Surfing on Android

Bytedance Android Engineer

hejinhai@bytedance.com

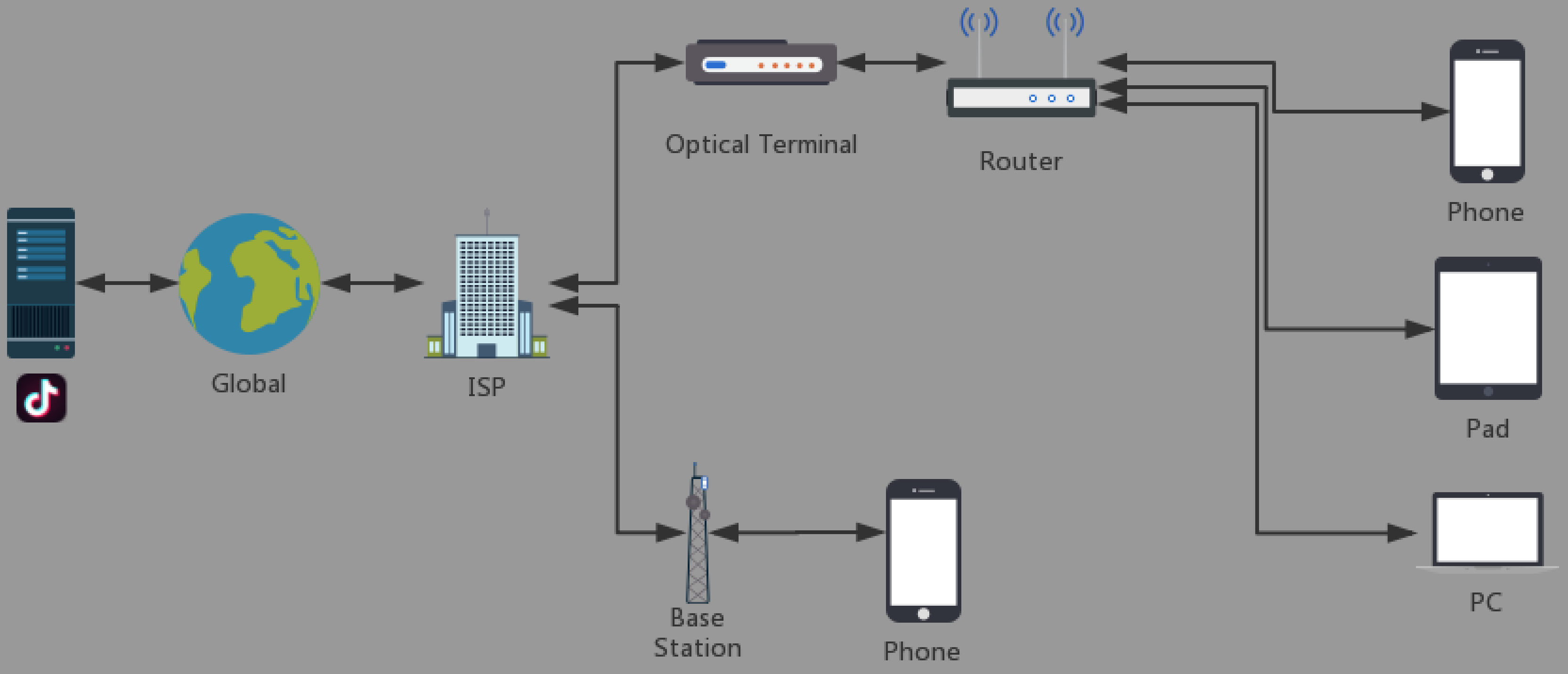
Points

- Web Basics
  - Internet
  - HTTP / RESTful API
  - Data-interchange formats
- Android Practice
  - Parse JSON
  - Fetch Data

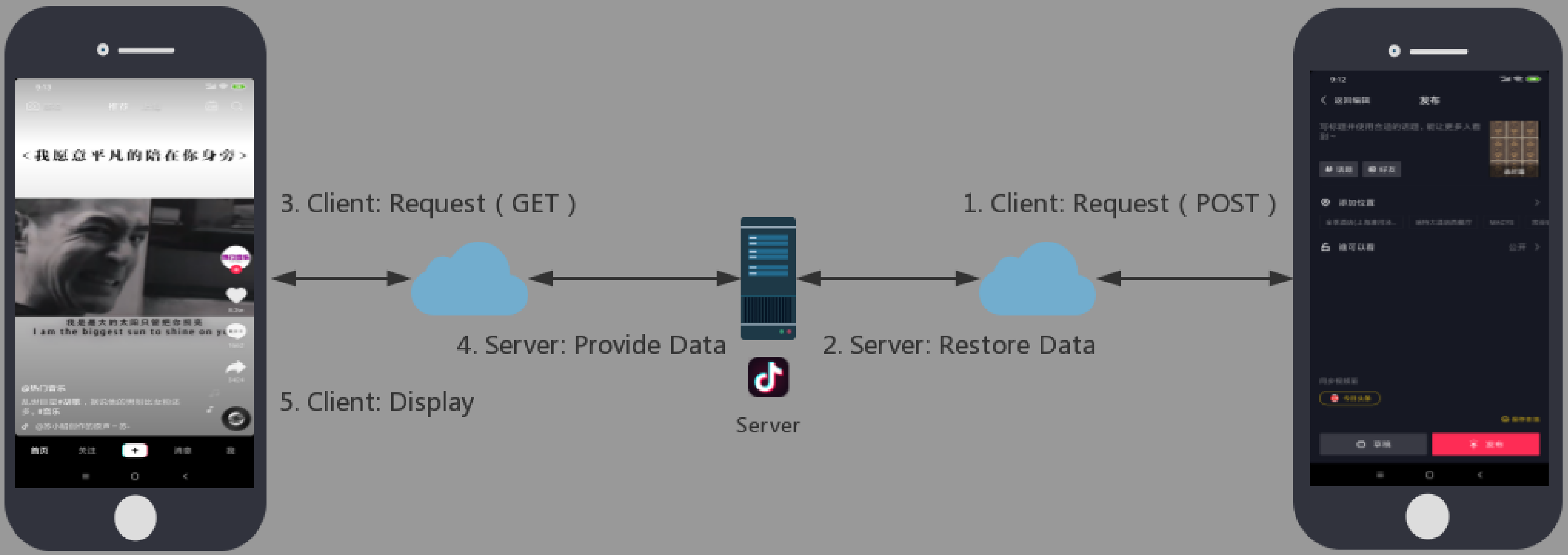
正确

Yes.

Internet (Cloud)



Web Service (C/S)



3. Client: Request (GET)

4. Server: Provide Data

5. Client: Display

1. Client: Request (POST)

2. Server: Restore Data

Server

Client

Client

HTTP

The Hypertext Transfer Protocol (Wiki)

- Standard

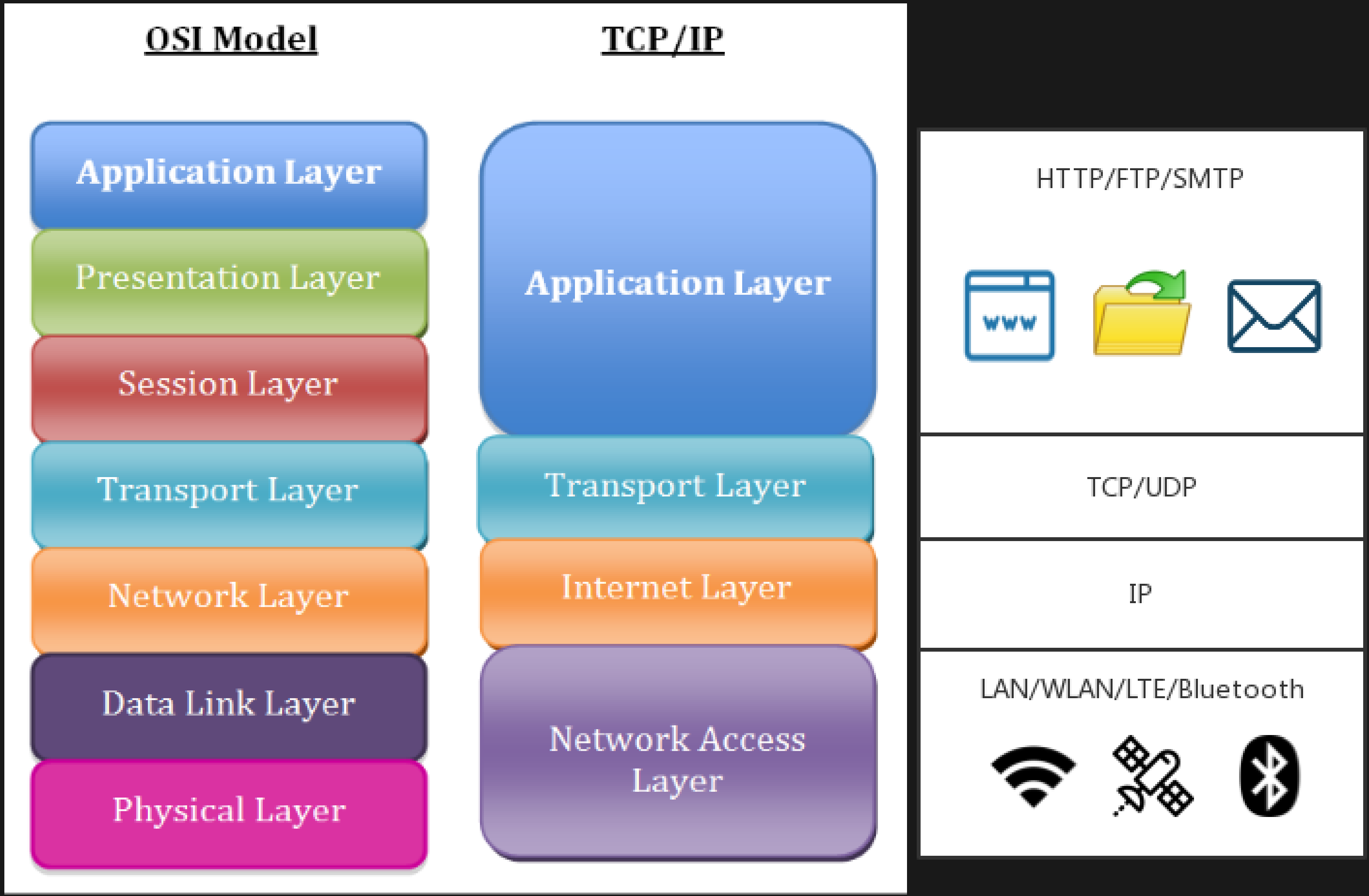
  besides sys(browser/app)

- Application layer

  easy to use

- URL: Uniform Resource Locator
- Request/Response(C/S Model)

Network Layers



OSI Model

Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

TCP/IP

Application Layer

Transport Layer

Internet Layer

Network Access Layer

HTTP/FTP/SMTP

TCP/UDP

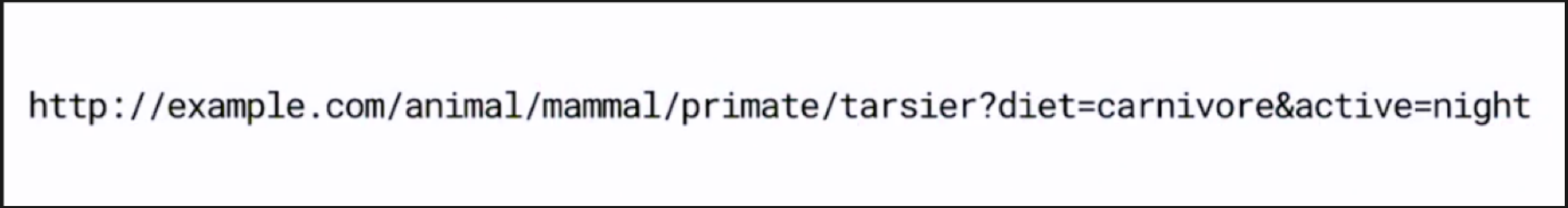
IP

LAN/WLAN/LTE/Bluetooth

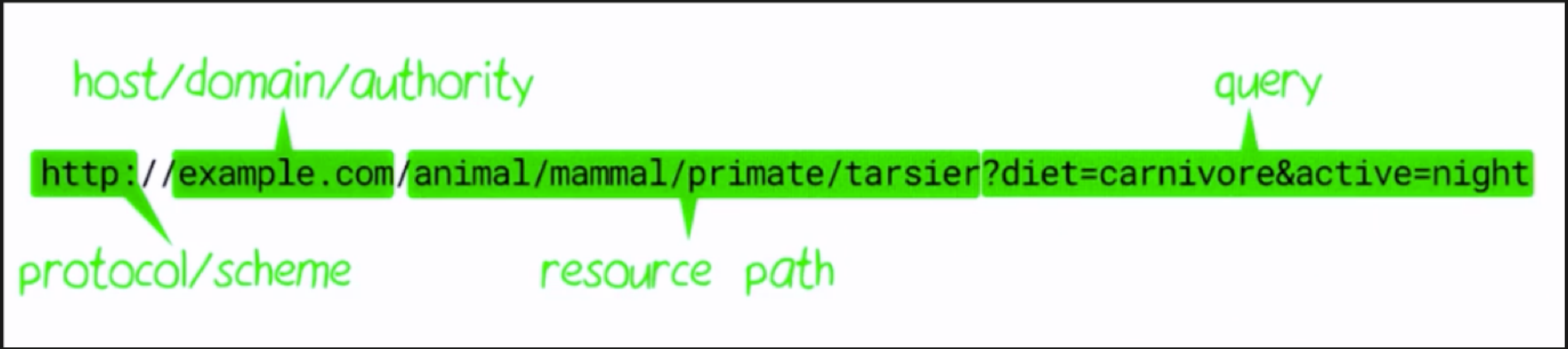
ByteDance 字节跳动

URL

Uniform Resource Locator



```
http://example.com/animal/mammal/primate/tarsier?diet=carnivore&active=night
```


host/domain/authority
query
http://example.com/animal/mammal/primate/tarsier?diet=carnivore&active=night
protocol/scheme
resource path

Request/Response

Client request

```
GET / HTTP/1.1
Host: www.example.com
```

Server response

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/json; charset=UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"

{"message": "hello"}
```

REST

REpresentational State Transfer

1. Resources

   web entity(text/music/picture/video)

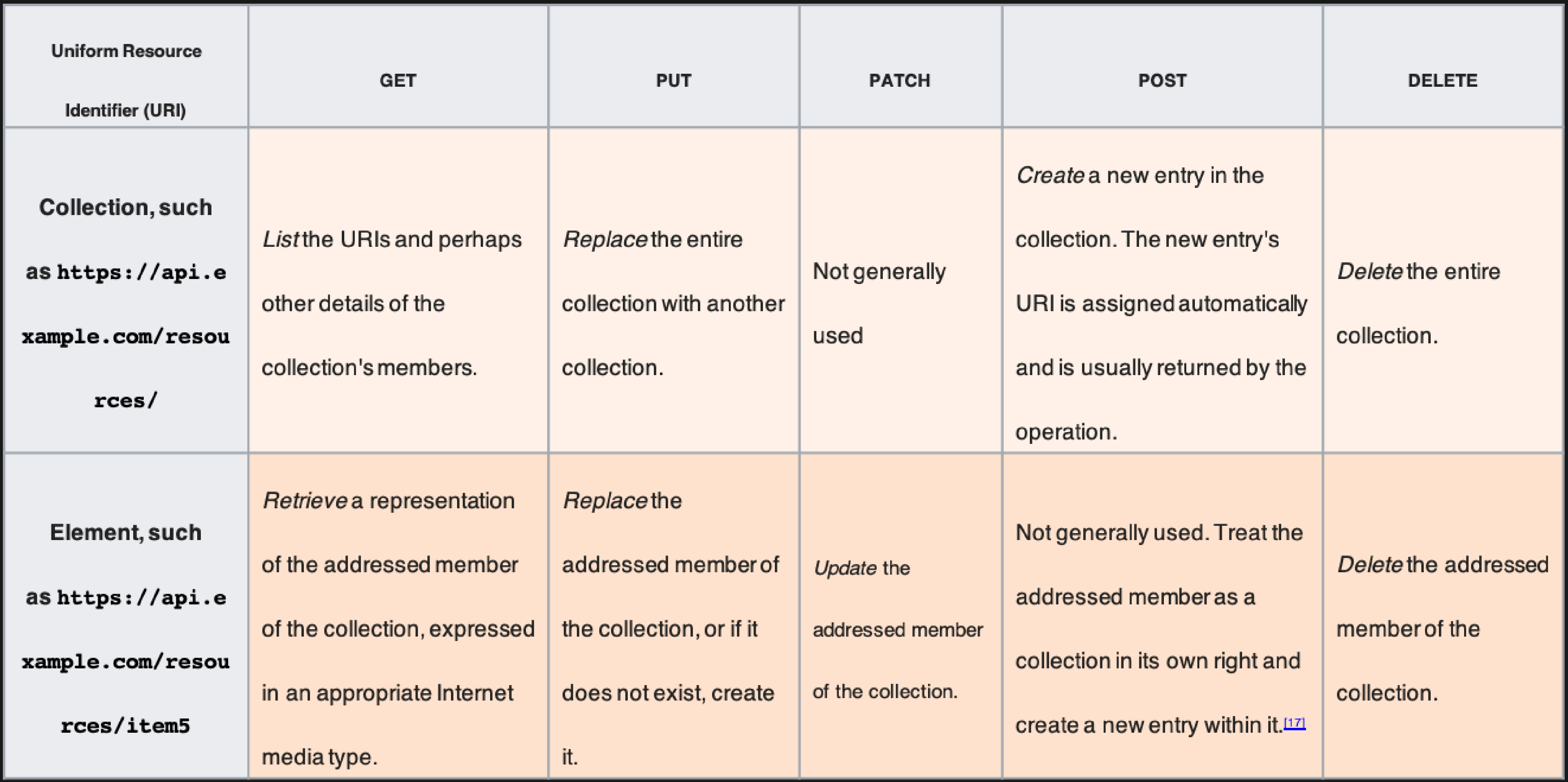
2. Representation

   type/format(html/json/jpg)

3. State Transfer

   methods(POST/GET/PUT/DELETE)

REST Methods



Uniform Resource Identifier (URI)	GET	PUT	PATCH	POST	DELETE
Collection, such as `https://api.example.com/resources/`	*List* the URIs and perhaps other details of the collection's members.	*Replace* the entire collection with another collection.	Not generally used	*Create* a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.	*Delete* the entire collection.
Element, such as `https://api.example.com/resources/item5`	*Retrieve* a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	*Replace* the addressed member of the collection, or if it does not exist, create it.	*Update* the addressed member of the collection.	Not generally used. Treat the addressed member as a collection in its own right and create a new entry within it.[17]	*Delete* the addressed member of the collection.

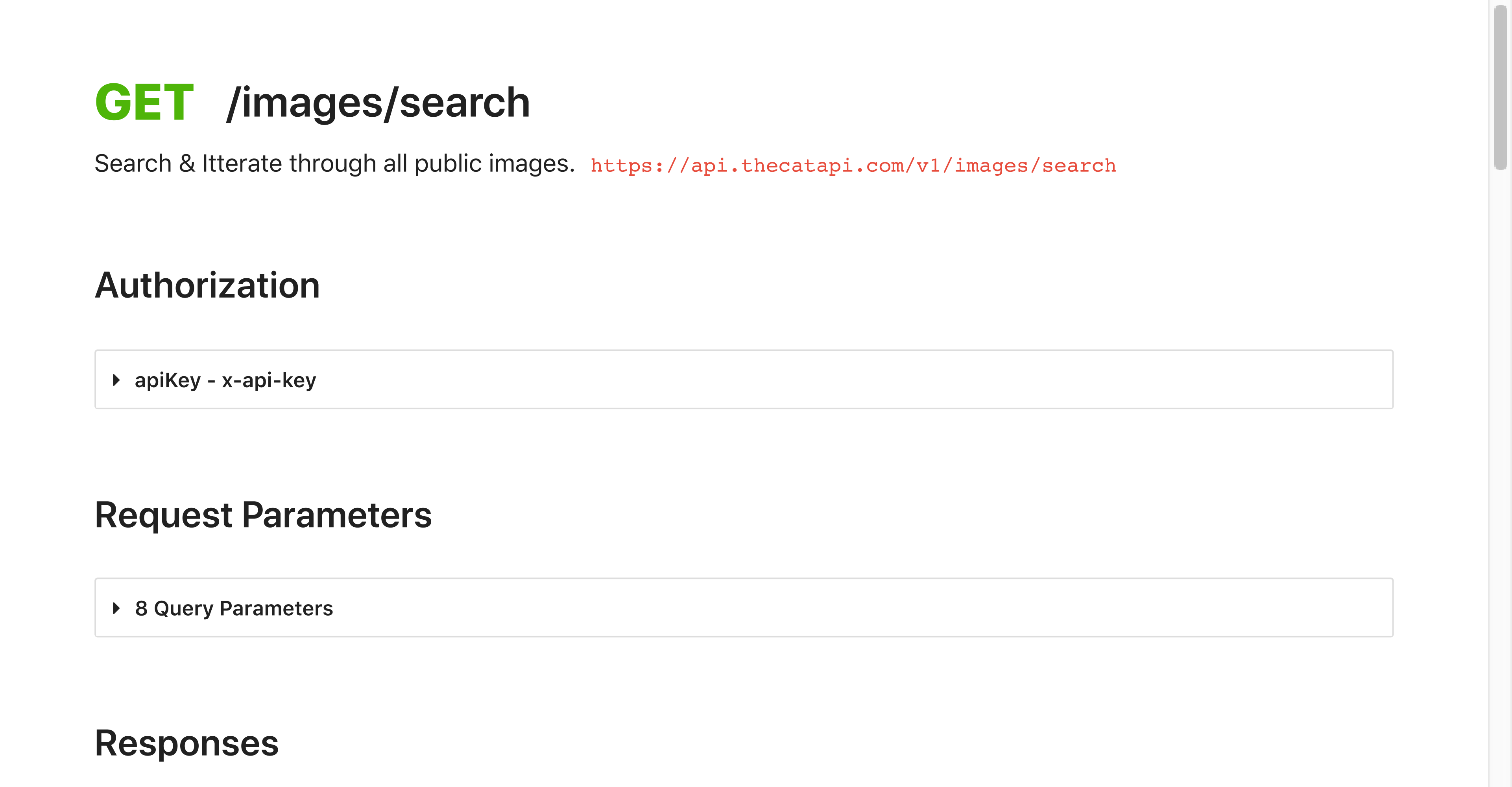
REST Methods example

Method	http://x.c/video/	http://x.c/video/123
GET	get all videos	get video 123
POST	post video	-
PUT	-	update video 123
DELETE	delete all videos	delete video 123

REST Documents

- REST Wiki
- REST API Tutorial
- 理解RESTful架构
- PostMan
- The Cat API
- Github API

The Cat API



GET /images/search

Search & Itterate through all public images. `https://api.thecatapi.com/v1/images/search`

Authorization

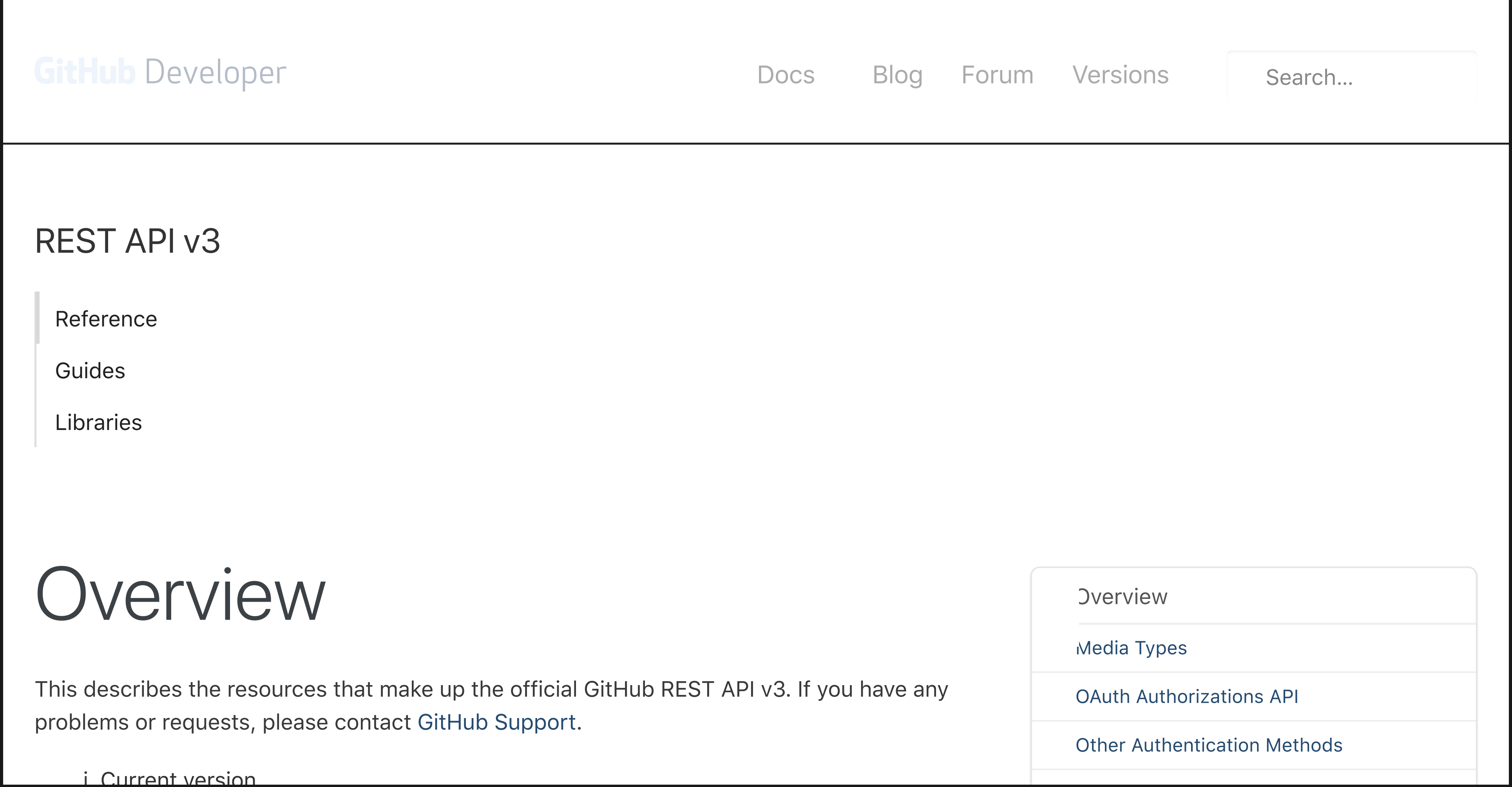
▸ apiKey - x-api-key

Request Parameters

▸ 8 Query Parameters

Responses

Github API



REST API v3

Overview

This describes the resources that make up the official GitHub REST API v3. If you have any problems or requests, please contact GitHub Support.

i. Current version

Data-interchange formats

Try to represent user info:

- name(Alice)
- age(20)
- languages(zh,en)

JSON vs XML

```
// JSON - JavaScript Object Notation
{
    "name": "Alice",
    "age": 20,
    "languages": ["zh", "en"]
}
```

```
<xml><!-- XML - Extensible Markup Language-->
    <name>Alice</name>
    <age type="int">20</age>
    <languages>
        <language>zh</language>
        <language>en</language>
    </languages>
</xml>
```

JSON data types

```json
{
    "string_name": "Alice",
    "number_age": 20,
    "boolean_vip": true,
    "array_languages": ["zh", "en"],
    "object_school": {
        "name": "Zhejiang University",
        "city": "Hangzhou"
    },
    "null_email": null
}
```

Json.cn (/)

在线解析 （/） 什么是JSON （/wiki.html）
JSON解析代码 （/code.html） JSON组件 （/component.html）

在此输入json字符串或XML字符串...

©2014 JSON.cn All right reserved. 京ICP备15025187号-1 (http://www.miitbeian.gov.cn/) 邮箱: service@json.cn

JSON Practice

- id(102381)
- title(Fireworks)
- author(name(Luna) country(China))
- url(http://x.c/video/102381.mp4)
- like_count(200)
- tags(Art,VLog)

Parse JSON

- JSONObject
- Gson

JSONObject

Doc

```java
public static String DATA = "{\"name\":\"Sylvia\",\"age\":26}";

JSONObject user = new JSONObject(DATA);
String name = user.getString("name");
int age = user.getInt("age");
```

Gson

Github

```java
public static String DATA = "{\"name\":\"Sylvia\",\"age\":26}";

public class User {
    @SerializedName("name")
    public String name;

    @SerializedName("age")
    public int age;
}

User user = new Gson().fromJson(DATA, User.class);
String result = user.name;
int age = user.age;
```

Fetch Data

- HttpUrlConnection
- Retrofit

Permissions

```
<manifest xmlns:tools="http://schemas.android.com/tools"
    package="com.bytedance.android.lesson.restapi"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"

    <application>
    ...
    </application>

</manifest>
```

HttpUrlConnection

Doc

```
String url = "https://api.thecatapi.com/v1/images/search";

// connect
URL netUrl = new URL(url);
HttpURLConnection urlConnection = (HttpURLConnection) netUrl.openConnection(

// read response data
BufferedInputStream in = new BufferedInputStream(urlConnection.getInputStrea
String data = readStream(in);

// clean up
urlConnection.disconnect();
```

Retrofit

Github

```
// definition
public interface ICNDBService {
    @GET("jokes/random") Call randomJoke();
}

Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("http://api.icndb.com/")
        .addConverterFactory(GsonConverterFactory.create())
        .build();
ICNDBService service = retrofit.create(ICNDBService.class);
Response response = service.randomJoke().execute();
Joke joke = response == null ? null : response.body();
```