

Surfing on Android

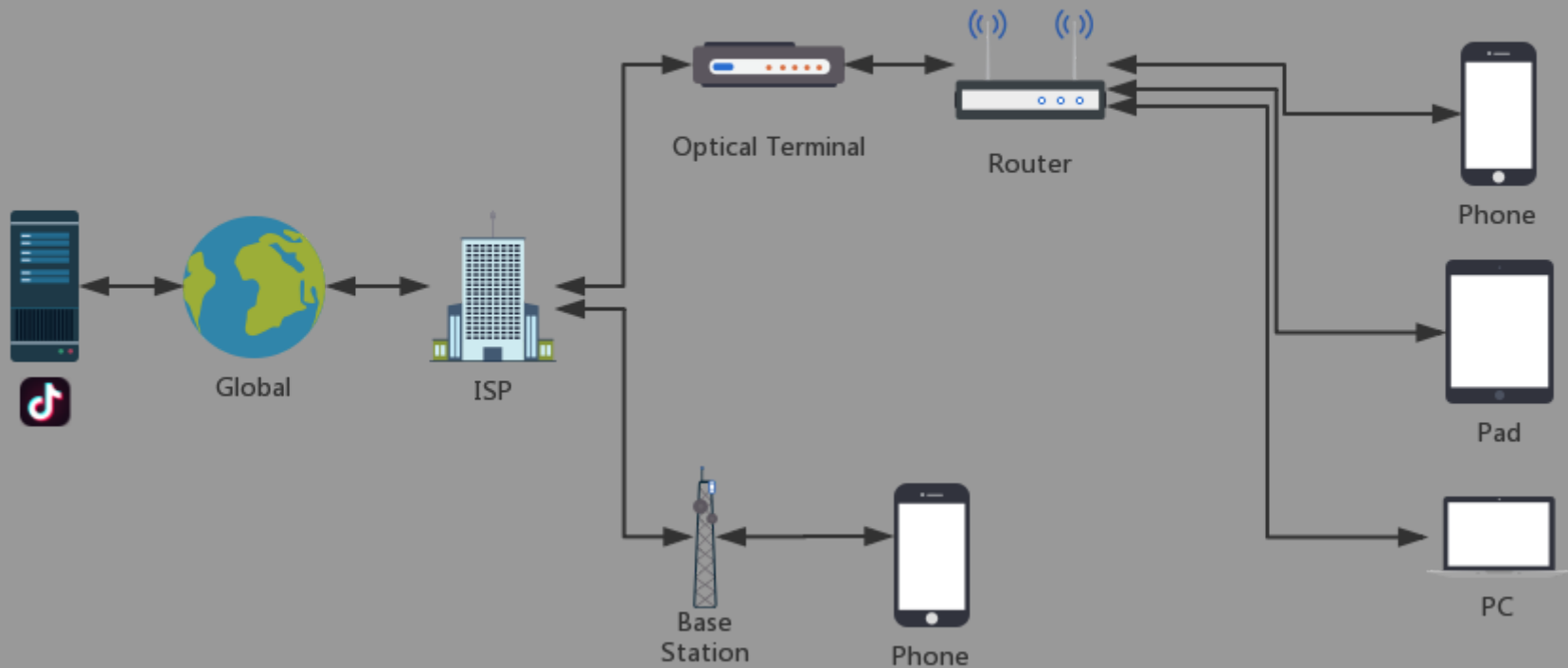
Bytedance Android Engineer
hejinhai@bytedance.com

Points

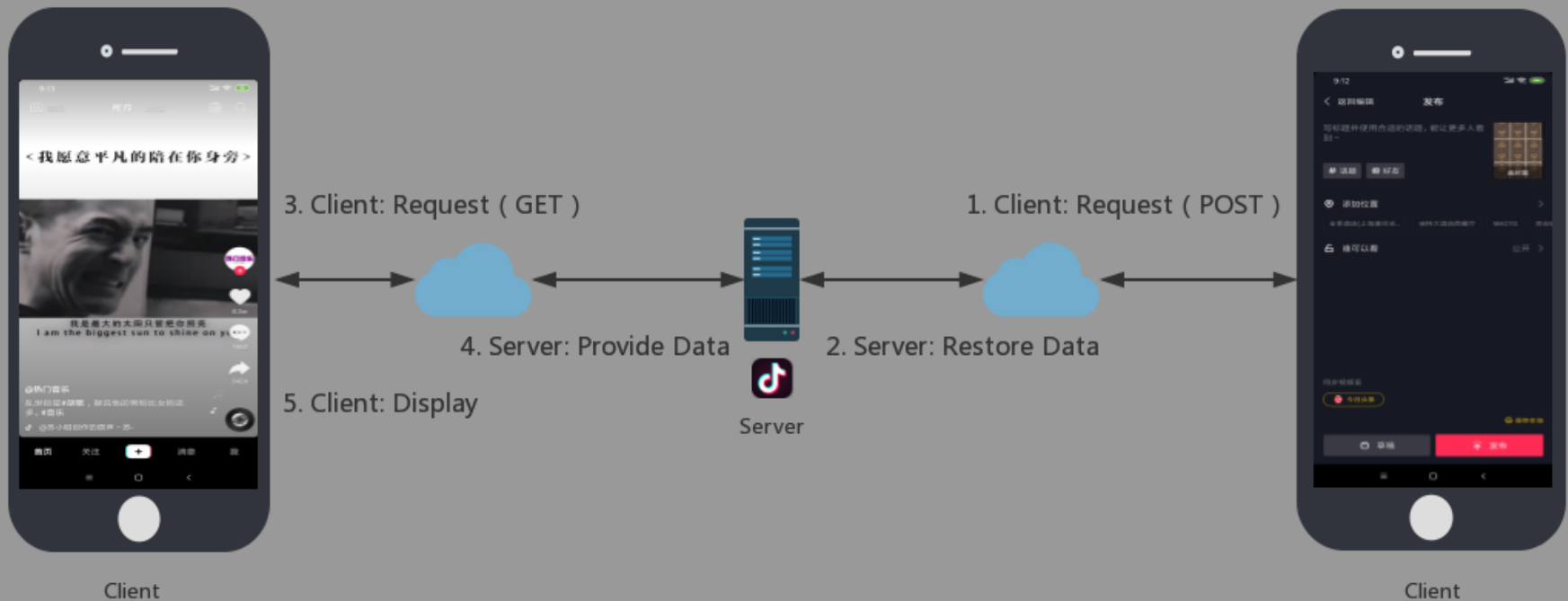
- Web Basics
 - Internet
 - HTTP / RESTful API
 - Data-interchange formats
- Android Practice
 - Parse JSON
 - Fetch Data



Internet (Cloud)



Web Service (C/S)

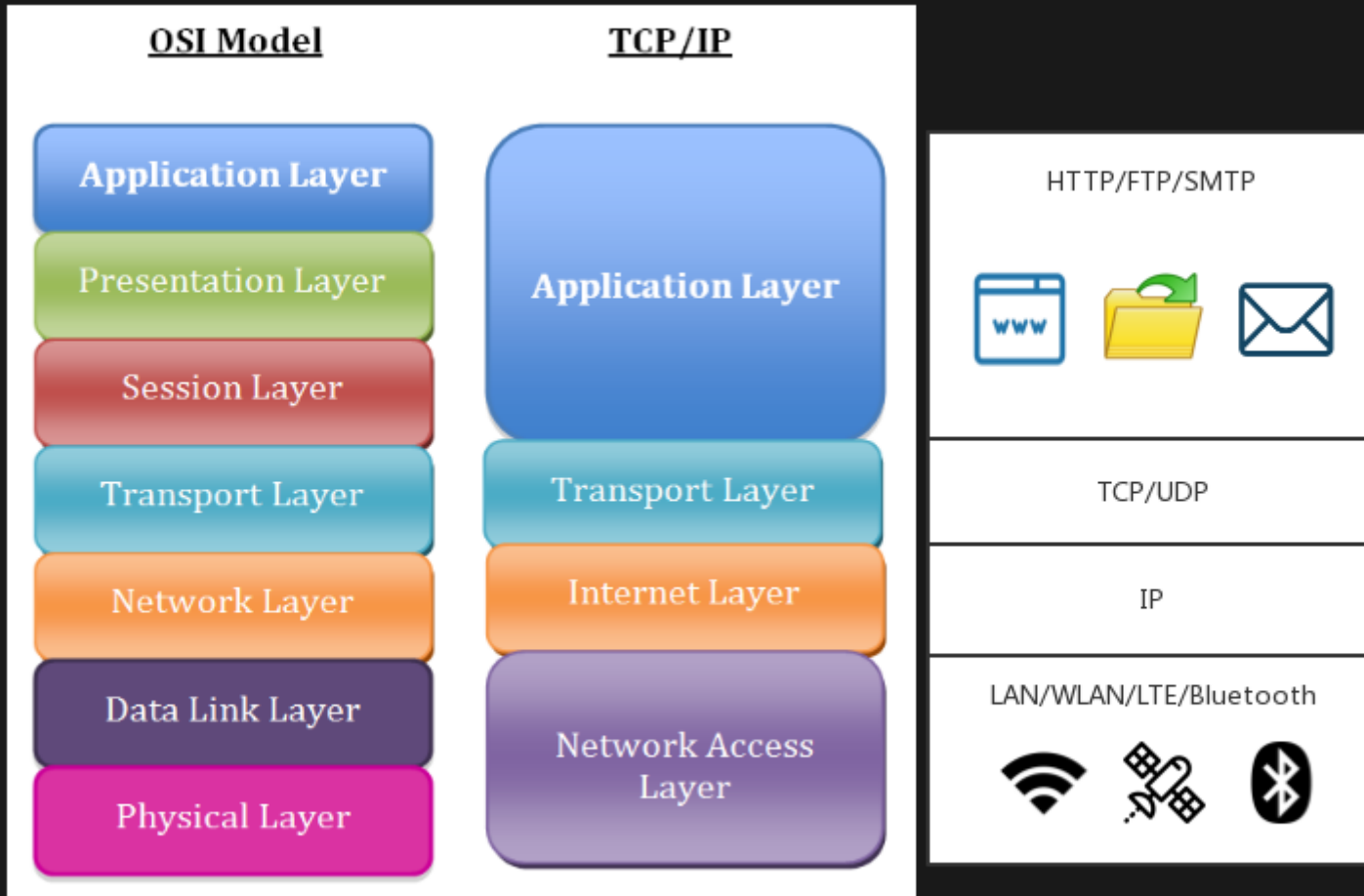


HTTP

The Hypertext Transfer Protocol ([Wiki](#))

- Standard
besides sys(brower/app)
- Application layer
easy to use
- URL: Uniform Resource Locator
- Request/Response(C/S Model)

Network Layers



URL

Uniform Resource Locator

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

The diagram illustrates the components of the URL `http://example.com/animal/mammal/primate/tarsier?diet=carnivore&active=night`. The URL is displayed with segments highlighted in green boxes. Labels in green text with callout lines point to these segments:
- `http://` is labeled `protocol/scheme`.
- `example.com` is labeled `host/domain/authority`.
- `/animal/mammal/primate/tarsier` is labeled `resource path`.
- `?diet=carnivore&active=night` is labeled `query`.

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" }
```

ByteDance

字节跳动

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 <code>https://api.example.com/resources/</code>	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 <code>https://api.example.com/resources/item5</code>	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 & Iterate through all public images. <https://api.thecatapi.com/v1/images/search>

Authorization

▸ apiKey - x-api-key

Request Parameters

▸ 8 Query Parameters

Responses

Github API

GitHub Developer

[Docs](#)

[Blog](#)

[Forum](#)

[Versions](#)

REST API v3

[Reference](#)

[Guides](#)

[Libraries](#)

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](#).

[Overview](#)

[Media Types](#)

[OAuth Authorizations API](#)

[Other Authentication Methods](#)

[v3.0.0](#) | [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>20</age>
  <languages>
    <language>zh</language>
    <language>en</language>
  </languages>
</xml>
```

JSON data types

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

在此输入json字符串或XML字符串...



JSON Practice

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

ByteDance

字节跳动

Parse JSON

- JSONObject
- Gson

JSONObject

Doc

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

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

Gson

Github

```
public static String DATA = "{\n\"name\":\n\"Sylvia\", \n\"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.getInputStream())
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();
```