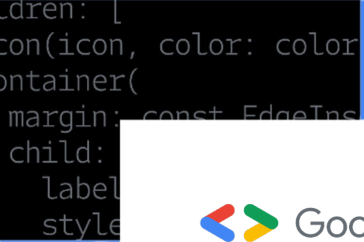
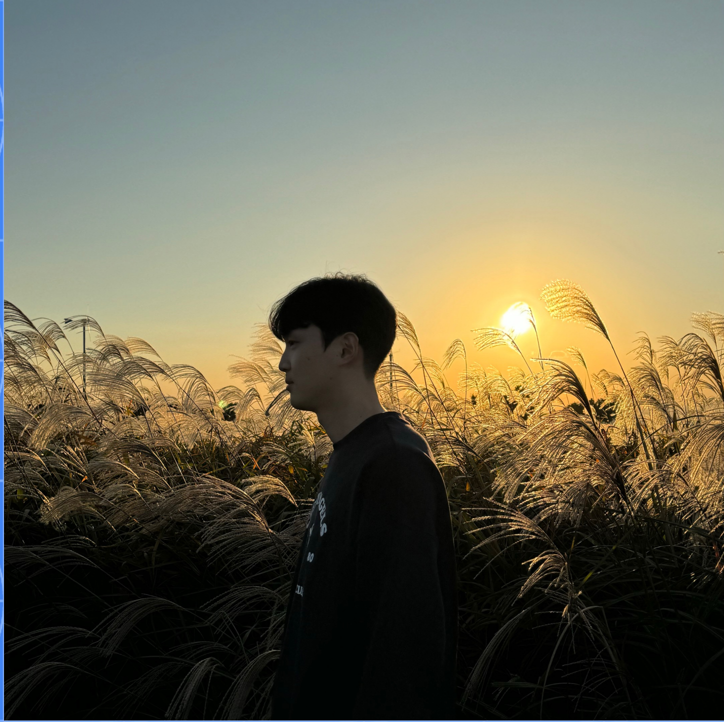


```
dren: [
con(icon, color: color
ontainer(
 margin: const EdgeIns
 child:
   label
   style
```

Google Developer Student Clubs
Kookmin University

Frontend Session #7 : RestAPI, useMemo, useCallback



Jinjae Kim
Lead, GDSC Kookmin University

Google Developer Student Clubs

Review

```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

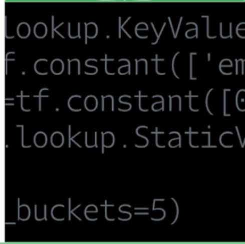
What Have We Done?

1. HTML / CSS / JavaScript
2. React JS Intro: Introduction, Structure
3. React JS #2: props, useState, useEffect
4. **React JS #3: RestAPI, useMemo, useCallback**
5. Next JS #1
6. Next JS #2: Project
7. Applications (React Native, Flutter, Android Studio, Swift)
8. Recap

Google Developer Student Clubs

Base of Web Communication

REST & REST API

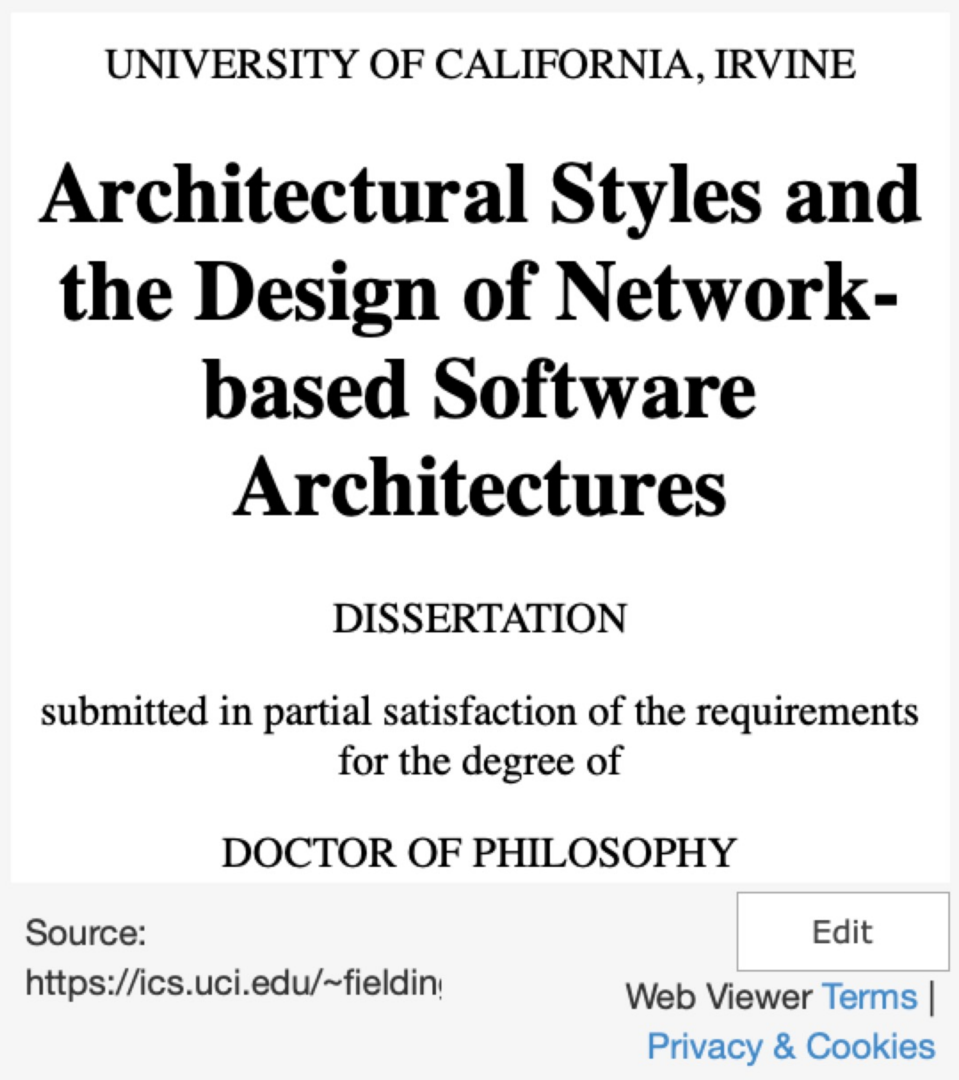


```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

REST
REpresentational State Transfer

REST Purpose?

```
To make standard

(use in Distributed
Hypermedia System)

1. HTTP Protocol
2. Client <-> Server
```



UNIVERSITY OF CALIFORNIA, IRVINE

Architectural Styles and the Design of Network-based Software Architectures

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Source:
https://ics.uci.edu/~fieldin

REST Features

1. Server-Client
2. Stateless
3. Cacheable
4. Layered System
5. Code-On-Demand (Optional)
6. Uniform Interface

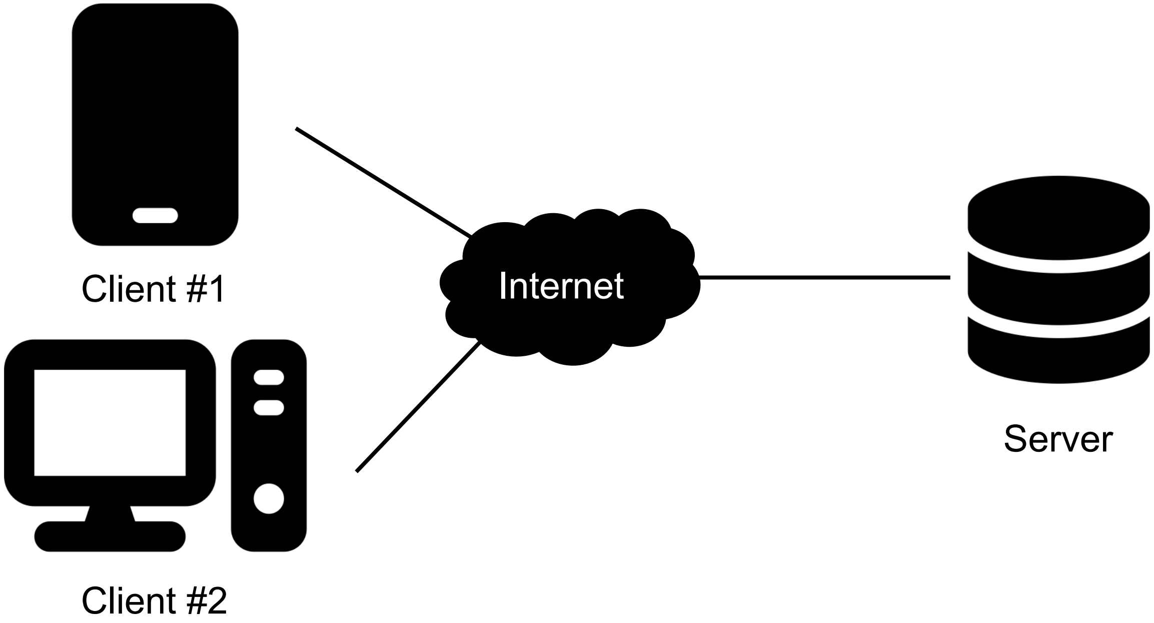
REST

1. Resource (URI)
   (http:// ~) (https:// ~)

2. Verb (HTTP Method)
   GET, POST, PUT, DELETE

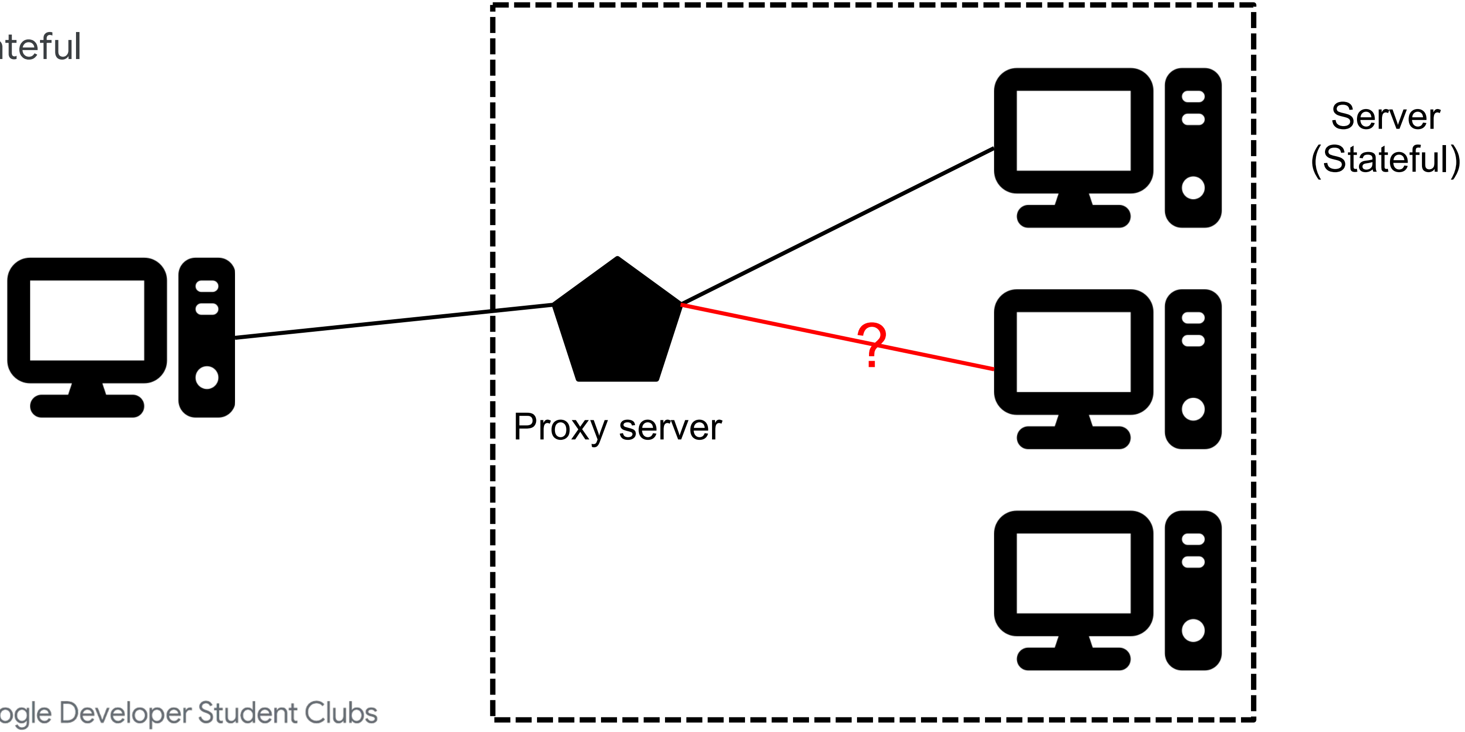
3. Representation of Resource
   JSON, XML, TEXT, RSS…

REST Features: 1. Server-Client

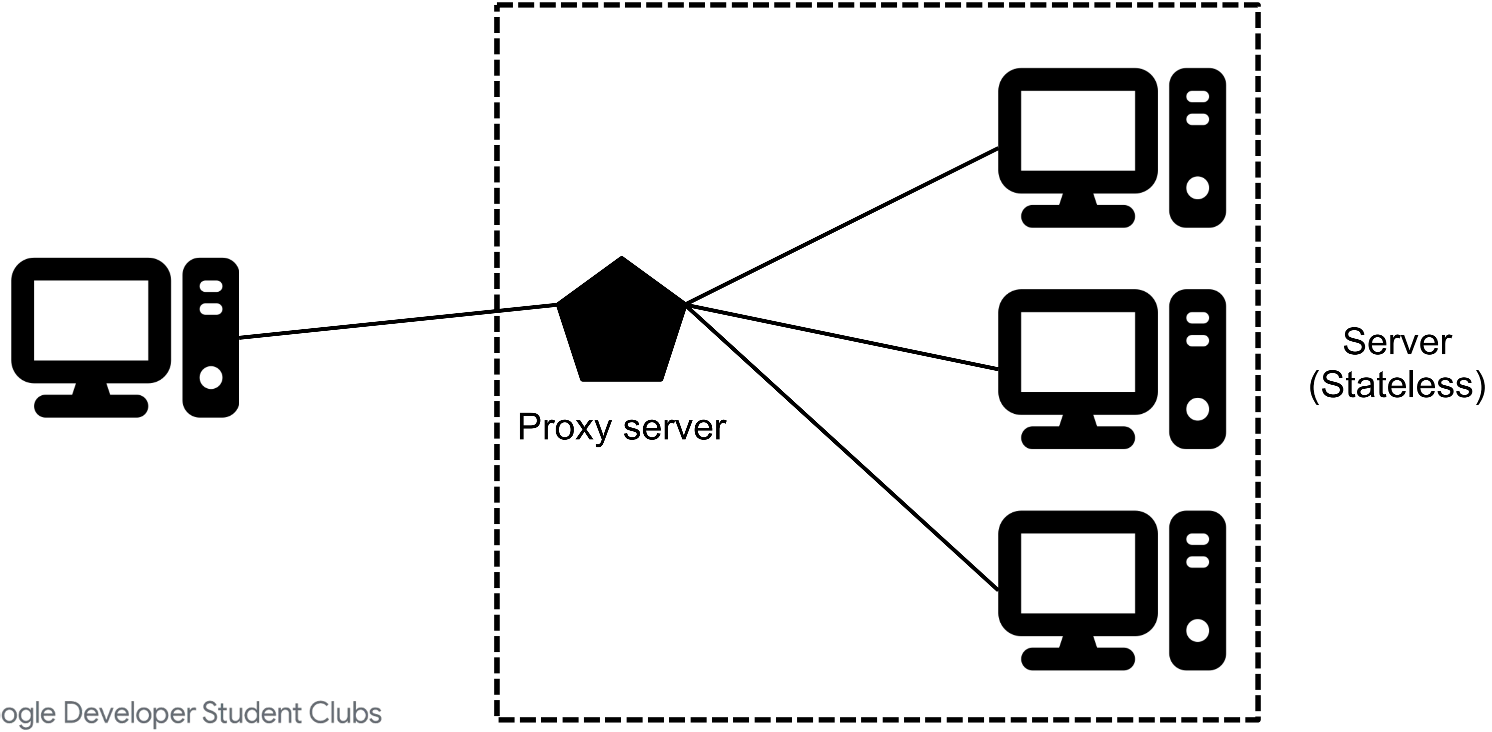


REST Features: 2. Stateless

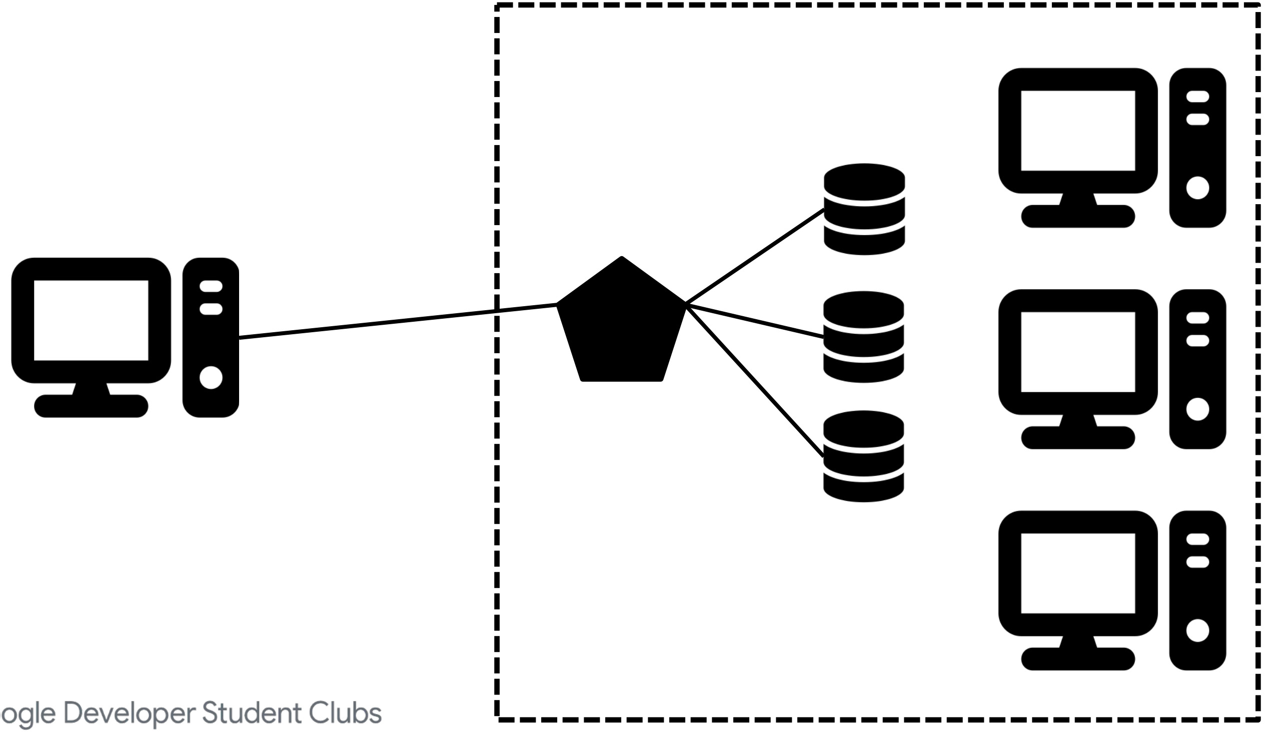
Cf) Stateful



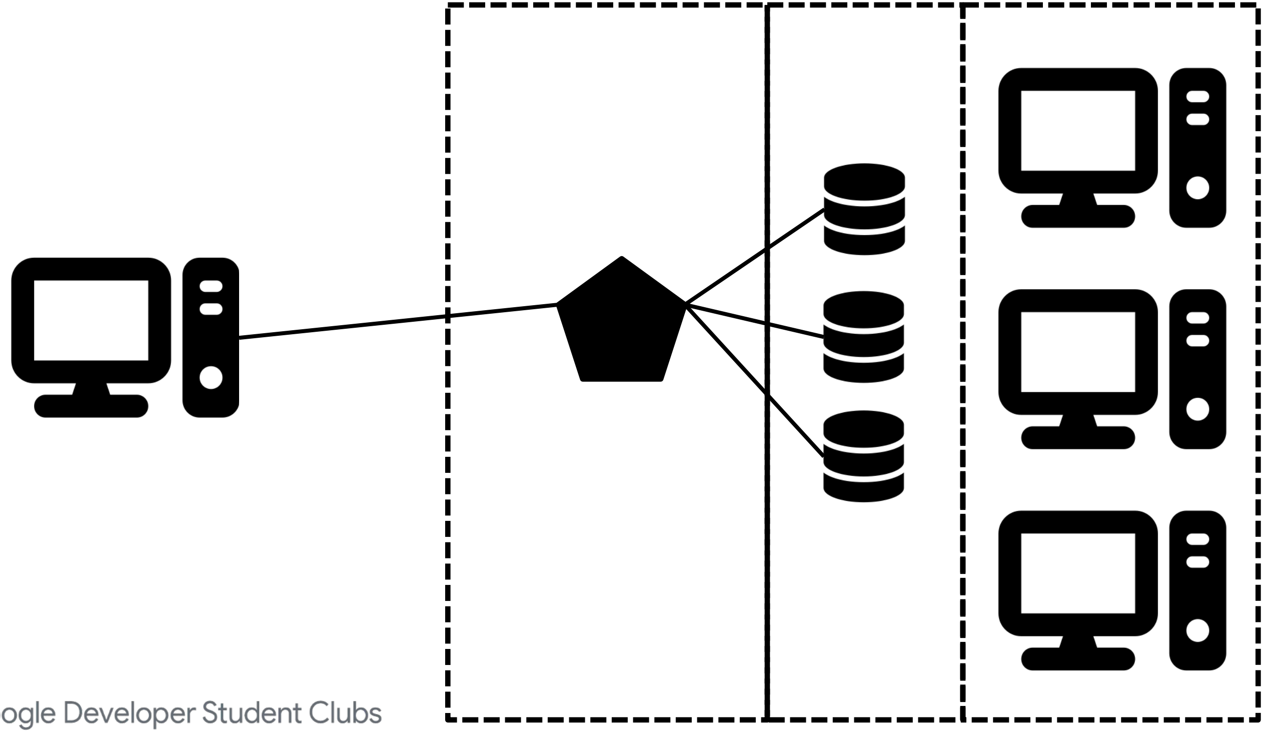
REST Features: 2. Stateless



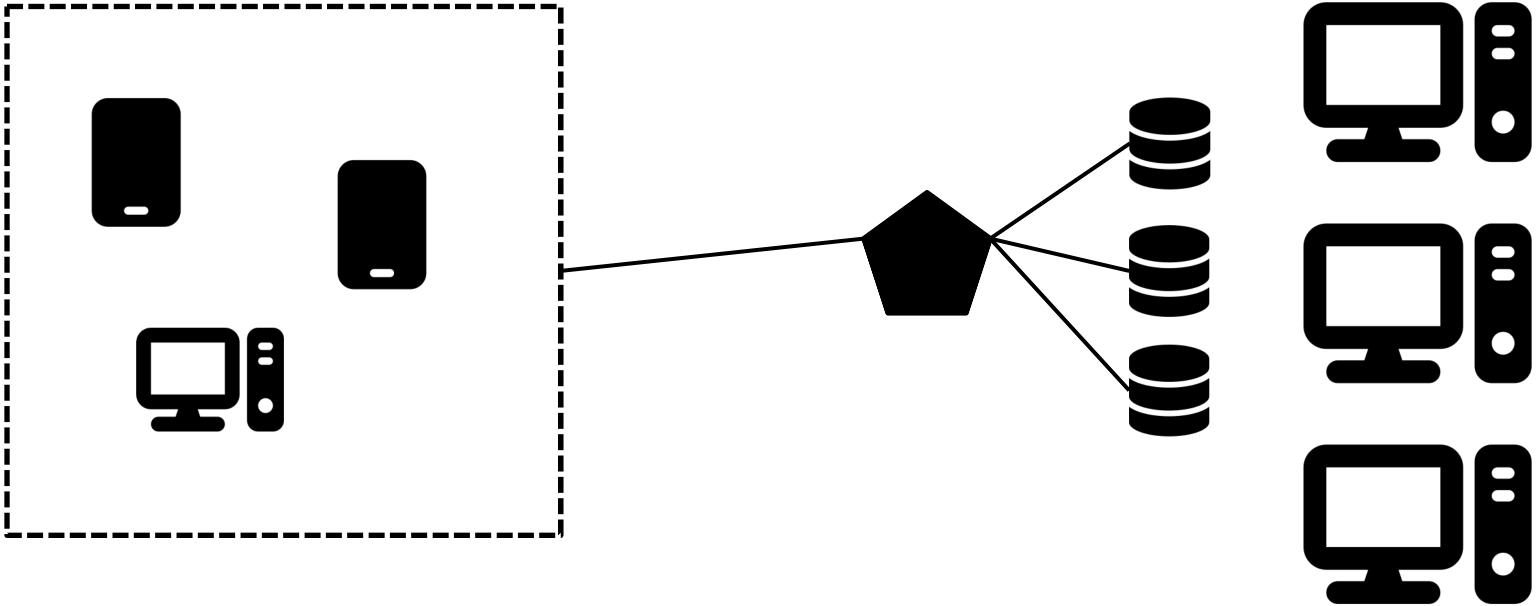
REST Features: 3. Cacheable



REST Features: 4. Layered System



REST Features: 6. Uniform Interface



REST API

API?	Application Programming Interface (for Information Communication)
REST API?	REST based API

REST API Rules

1. Slash(/) is for hierarchy
   (https://www.example.com/members/12)
2. The last character of URI does not contain slash(/)
   (https://www.example.com/members/12/) (X)
3. Use hyphen(-) for readability
4. Do not use under hyphen(_)
5. Do not use Uppercase in URL
6. Do not contain extensions (.jpg .png …)

RESTful

REST based Web Service: RESTful

Google Developer Student Clubs
Base of Web Communication

HTTP

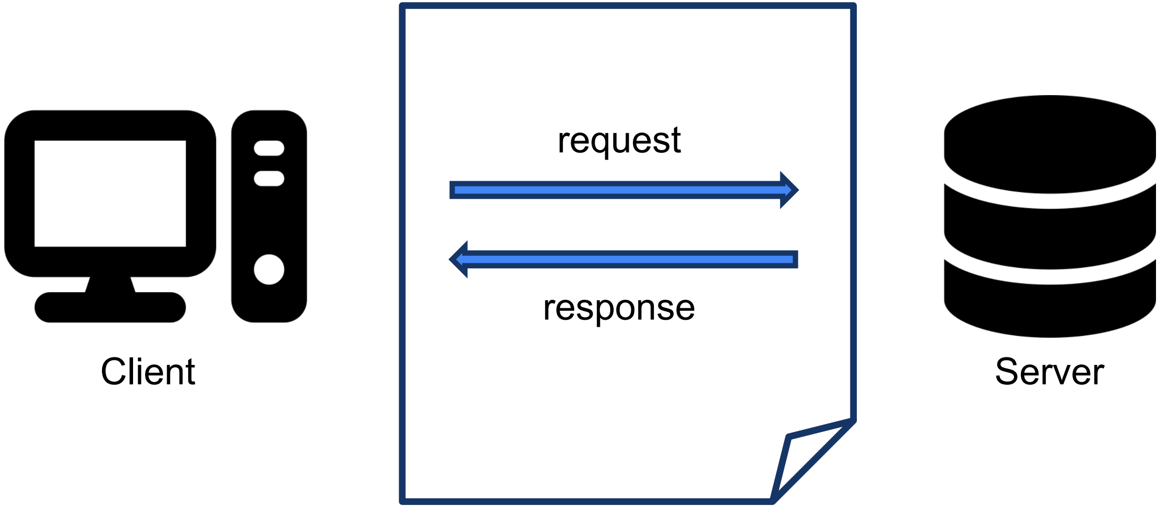
```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

HTTP

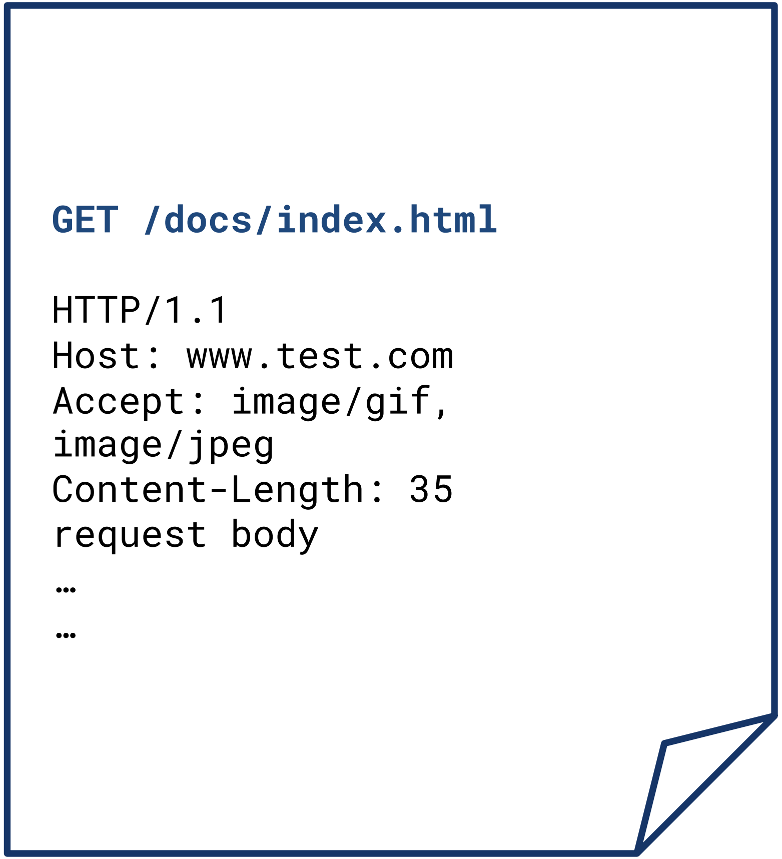
HyperText Transfer Protocol

```
Stateless Protocol

State? Cookie, Session, Token…
```

HTTP



HTTP (Request)



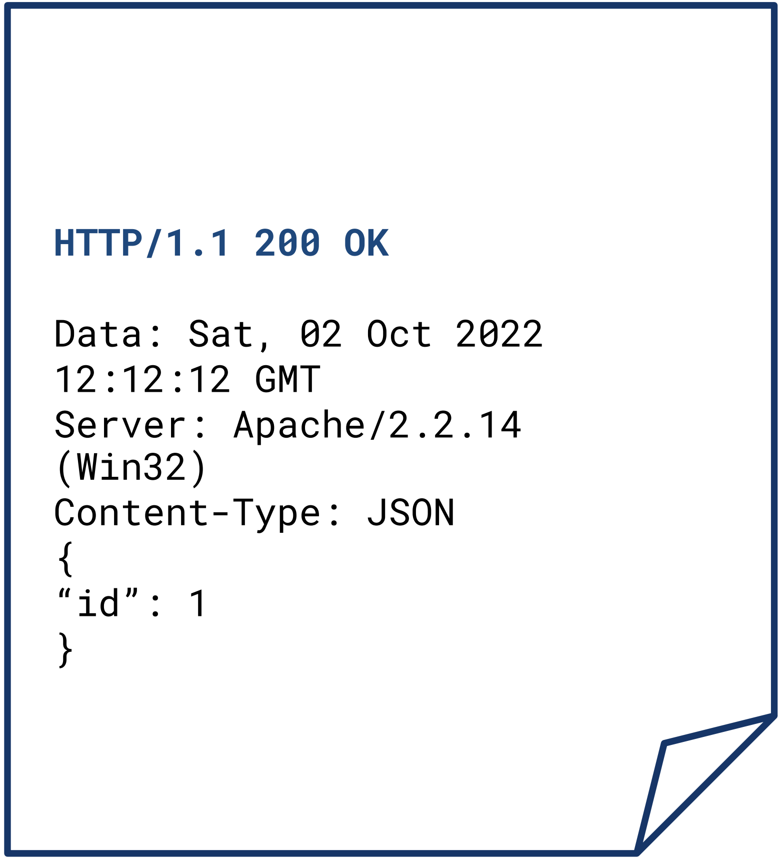
```
GET /docs/index.html

HTTP/1.1
Host: www.test.com
Accept: image/gif,
image/jpeg
Content-Length: 35
request body
…
…
```

HTTP (Request)

HTTP Transfer Method

GET	● REQUEST BODY (X)
POST	● REQUEST BODY (O)
PUT	● REQUEST BODY (O)
DELETE	● REQUEST BODY (X)

HTTP (Response)



```
HTTP/1.1 200 OK

Data: Sat, 02 Oct 2022
12:12:12 GMT
Server: Apache/2.2.14
(Win32)
Content-Type: JSON
{
“id”: 1
}
```

HTTP (Response)

https://developer.mozilla.org/ko/docs/Web/HTTP/Status

HyperText Transfer Protocol

- 1xx (정보): 요청을 받았으며 프로세스를 계속한다
- 2xx (성공): 요청을 성공적으로 받았으며 인식했고 수용하였다
- 3xx (리다이렉션): 요청 완료를 위해 추가 작업 조치가 필요하다
- 4xx (클라이언트 오류): 요청의 문법이 잘못되었거나 요청을 처리할 수 없다
- 5xx (서버 오류): 서버가 명백히 유효한 요청에 대해 충족을 실패했다

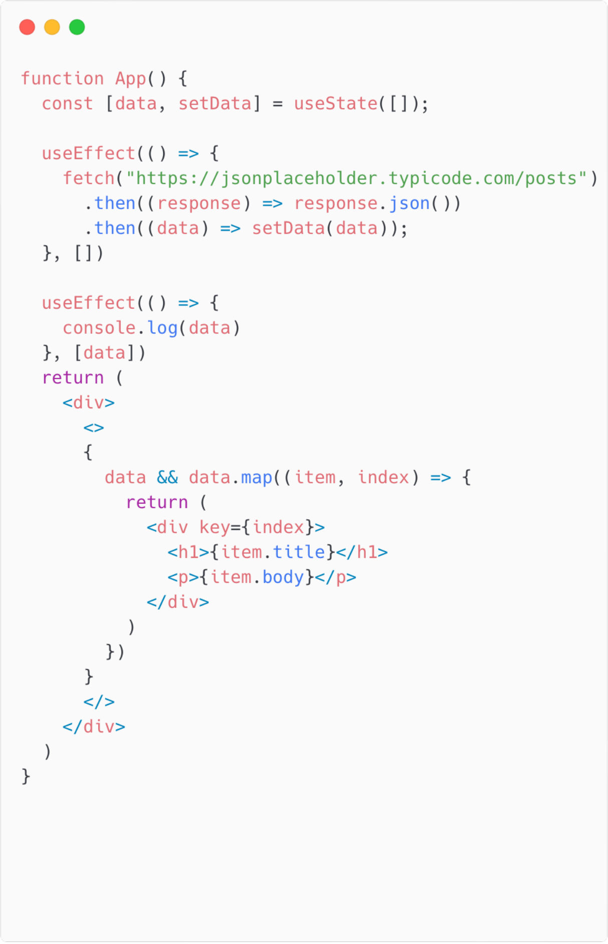
Google Developer Student Clubs

Base of Web Communication

fetch & axios

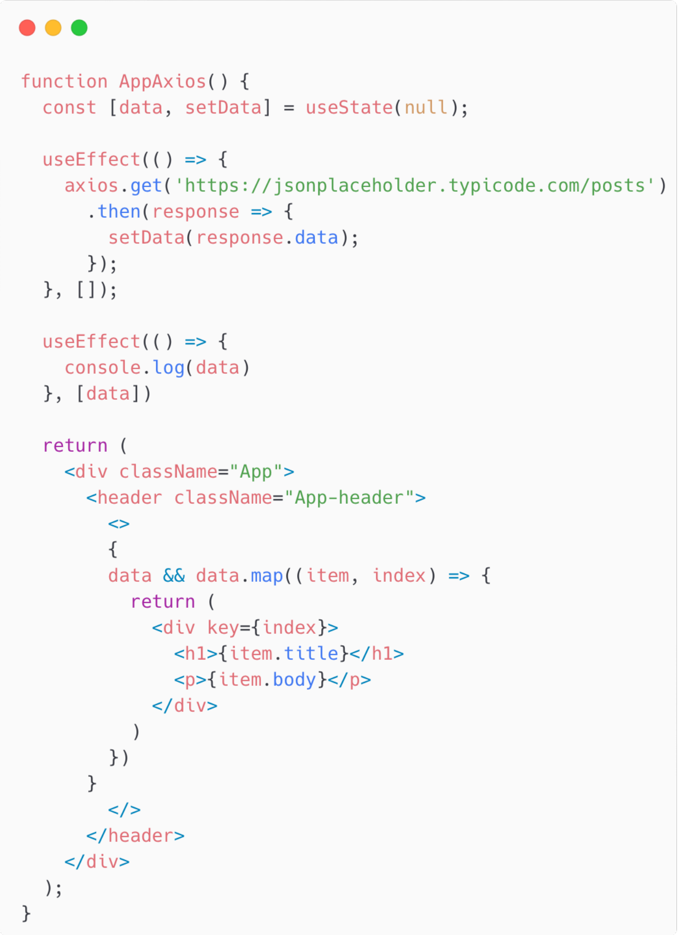
```
lookup.KeyValue
f.constant(['em
=tf.constant([0
.lookup.StaticV

_buckets=5)
```

Fetch



```
function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/posts")
      .then((response) => response.json())
      .then((data) => setData(data));
  }, [])

  useEffect(() => {
    console.log(data)
  }, [data])
  return (
    <div>
      <>
      {
        data && data.map((item, index) => {
          return (
            <div key={index}>
              <h1>{item.title}</h1>
              <p>{item.body}</p>
            </div>
          )
        })
      }
      </>
    </div>
  )
}
```

axios



```
function AppAxios() {
  const [data, setData] = useState(null);

  useEffect(() => {
    axios.get('https://jsonplaceholder.typicode.com/posts')
      .then(response => {
        setData(response.data);
      });
  }, []);

  useEffect(() => {
    console.log(data)
  }, [data])

  return (
    <div className="App">
      <header className="App-header">
        <>
        {
        data && data.map((item, index) => {
          return (
            <div key={index}>
              <h1>{item.title}</h1>
              <p>{item.body}</p>
            </div>
          )
        })
      }
        </>
      </header>
    </div>
  );
}
```

```
1*/
ild: Column(
crossAxisAlignment: CrossAxis
children: [
  /*2*/
  Conta
    pad
    chi
      '
      s

      )
    ),
  ),
  Text(
    'Ka
    sty
      c
    ),
  ),
]
```

Assignment #1

Google Developer Student Clubs

Solve Algorithm Problems (DUE ~1/26)

1. https://school.programmers.co.kr/learn/courses/30/lessons/12906
2. https://school.programmers.co.kr/learn/courses/30/lessons/42746
3. https://school.programmers.co.kr/learn/courses/30/lessons/42579

Toy Project #1

Toy Project #1 (DUE ~2/4)

서울시 내 특정 위치의 실시간 인구 현황을 확인할 수 있는 웹사이트를 제작해주세요.

(Open API)
https://data.seoul.go.kr/dataList/OA-21778/A/1/datasetView.do