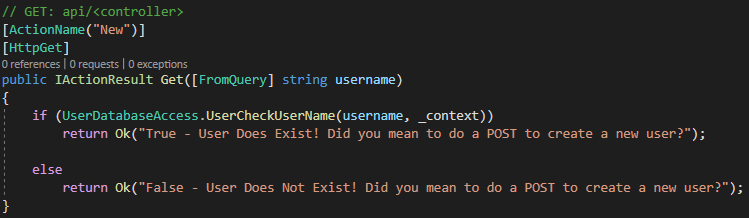


Figure 3: User Get Server Function

Figure 2 shows how the a GET request is used to ask the server if a user exists. Whereas figure 3 shows how the server handles that request.

A POST request is used when the client asks the server to accept whatever data is enclosed in the request. This data could be sent through the header or the body of the request.

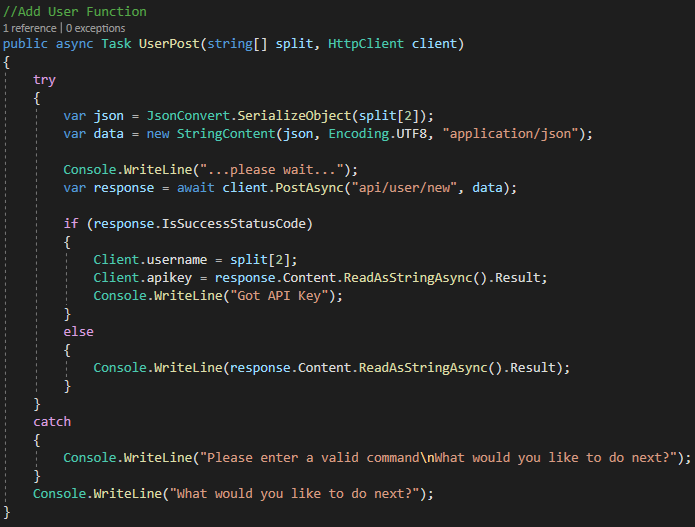


Figure 4: User Create Post Client Function

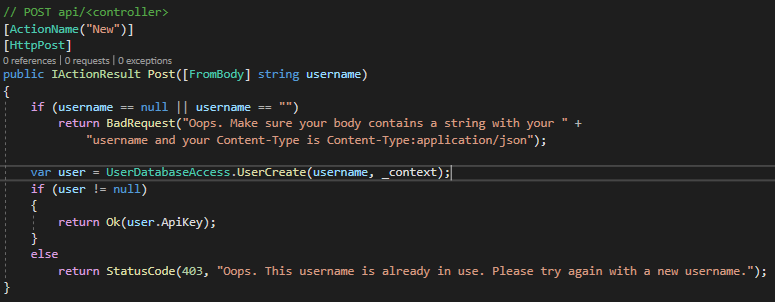


Figure 5: User Create Post Server Function

Figure 4 is the code for the clients User Post function which asks the server to create a new user in its database. It does this by sending the user information in a json object through the URI’s body. Figure 5 however shows how the server handles the POST request and json object in the body.

A DELETE request is made when the client requests that the server deletes a resource identified within the URI.

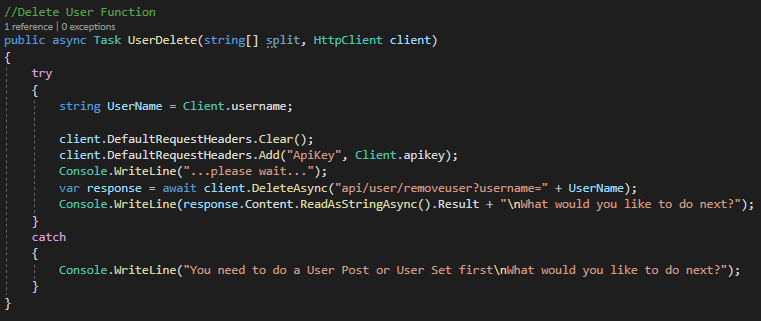


Figure 6: User Delete Client Function



Figure 7: User Delete Server Function

Figure 6 shows how the client adds data to the header and sends a DELETE request to the server. Figure 7 shows how the server pulls the data from the header and processes the DELETE request.

# Client and Server ApiKey Use

In the context of this project, the user’s ApiKey is sent within the header of the request URI. The server then uses the ApiKey to authorise the request. If a user associated with the ApiKey exists and has the valid role, access is granted to the action requested in the URI.

The ApiKey as a means of authentication may be secure as the ApiKey given by the client is not stored and destroyed when the request is processed.

Although, due to the lack of hashing of communications, the ApiKey would be readable if intercepted. Therefore, in this case, the authentication implementation is unsecure.

In the real world, it would be imperative to hash all communications between the server and the client to add security against data being intercepted.

# RSA

RSA is a cryptosystem that makes use of a public and private key.

1. Two random prime numbers, and , are computed such that .
2. Next, compute the totient
3. Then choose co-prime to
4. is then chosen to satisfy

The public key is . For a padded message , it is .

The private key is . The decryption function is .

# AES

AES is a block cipher that uses 128, 192 or 256 bit key and block sizes. There are ten to fourteen rounds of encryption with one round consisting of four steps: SubBytes, Shiftrows, MixColumns and AddRoundKey.

* SubBytes is a mathematically calculated substitution matrix with the inverse substitution applied on a per-bit basis.
* ShiftRows prevents columns existing as smaller blocks which can be independently reversed.
* MixColumns takes the input columns to create a different output. Diffusion is then added by seeing if any bits in the plaintext change, then on average, half of the computed cipher should change.
* AddRoundKey then generates a round key based on the symmetric key using XOR to produce the output.

# Entity Framework

Entity Framework, developed by Microsoft, is a framework for SQL Databases. It offers the functionality of converting objects to table entries without the use of SQL queries. Code First targets a database that does not yet exist and creates it. It can also add tables to existing databases. Model First also works if the database does not exist. A model is saved in an EDMX file which can be defined in Entity Framework’s designer. A database schema can then be generated using SQL. Database First creates model codes (DbContext) from the database and those classes are used to link the database and the controller. Unlike Code and Model First, the entity framework can only be created using an existing database.

Migrations are a way to update a databases schema. They allow the creation of an initial database. They are also a way to update changes to a database as well as keep track of Entity Framework models.

# Conclusion

All tasks for this project were able to be completed. However, there were hurdles in understanding the communication between the different systems including understanding how to use HTTP Client and HTTP responses. These problems were often overcome by reading official documentation and testing any examples provided.