Dimas Diandra Audiansyah

185150701111018

Review OpenAPI 3 – Components Object

Link: https://github.com/andrisan-class/review-openapi-3-specification-tim-tiga

Components Object

Disini saya akan menjelaskan tentang components object, bagian dari schema dimana components ini sering digunakan pada operasional API dan menempatkan kode menggunakan $ref untuk memanggil/mereferensikan dari components yang sudah dibuat. Penggunaan components ini biasanya untuk menghindari duplikasi kode dan berfungsi sebagai wadah untuk macam-macam definisi/schema yang dapat digunakan lagi seperti schema, parameters, responses, examples, links, security schemes, request bodies, headers, callbacks dan pathItems.

Definisi yang sudah disebutkan tadi bersifat optional jadi jika dalam schema membutuhkan hanya beberapa saja dari definition tadi maka tinggal memanggilnya dan definisi ini bukan sebuah parameter dan respons yang berlaku untuk semua operasi, melainkan hanya potongan – potongan informasi untuk dipanggil atau direferensikan pada tempat lain.

Type data dan deskripsi dari isi components:

1. Schemas, type data – Map[string, Schema Object]

Deskripsi - Objek untuk menampung Objek Schema yang dapat digunakan kembali.

1. Responses, type data – Map[string, Response Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Response yang dapat digunakan kembali.

1. Parameters, type data – Map[string, Parameter Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Parameter yang dapat digunakan kembali.

1. Examples, type data – Map[string, Example Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Example yang dapat digunakan kembali.

1. requestBodies, , type data – Map[string, Request Body Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Request Body yang dapat digunakan kembali.

1. Headers, type data – Map[string, Header Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Header yang dapat digunakan kembali.

1. SecuritySchemes, type data – Map[string, Security Scheme Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Security Scheme yang dapat digunakan kembali.

1. Links, type data – Map[string, Link Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Link yang dapat digunakan kembali.

1. Callback, type data – Map[string, Callback Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Callback yang dapat digunakan kembali.

1. pathItems, type data – Map[string, Path Item Object ¦ Reference Object]

Deskripsi - Objek untuk menampung Objek Path Item yang dapat digunakan kembali.

Penamaan component hanya bisa terdiri dari karakter berikut:

* A - Z
* a - z
* 0 – 9
* .
* - \_

Contohnya:

* Test
* New\_Test
* org.example.Test
* 401-Unauthorized

Pemanggilan Component dengan mereferensikan component melalui $ref dari bagian lain dari spesifikasi API:

* $ref: ‘#/components/<type>/<name>’ atau $ref: ‘#/components/schemas/Test’

Sebuah component juga dapat ditentukan secara eksternal dimana isi dari konten sudah dinisialisasi dan bisa bisa digunakan didalam component lain dengan menggunakan $ref yang mereferensikan dari eksternal ke internal. Contohnya seperti berikut:

components:

schemas:

Pet:

$ref: '../models/pet.yaml'

# Bisa menggunakan '#/components/schemas/Pet'

User:

$ref: 'https://api.example.com/v2/openapi.yaml#/components/schemas/User'

# Bisa menggunakan '#/components/schemas/User'

responses:

GenericError:

$ref: '../template-api.yaml#/components/responses/GenericError'

# Bisa menggunakan '#/components/responses/GenericError'

Perbedaan dari OpenAPI 2.0 dan OpenAPI 3.0

Dalam OpenAPI 2.0 bagian untuk components memiliki definitions, parameters dan securityDefinitions yang terpisah. Namun pada OpenAPI 3.0 semuanya diatur semua didalam Components dan yang awalnya dengan nama definitions diubah menjadi nama schemas dan securityDefinitions diubah menjadi securitySchemes. Berikut contohnya:

* '#/definitions/User' → '#/components/schemas/User'
* '#/parameters/offsetParam' → '#/components/parameters/offsetParam'