**Distributed Systems Programming 600089**

Distributed Systems API: Report

Student Number: 201601628

Submitted April 2020

Words: 1000

# 1. APIs

Outline what an API is and does, how it manages requests and discuss why this API is stateless and describe the difference between a stateless and stateful server.

<https://www.howtogeek.com/343877/what-is-an-api/>

Think of an API like a menu in a restaurant. The menu provides a list of dishes you can order, along with a description of each dish. When you specify what menu items you want, the restaurant’s kitchen does the work and provides you with some finished dishes. You don’t know exactly how the restaurant prepares that food, and you don’t really need to.

Similarly, an API lists a bunch of operations that developers can use, along with a description of what they do. The developer doesn’t necessarily need to know how, for example, an operating system builds and presents a “Save As” dialog box. They just need to know that it’s available for use in their app.

If you want to capture photos or video from the iPhone’s camera, you don’t have to write your own camera interface. You use the camera API to embed the iPhone’s built-in camera in your app. If APIs didn’t exist to make this easy, app developers would have to create their own camera software and interpret the camera hardware’s inputs. But Apple’s operating system developers have done all this hard work so the developers can just use the camera API to embed a camera, and then get on with building their app. And, when Apple improves the camera API, all the apps that rely on it will take advantage of that improvement automatically.

APIs are also used to control access to hardware devices and software functions that an application may not necessarily have permission to use. That’s why APIs often play a big role in security.

 asking to see your precise location, that website is attempting to use the geolocation API in your web browser.

However, browsers also expose this information via an API because it’s possible to control access to it.

APIs are used for all kinds of other reasons, too. For example, if you’ve ever seen a Google Maps object embedded on a website, that website is using the Google Maps API to embed that map. Google exposes APIs like this to web developers, who can then use the APIs to plop complex objects right on their website. If APIs like this didn’t exist, developers might have to create their own maps and provide their own map data just to put a little interactive map on a website.

The OAuth standard also defines a number of APIs that allow you to sign into a website with another service—for example, to use your Facebook, Google, or Twitter accounts to sign into a new website without creating a new user account just for that site. APIs are standard contracts that define how developers communicate with a service, and the kind of output those developers should expect to receive back.

# 2. Route Mapping in WebAPI

Briefly explain what route mapping is, how WebAPI uses the id parameter and what actions are.

<https://docs.microsoft.com/en-us/aspnet/web-api/overview/web-api-routing-and-actions/routing-in-aspnet-web-api>

# 3. HTTP Requests

Briefly outline what GET, POST and DELETE requests are and provide screenshots of where you have used these requests in your server project to illustrate your written work.

# 4. API Keys

In the created API, methods requiring an authenticated user expect to receive an API key in the header of the request, which is then searched for in the database for verification. Briefly describe how your Server and Client use the API key. Identify if you think an API key is a good or bad option for identifying users, giving your reasons. Is the API key safe in this project? How would you ensure this API key was kept safe if you were developing this Server/Client in the ‘real world’?

# 5. The RSA Algorithm

Outline the steps in the RSA algorithm. IN BULLET POINTS

<https://www.tutorialspoint.com/cryptography_with_python/cryptography_with_python_understanding_rsa_algorithm.htm>

# 6. The AES Algorithm

Outline the steps in the AES algorithm. IN BULLET POINTS

<https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm>

# 7. Entity Framework

Briefly describe what the Entity Framework is and what it does.

<https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

Entity framework does things…

Compare code first, model first and database first techniques and describe what a migration is/does.

There are several documented approaches to developing applications with Entity Framework:

* Code First
* Model First
* Database First

The suitability of each approach largely depends on the scope of the project and the information available at the time of development.

The code first method allows the developer to focus on development of backend code without concern for database design, which is instead managed by Entity Framework. Through a variety of decorators and conventions (e.g. “[Key]” or PrimaryKeyId) and the manual creation of “context” simply to define the required collections to be stored, EF can generate any number of SQL databases and, likewise, tables within them.

On the other hand, the model first approach requires that…

Finally, in database first…

# 8. Reflections

Finally, write a short reflective statement about which tasks you completed and to what level, any problems you had with any of the functionality and how you overcame these problems (if you managed to).