**1.REST API->**

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Body** |
| **1** | **Rest** | **…** |
| **2** | **Ajax** | **…** |
| **3** | **Json** | **…** |
|  |  |  |

기계와 기계가 통신규약인 Http를 이용한 통신 방안 REST API  
REST API는 내 컴퓨터가 아닌 남의 컴퓨터를 실행시킨다. Resource는 URI를 포함하고 있음. 예를 들어 <http://example.com/topics>이면 topics가 resource가 된다.   
\* 여기서 주의해야할 부분이 URL(Uniform Resource Locator) 과 URN(Uniform Resource Name) 을 포함한 부분이 URI(Uniform Resource Identifier)이다.

**<-Resource**

Collection -> <http://example.com/topics>

Element-> <http://example.com/topics/1>, <http://example.com/topics/rest>

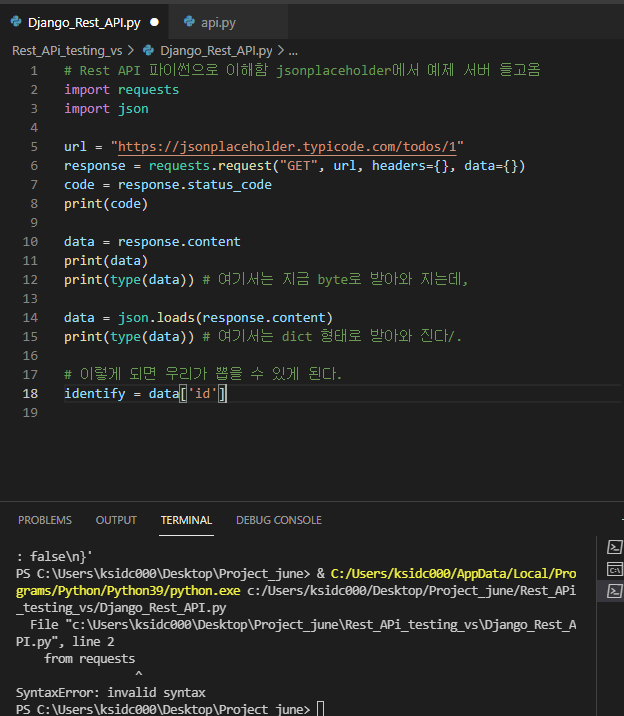
하지만 URI는 정보를 식별하는 이름일 뿐이고, 가공할 수 있어함. Create, Read, Updatd, Delete 의 method를 Rest API 그대로 사용한다.   
(http의 method는 create->post, Read->get, Update->put | patch, Delete->delete)

Rest API를 실행 실험을 하기 위해서, ATOM 개발환경에서 node.js/npm으로 이를 확인하였다.(개발언어 중요하지 않음. 이 부분은 Rest API를 확인하기 위함)

개발 테스팅을 임의적으로 해본 결과 -> 개발자 도구 Network와 preview에서 클라이언트와 서버간의 통신 내용을 확인할 수 있고,  
Rest API에서 클라이언트와 서버가 어떤 데이터 타입으로 상호작용 하는지는 규정하고 있지 않음. Json, xml 뭐든 상관없음. 하지만 웹서버에서 대부분 json으로 받고 있음. 규정하고 있는 부분은 resource를 식별할 때 URI를 통해서 식별한다는 것임. 어떤 행위를 할때는 POST, GET, PUT, DELETE 를 사용하다는 것임. 이것이 REST API 임.

->HTTP프로토콜을 HTTP프로토콜 답게 사용하자는 것이 Rest API가 주장하는 것으로 이해하면 된다.

좀더 구체적으로 Rest API 통신과 JSON의 개념을 얻기 위해서, VS code에서 파이썬으로 실습을 진행하였다.



(\*HTTP란 HyperText Transfer Protocol의 줄임말로 www 상에서 사용하는 프로토콜임.   
HTTP: <https://_______/______>?값\_\_\_ <-이 부분 쿼리 스트링임.)

**GET이라는 방식과 & POST 방식의 차이점 ->**  
GET은 클라이언트에서 서버로 어떠한 리소스로부터 정보를 요청하기 위해 사용되는 메서드임. 예를 들어 게시판의 게시물을 조회할 때 쓸 수 있으며, Get을 통한 요청은 URL 주소 끝에 파라미터로 포함되어 전송되며, 이 부분을 쿼리 스트링(query string)이라고 한다. 방식은 URL 끝에 “?”를 붙이고 그 다음 변수명1=값&변수명2=값..형식으로 이어 붙이면 되는 것임.   
ex) [www.example.com/show?name1=value1&name2=value2](http://www.example.com/show?name1=value1&name2=value2)

서버에서는 name1과 name2 라는 파라미터 명으로 각각 value1과 value2의 파라미터 값을 전달 받을 수 있는 것임.

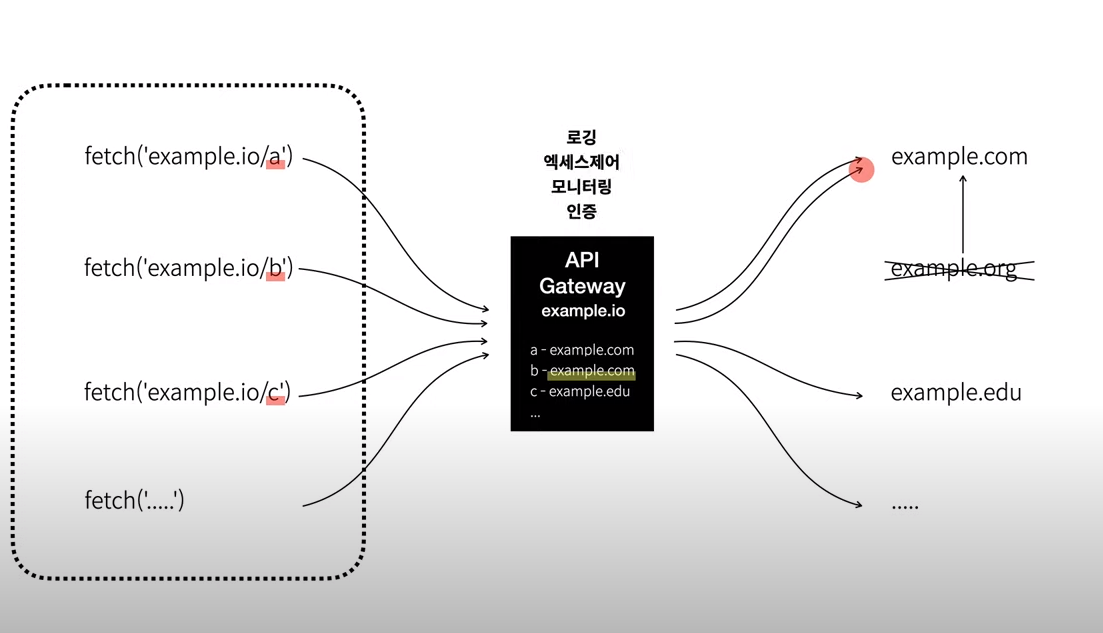
이에 반해서 POST로 데이터를 전송할 때 길이 제한이 따로 없어서 용량이 큰 데이터를 보낼 때 사용하거나 GET처럼 데이터가 외부적으로 드러나는 것은 아니라서 보안이 필요한 부분에 많이 사용된다. (하지만 데이터를 암호화 하지 않으면 body의 데이터도 결국 볼 수 있는건 똑같음.)  
POST는 전송할 데이터를 HTTP 메시지 body 부분에 담아서 서버로 보낸다.(body의 타입은 content-Type 헤더에 따라 결정되는 것임.)

결국 차이는 GET은 서버의 리소스에서 데이터를 요청할 때, POST는 서버의 리소스를 새로 생성하거나 업데이트 할 때 사용된다. DB로 따지면 GET은 SELECT에 가깝고, POST는 Create에 가깝다고 보면 된다.

이후 프로젝트 진행 절차:   
**(1)Python**(language)\_ **django**(framework)\_**dbqlite**(DB)   
**(2)PHP**(language)\_**Lalavel(framework)\_mysql**(DB)  
**(3**)**Java**(language)\_**Spring(framework)\_mysql**(DB) Spring Boot Restful CRUD API 테스팅,   
\*(장고는 DB를 dbqlite3을 쓰는 것으로 알고 있습니다. SQL, Oracle이 무거워서 그렇게 쓰는 것 같습니다.)

**2. API gateway->**

Rest API 와 같은 인터넷 기반의 API를 사용하면 여러 서버들로 API가 분산된다. 각 서버는 도메인이 다른거지, 클라이언트 입장에서는 url=’’example.com‘, url=’’example.org‘, url=’’example.edu‘ 와 같이 수많은 코드가 필요해진다. (장고 프레임 워크로 치면 get\_absoulte\_url() 이런식으로)  
 이때 만약에 example.org가 example.com으로 통합 되었을 경우에 example.org를 부른 코드가 1억개라면, 1억개를 다 수정해야하는 문제가 발생할 수 있다. 이때 API Gateway가 중요한 역할을 하게 된다. API Gateway가 example.io 라고 한다면, 원래 각각 불렀던 코드들이 example.io/a 혹은 example.io/b 이런식으로 만들어져서, 후에 수정이 간편해지게 된다.



-API 유형도 총 3가지로, HTTP API, WebSocket API, REST API, 로 나눌 수 있다.

-API gateway에 진입하기 위해서, 첫 주소가 필요하다. 제공해준 주소로 API gateway로 진입한다. 즉, 단일 진입점인 API gateway로 들어오는 사용자의 요청을 적당한 백엔드로 맵핑, 라우팅, 연결 시켜주는 것이 API gateway 인 것이다.

**Open Source API gateway 비교 분석 정리**

1. Kong Gateway : Lua -> 2021.07.06