

API Architecture

API Documentation & API
Documentation Structure
and Criteria



API architecture , API Documentation

- ☐ API Documentation Structure and Criteria
- ☐ Swagger tools
- ☐ Mock API (Swagger CodeGen)

Objektif sesi

- Memahami Konsep Open API
- Memahami Bagaimana cara membuat API Contract
- Memahami Penerapan Swagger pada aplikasi

API Documentation

API documentation memiliki peran penting dalam pengembangan perangkat lunak dan integrasi sistem. Berikut adalah beberapa alasan mengapa API documentation sangat diperlukan:

Sebagai

API documentation memberikan panduan yang jelas tentang cara menggunakan API. Ini memungkinkan pengguna atau pengembang untuk memahami fungsionalitas dan cara berinteraksi dengan API.

Panduan

Penggunaan:

Mengurangi

Kesalahan

Implementasi:

Dengan dokumentasi yang baik, kesalahan implementasi dapat diminimalkan. Pengguna dapat merujuk pada dokumentasi untuk memastikan bahwa mereka menggunakan API dengan benar dan sesuai dengan spesifikasi yang telah ditentukan.

Pemahaman

tentang

Endpoint

dan

Parameter:

Dokumentasi memberikan informasi rinci tentang setiap endpoint API, parameter yang diperlukan, dan respons yang diharapkan. Hal ini membantu pengguna memahami struktur permintaan yang benar dan apa yang diharapkan sebagai tanggapan.

API Documentation Structures

Secara umum, struktur dokumentasi API mencakup beberapa bagian kunci yang memberikan panduan lengkap bagi pengguna dan pengembang. Berikut adalah struktur umum untuk dokumentasi API:

Getting Started

Authentication

Endpoint Reference

Request and Response Examples

API Documentation Structures: Getting Started

Instruksi langkah demi langkah untuk memulai menggunakan API, termasuk cara mendapatkan kunci API, otorisasi, dan permintaan pertama.

Tujuan: Memudahkan pengguna baru untuk mulai menggunakan API dengan cepat.

Get started

Searching for news
articles

Get curated breaking
news headlines

Documentation

Authentication

Endpoints

Everything

Get started

To get started you'll need an API key. They're free while you are in development.

[Get API key](#)

You should know how to make web requests in your chosen programming language. We have included some crude ways to do this in our examples below if you need a place to start. Alternatively you can use one of our [client libraries](#).

API Documentation Structures: Authentication

Informasi tentang cara otentikasi atau mengamankan akses ke API, seperti penggunaan API key, token, atau metode otentikasi lainnya.

Tujuan: Menyediakan petunjuk tentang cara melindungi API dan mengamankan akses.

Get started

Searching for news articles

Get curated breaking news headlines

Documentation

Authentication

Endpoints

Everything

Top headlines

Sources

Errors

Client libraries

Node.js

Ruby

Python

Authentication

Authentication is handled with a simple API key.

They're free while you are in development, and you can get one here:

[Get API key](#)

You can attach your API key to a request in one of three ways:

- Via the `apiKey` querystring parameter.
- Via the `X-API-Key` HTTP header.
- Via the `Authorization` HTTP header. Including `Bearer` is optional, and be sure not to base 64 encode it like you may have seen in other authentication tutorials.

We strongly recommend the either of last two so that your API key isn't visible to others in logs or via request sniffing.

If you don't append your API key correctly, or your API key is invalid, you will receive a `401 - Unauthorized` HTTP error.

Via querystring

```
GET https://newsapi.org/v2/everything?q=keyword&apiKey=API_KEY
```

Via X-API-Key HTTP header

```
X-API-Key: API_KEY
```

Via Authorization HTTP header

```
Authorization: API_KEY
```

API Documentation Structures:

Endpoint References

Daftar lengkap dari setiap endpoint API, dengan deskripsi, parameter yang diperlukan, dan respons yang diharapkan.

Tujuan: Memberikan panduan detail tentang cara menggunakan setiap fitur API.

HTTP method	Metode HTTP request (GET, POST, PUT, PATCH, HEAD, DELETE)
URL	Alamat web yang dituju
URL parameter	Parameter yang melekat di URL sebagai input
Query parameter	Parameter dalam format key-value yang melekat di URL sebagai input
Request header	Key-value yang disertakan ke dalam header, wajib maupun optional
Request body	Key-value yang disertakan ke dalam body, wajib maupun optional

API Documentation Structures:

Endpoint References

All articles about Bitcoin

Definition

```
GET https://newsapi.org/v2/everything?q=bitcoin&apiKey=API_KEY
```

HTTP method	GET
URL	https://newsapi.org
URL parameter	/v2/everything
Query parameter	q & apiKey
Request header	-
Request body	-

API Documentation Structures: Request and Response Examples

Contoh permintaan dan respons untuk membantu pengembang memahami format yang diharapkan.

Tujuan: Memandu pengguna tentang cara merancang permintaan dan menanggapi hasil.

Response Header	Informasi mengenai response
Response Body	Data yang diberikan server
Status Code	Status response, mengikuti standar HTTP status code

API Documentation Structures: Request and Response Examples

Response

200

HEADERS

Content-Type:application/json

BODY

```
01 {
02   "status": "200",
03   "message": "OTP Successfully sent",
04   "otp": "123456"
05 }
```

Response Header

Content-Type:application/json

Response Body

```
{
  "status": "200",
  "message": "OTP Successfully sent",
  "otp": "123456"
}
```

Status Code

200

API architecture , API Documentation



API Documentation Structure and Criteria



Swagger tools



Mock API (Swagger CodeGen)

API Documentation Menggunakan Swagger

Berikut adalah panduan langkah-demi-langkah tentang cara membuat API documentation menggunakan Swagger di sebuah aplikasi.

Swagger merupakan alat yang populer untuk menghasilkan dokumentasi API yang rinci dan interaktif.



Swagger™

Supported by SMARTBEAR

API Documentation Menggunakan Swagger

Langkah 1: Instalasi swagger pada proyek

Anda perlu menginstal paket Swagger di proyek Node.js menggunakan npm atau yarn. Pastikan Anda telah menginstal Swagger pada direktori ***user-service***.



bash

```
1  npm install swagger-ui-express swagger-jsdoc
2
```

API Documentation Menggunakan Swagger

Langkah 2: Konfigurasi Swagger

Buat file konfigurasi Swagger ini biasanya adalah file JavaScript atau JSON yang mendefinisikan informasi tentang API Anda seperti judul, deskripsi, dan definisi endpoint.

*Buat file konfigurasi Swagger di direktori user-service.
Misalnya, swagger-config.js:*

```
1  const swaggerJsdoc = require("swagger-jsdoc");
2
3  const options = {
4    definition: {
5      openapi: "3.0.0",
6      info: {
7        title: "User Service API",
8        version: "1.0.0",
9        description: "API documentation for user service.",
10      },
11    },
12    apis: ["../routes/*.js"],
13  };
14
15  const specs = swaggerJsdoc(options);
16
17  module.exports = specs;
18
```

API Documentation Menggunakan Swagger

Langkah 3: Tambahkan Middleware Swagger ke User Service

Di dalam file index.js di dalam direktori user-service, tambahkan middleware Swagger:

```

1  const express = require("express");
2  const swaggerUi = require("swagger-ui-express");
3  const specs = require("../swagger-config"); // Mengimpor konfigurasi Swagger
4
5  const app = express();
6
7  // Endpoint untuk dokumentasi Swagger
8  app.use("/api-docs", swaggerUi.serve, swaggerUi.setup(specs));
9
10 // Definisikan rute API Anda di sini
11
12 const PORT = process.env.PORT || 3000;
13 app.listen(PORT, () => {
14   console.log(`Server is running on port ${PORT}`);
15 });
16

```


API Documentation Menggunakan Swagger

Langkah 4: Tambahkan Anotasi Swagger ke Endpoint Anda

Di dalam direktori routes dalam user-service, tambahkan anotasi Swagger ke setiap file rute Anda, misalnya ***userRoutes.js***

```
1  /**
2   * @swagger
3   * /users:
4   *   get:
5   *     summary: Get all users
6   *     description: Retrieve a list of all users.
7   *     responses:
8   *       200:
9   *         description: List of users successfully retrieved.
10  *       500:
11  *         description: Internal server error.
12  */
13
```

API Documentation Menggunakan Swagger

Langkah 5: Menjalankan User Service

Dengan langkah-langkah di atas, Anda akan memiliki dokumentasi Swagger yang terintegrasi dengan user-service, dan dapat diakses melalui ***/api-docs*** pada server ***user-service***.

Anda dapat mengulangi proses ini untuk setiap layanan lain yang ingin Anda dokumentasikan.

API architecture , API Documentation



API Documentation Structure and Criteria



Swagger tools



Mock API (Swagger CodeGen)

Swagger Codegen

Swagger Codegen adalah Open-source project yang memungkinkan pengembang menghasilkan kode API dalam berbagai bahasa pemrograman.

Dibutuhkan OpenAPI Specification sebagai masukan, yang menjelaskan endpoints API, format request/response, mekanisme autentikasi, dan banyak lagi.

Dengan ini, Swagger Codegen dapat secara otomatis menghasilkan kode boilerplate yang dapat disesuaikan dan diintegrasikan ke dalam aplikasi Anda.



Swagger CodegenTM

Supported by SMARTBEAR

Swagger Codegen: Benefits

→ **Konsistensi**

Dengan menggunakan Swagger Codegen, Anda dapat memastikan bahwa kode yang dihasilkan untuk klien dan server mengikuti pola yang konsisten sesuai dengan spesifikasi OpenAPI. Ini membantu mencegah kesalahan dan memastikan bahwa semua bagian aplikasi Anda bekerja bersama dengan benar.

→ **Productivity**

Menghasilkan kode klien dan server secara otomatis dari definisi OpenAPI dapat menghemat banyak waktu dan usaha. Anda tidak perlu menulis kode boilerplate yang sama berulang kali, karena sebagian besar kode dapat dibuat secara otomatis.

→ **Maintainability**

Dengan menggunakan Swagger Codegen, Anda dapat memastikan bahwa semua perubahan pada API terdokumentasi dengan baik dan diterapkan secara konsisten di seluruh kode. Ini membantu dalam pemeliharaan jangka panjang aplikasi Anda.

Swagger Codegen: Preparation

Sebelum menggunakan swagger codegen ada satu hal yang harus dipersiapkan, salah satunya adalah membuat json file yang berisikan konfigurasi dari swagger codegen.

Salah satu contoh yang bisa ditiru yaitu <https://petstore.swagger.io/v2/swagger.json>

```
{
  swagger : 2.0
  info : {
    description : This is a sample server Petstore server.
                  You can find out more about Swagger at
                  [http://swagger.io](http://swagger.io) or
                  on [irc.freenode.net, #swagger]
                  (http://swagger.io/irc/). For this
                  sample, you can use the api key `special-
                  key` to test the authorization filters.
    version : 1.0.6
    title : Swagger Petstore
    termsOfService : http://swagger.io/terms/
    contact : { 1 prop }
    license : { 2 props }
  }
  host : petstore.swagger.io
  basePath : /v2
  tags : [ 3 items ]
  schemes : [ 2 items ]
  paths : { 14 props }
  securityDefinitions : { 2 props }
  definitions : { 6 props }
  externalDocs : { 2 props }
}
```

Swagger Codegen: Sample

Buat node server menggunakan swagger codegen dari json yang sudah di generate seperti pada json berikut **<https://petstore.swagger.io/v2/swagger.json>**

```
bash

1  git clone https://github.com/swagger-api/swagger-codegen
2  cd swagger-codegen
3  mvn clean package
4  java -jar modules/swagger-codegen-cli/target/swagger-codegen-cli.jar generate \
5      -i https://petstore.swagger.io/v2/swagger.json \
6      -l nodejs-server \
7      -o samples/server/petstore/nodejs
```

API architecture , API Documentation



API Documentation Structure and Criteria



Swagger tools



Mock API (Swagger CodeGen)



Thank you