



# REST API in Flutter

PowerPoint by Wanida Khamprapai

# REST API

## API (Application Programming Interface)

- The rules that you must follow to communicate with other software systems.
- Developers expose or create APIs so that other applications can communicate with their applications programmatically.

## REST (Representational State Transfer)

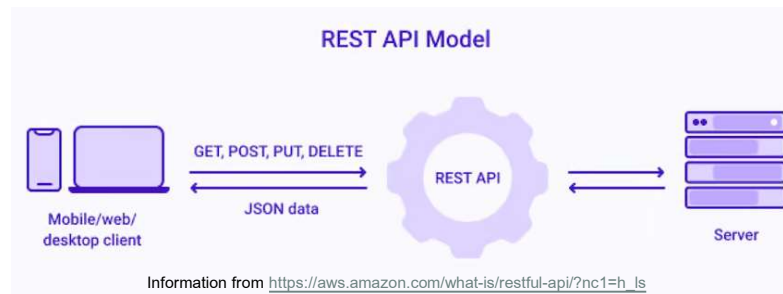
- A software architecture that imposes conditions on how an API should work.
- REST was initially created as a guideline to manage communication on a complex network like the Internet.
- You can use REST-based architecture to support high-performing and reliable communication at scale. You can easily implement and modify it, bringing visibility and cross-platform portability to any API system

Information from [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)

# REST API

## These are the general steps for any REST API call:

- The client sends a request to the server. The client follows the API documentation to format the request in a way that the server understands.
- The server authenticates the client and confirms that the client has the right to make that request.
- The server receives the request and processes it internally.
- The server returns a response to the client. The response contains information that tells the client whether the request was successful. The response also includes any information that the client requested.



# REST API

REST APIs require requests to contain the following main components:

- **Unique resource identifier**

- The server identifies each resource with unique resource identifiers. For REST services, the server typically performs resource identification by using a Uniform Resource Locator (URL).
- The URL specifies the path to the resource. A URL is similar to the website address that you enter into your browser to visit any webpage.
- The URL is also called the request endpoint and clearly specifies to the server what the client requires.

Information from [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)

# REST API

REST APIs require requests to contain the following main components: (Cont.)

- **Method**

- Developers often implement REST APIs by using the Hypertext Transfer Protocol (HTTP). An HTTP method tells the server what it needs to do to the resource. The following are four common HTTP methods:
  - **GET:** To access resources that are located at the specified URL on the server.
  - **POST:** To send data to the server.
  - **PUT:** To update existing resources on the server.
  - **DELETE:** To remove the resource.

Information from [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)

# REST API

REST APIs require requests to contain the following main components: (Cont.)

- **HTTP headers**

- The metadata exchanged between the client and server. For instance, the request header indicates the format of the request and response, provides information about request status, and so on.
  - **Data:** REST API requests might include data for the POST, PUT, and other HTTP methods to work successfully.
  - **Parameter:** REST API requests can include parameters that give the server more details about what needs to be done. The following are some different types of parameters:
    - Path parameters that specify URL details.
    - Query parameters that request more information about the resource.
    - Cookie parameters that authenticate clients quickly.

Information from [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)

# REST API

## REST API authentication methods

- A REST web service must authenticate requests before it can send a response. Authentication is the process of verifying an identity.
- **HTTP authentication:** HTTP defines some authentication schemes that you can use directly when you are implementing REST API. The following are two of these schemes:
  - Basic authentication: The client sends the username and password in the request header.
  - Bearer authentication: The process of giving access control to the token bearer. The bearer token is typically an encrypted string of characters that the server generates in response to a login request. The client sends the token in the request headers to access resources.

Information from [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)

# REST API

## REST API authentication methods (Cont.)

### ■ API Keys:

- API keys are another option for REST API authentication.
- In this approach, the server assigns a unique generated value to a first-time client.
- Whenever the client tries to access resources, it uses the unique API key to verify itself. API keys are less secure because the client has to transmit the key, which makes it vulnerable to network theft.

### ■ OAuth:

- OAuth combines passwords and tokens for highly secure login access to any system.
- The server first requests a password and then asks for an additional token to complete the authorization process.
- It can check the token at any time and also over time with a specific scope and longevity.

Information from [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)



# REST API

## REST API server response

- **Status line:**
  - The status line contains a three-digit status code that communicates request success or failure.
  - For instance, 2XX codes indicate success, but 4XX and 5XX codes indicate errors. 3XX codes indicate URL redirection.
  - The following are some common status codes: (<https://www.restapitutorial.com/httpstatuscodes.html>)
    - 200: Generic success response
    - 201: POST method success response
    - 400: Incorrect request that the server cannot process
    - 404: Resource not found

Information from [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)

# REST API

## REST API server response (Cont.)

### ■ Message body:

- The response body contains the resource representation.
- The server selects an appropriate representation format based on what the request headers contain. Clients can request information in XML or JSON formats, which define how the data is written in plain text.
- For example, if the client requests the name and age of a person named John, the server returns a JSON representation as follows:

```
'{"name":"John", "age":30}'
```

### ■ Headers:

- The response also contains headers or metadata about the response. They give more context about the response and include information such as the server, encoding, date, and content type.

Information from [https://aws.amazon.com/what-is/restful-api/?nc1=h\\_ls](https://aws.amazon.com/what-is/restful-api/?nc1=h_ls)

# Implementing REST API in Flutter

## Step 1: Setting up the project

- Install the **http** dependency in order to use API in the application. (To find the latest version of the package and more, check <https://pub.dev>)


## Step 2: Creating a request

- The basic request uses the get method to fetch the data from the specified URL in JSON format.

## Step 3: Creating a model class to parse the JSON

## Step 4: Creating a file that handles the API call, and write specific methods to fetch and parse data

## Step 5: Use the data in your app



# Question & Answer

#### ---- Demo API ----

<https://fakestoreapi.com>

<https://dog.ceo/dog-api/>

<https://jsonplaceholder.typicode.com>

<https://www.balldontlie.io/#introduction>

#### ---- convert JSON to Dart ----

[https://javiercbk.github.io/json\\_to\\_dart/](https://javiercbk.github.io/json_to_dart/)

#### ---- Icons for app ----

<https://www.flaticon.com/>

<https://icons8.com/>

<https://romannurik.github.io/AndroidAssetStudio/>

<https://makeappicon.com>

<https://icon.kitchen>

# Change application name and icon

## Changing the app name for Android

- Navigate to the android > app > src > main and open the AndroidManifest.xml
- Under the application tag, Find the **android:label** and replace its value with the new app name
- Save and run your app

```
<application
  android:name="io.flutter.app.FlutterApplication"
  android:label="your_app_name"
  android:icon="@mipmap/ic_launcher">
```

## Changing the app name for iOS

- Navigate to the ios>Runner and open the info.plist file.
- Find the key named as CFBundleName and replace the string value (below it) to reflect the new app name.
- Save and run your app

```
<key>CFBundleName</key>
<string>your_app_name</string>
```

# Change application name and icon

## Changing the app icon for Android

- Install the **flutter\_launcher\_icons** dependency (<https://pub.dev>) and add this line in pubspec.yaml

```
flutter_icons:  
  android: true  
  image_path: "assets/icon/icon.png"
```

- Run the package by the following command (Go to TERMINAL)

```
flutter pub run flutter_launcher_icons:main
```

## Changing the app icon for iOS

- The icon config files are located inside ios/Runner/Assets.xcassets/AppIcon.appiconset.
- The Contents.json file defines a list of icon images with different sizes and scales.
- The files are located in the same folder. Before updating our application icon to our desired one, It's recommended to follow the guideline for designing the iOS icon.