Http protocol & RESTful API

A brief overview

jimmg35

What is HTTP protocol?

HTTP stands for **H**yper**T**ext **T**ransfer **P**rotocol

A protocol widely used between Client and server side

Then what is HTTPS?

The S stands for Secure

The content of packect will be encrypted

You can think of it as an extension of HTTP

Simply put, we have a target called **URI**

Uniform **R**esource **I**dentifier

Something looks like this…

https://**www.geo.ntnu.edu.tw**/**index.php/kuo-chen-chang**/

https://**www.geo.ntnu.edu.tw**/**index.php/kuo-chen-chang**/

This URI specify what kind of protocol we’re using.

The Domain name of the server.

And the relative path of resource on that server.

https://www.geo.ntnu.edu.tw/index.php/kuo-chen-chang/ **>> Path to Harry**

https://www.geo.ntnu.edu.tw/index.php/chun-chia-chang/ **>> Path to AKA**

There exists such uniform patterns of URI.

Depends on how backend developer designs it.

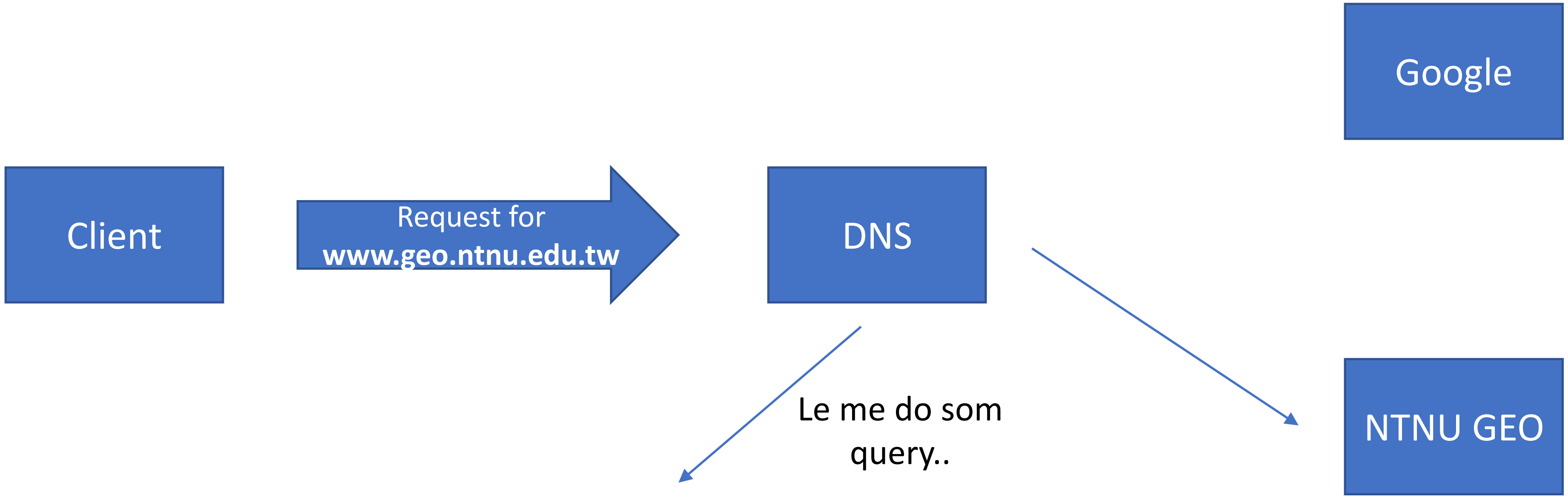
Just before we get start, something need to be clarified…

www.geo.ntnu.edu.tw

The domain name is an image of IP address
ends with specific port number.

So why the browser can identify **www.geo.ntnu.edu.tw**?

Because of the Domain name server(DNS)!



www.geo.ntnu.edu.tw	140.122.64.119
www.google.com	8.8.8.8

Domain Hijacking?

From now, we have gained some basic knowledge of our target.

Let’s see what kind of Action we can take!

Several methods that comes in handy…

GET

HEAD

POST

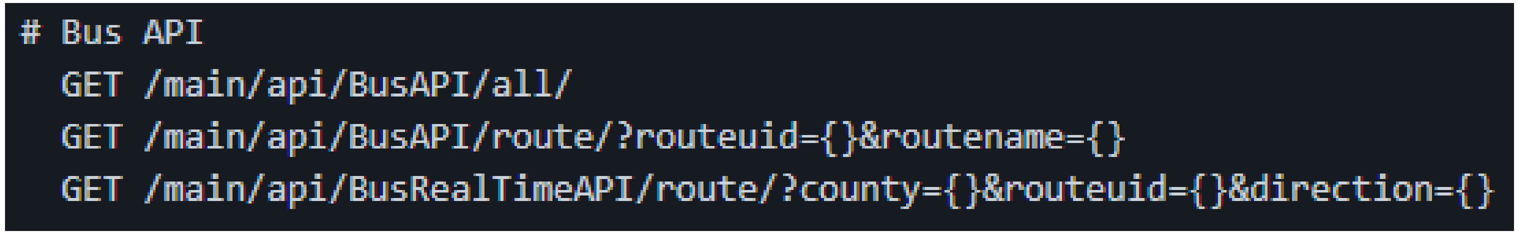
PUT

DELETE

Whenever we enter a web page, we’re using **GET method** to achieve this.

GET will query the resources with parameters attached after the URI

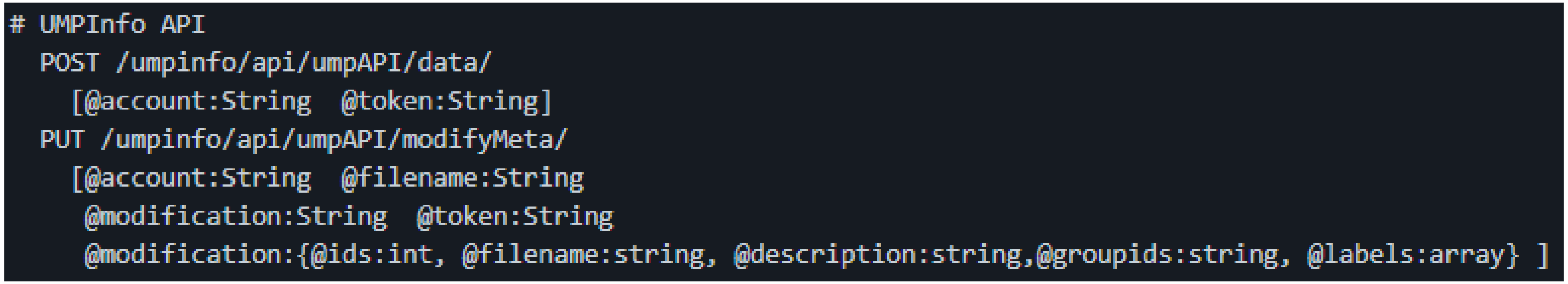
Here’s some example of GET method!



```
# Bus API
  GET /main/api/BusAPI/all/
  GET /main/api/BusAPI/route/?routeuid={}&routename={}
  GET /main/api/BusRealTimeAPI/route/?county={}&routeuid={}&direction={}
```

Github ~ jimmg35/GIAP_Backend

Whenever we want to create new data on server side, we use **POST method**.



```
# UMPInfo API
  POST /umpinfo/api/umpAPI/data/
    [@account:String  @token:String]
  PUT /umpinfo/api/umpAPI/modifyMeta/
    [@account:String  @filename:String
     @modification:String  @token:String
     @modification:{@ids:int, @filename:string, @description:string,@groupids:string, @labels:array} ]
```

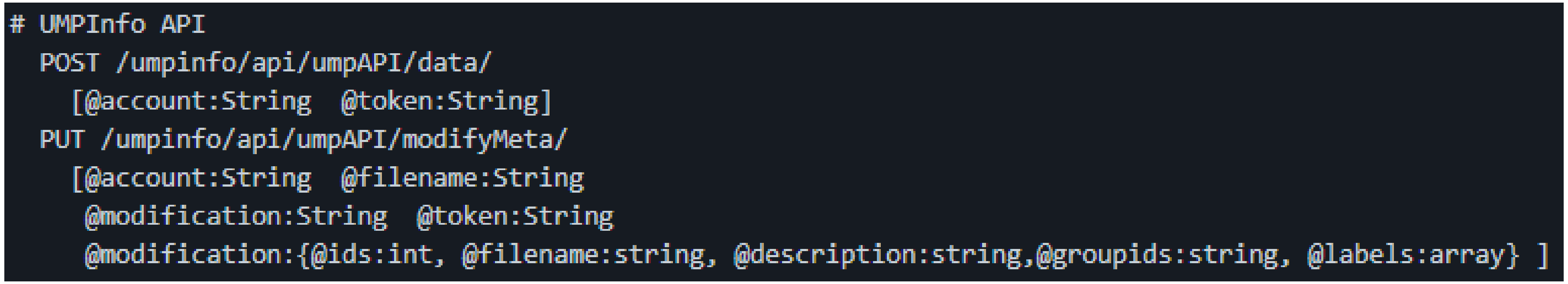
So what method we use whenever we want to login?

POST!!

Since we treat our account and password as parameters,
GET method will attach your password to the end of URI!!

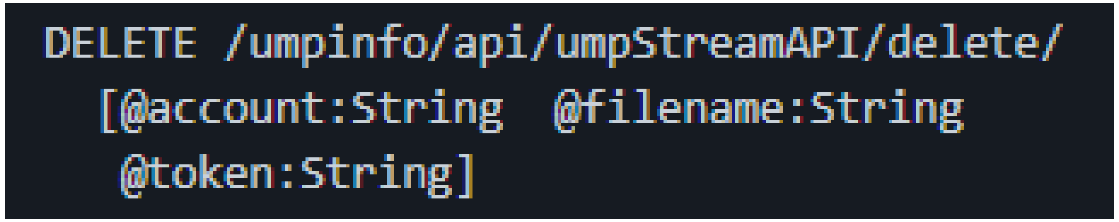
POST method will hide those information in the body!
Which is more secure!

Whenever we want to modify data on server side, we use **PUT method**.



```
# UMPInfo API
  POST /umpinfo/api/umpAPI/data/
    [@account:String  @token:String]
  PUT /umpinfo/api/umpAPI/modifyMeta/
    [@account:String  @filename:String
     @modification:String  @token:String
     @modification:{@ids:int, @filename:string, @description:string,@groupids:string, @labels:array} ]
```

Whenever we want to delete data on server side, we use delete.



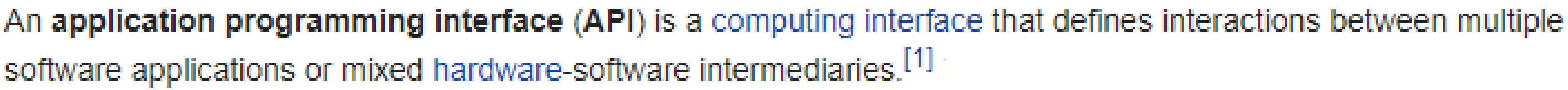
```
DELETE /umpinfo/api/umpStreamAPI/delete/
  [@account:String  @filename:String
   @token:String]
```

We have introduced several popular HTTP method…

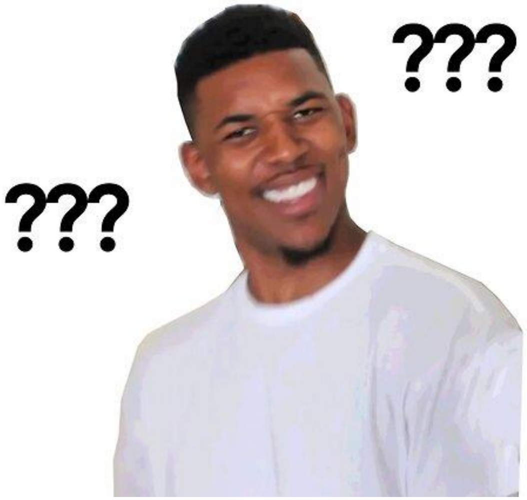
So what is RESTful API?

before we answer this question, we must
ask ourselves, what is API?

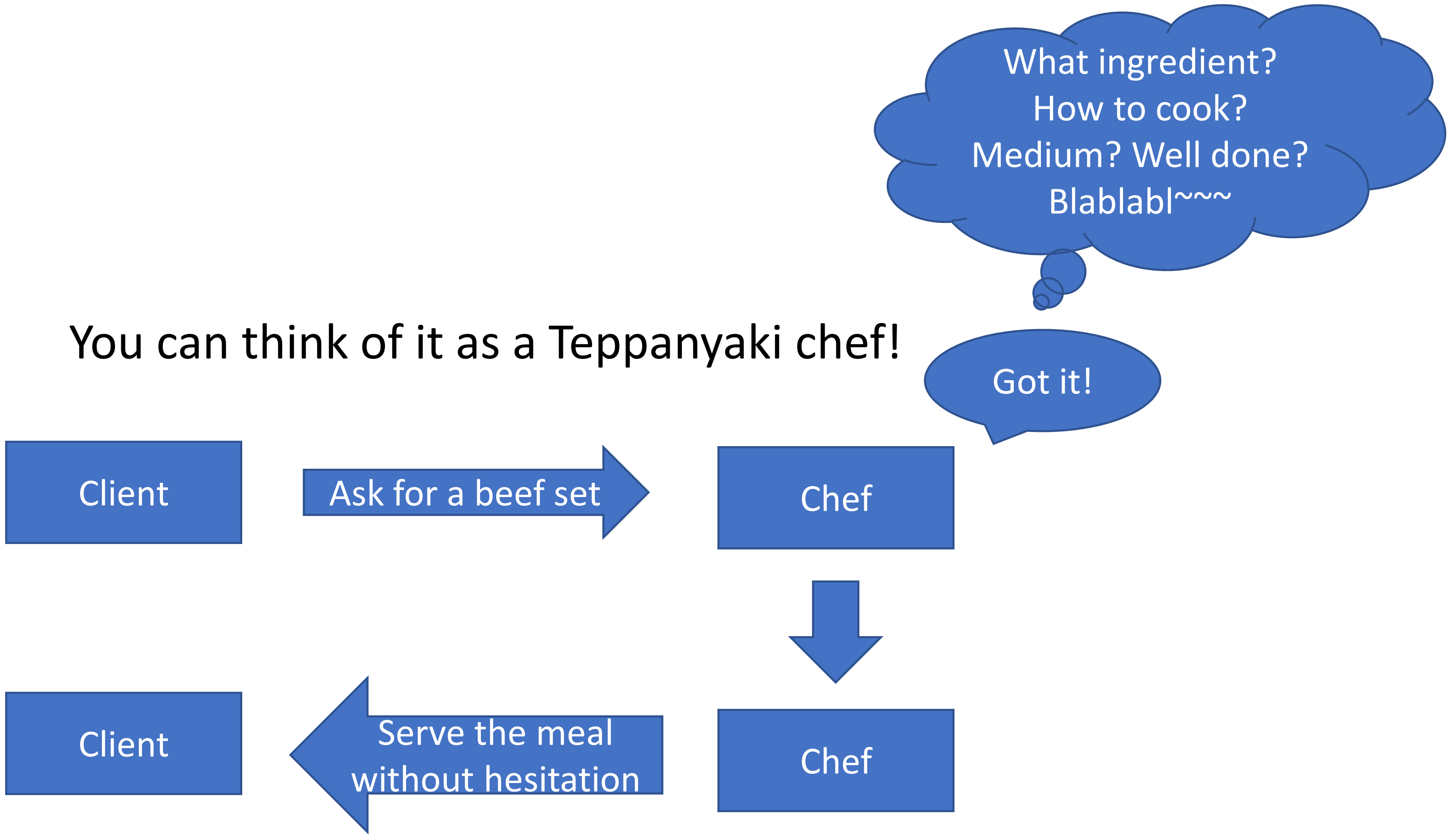
According to Wiki…



An **application programming interface** (**API**) is a computing interface that defines interactions between multiple software applications or mixed hardware-software intermediaries.[1]



You can think of it as a Teppanyaki chef!



What ingredient?
How to cook?
Medium? Well done?
Blablabl~~~

Got it!

Client

Ask for a beef set

Chef

Chef

Serve the meal without hesitation

Client

The client doesn’t need to have knowledges before having a Teppanyaki meal!

The only thing that clients need to manage is communication with the chef!

That is exactly how API works!

You just need to figure out how to use it!
No need for underlying knowledges!

Several features of RESTful API:

1. Uniform Interface
2. Stateless
3. Cacheable
4. Layered System
5. Code-On-Demand

Make API request using Postman!

Download Postman | Try Postman for Free

https://www.postman.com/downloads/

Make API request using Python!
Using python requests module!