

# DOKUMENTASI SWAGGER PROJECT BLOG GOLANG

Oleh : Ali Murtadho

Dokumentasi ini dirancang dengan swagger dan diakses pada local server dengan menjalankan di <http://localhost:8080/swagger/index.html> dan bisa dilihat untuk repositorynya pada link <https://github.com/ali-murtadho/Blogs-Project-Backend-App>

Pada swagger ini terdapat Restful API untuk Auth User, Category, dan Blog yang saling berelasi. Berikut merupakan daftar isi dari dokumentasi ini.

## DAFTAR ISI

<b>SWAGGER</b>	2
<b>AUTH USER</b>	3
1. Register (POST)	3
2. Login (POST)	4
<b>CATEGORY</b>	5
1. Create Category (POST)	5
2. Get Category By ID (GET)	6
3. Update Category by ID (PATCH)	7
4. Delete Category (DELETE)	8
5. Get All Category (GET)	8
6. Get Blog by Category (GET)	9
<b>BLOG</b>	10
1. Create Blog (POST)	10
2. Get Blog By ID (GET)	11
3. Update Blog by ID (PATCH)	12
4. Get All Blog (GET)	13
5. Delete Blog (DELETE)	14

# SWAGGER

## Auth

**POST** /login Login as a User

**POST** /register Register as a new User

## Models

controllers.BlogInput >

controllers.CategoryInput >

controllers.LoginInput >

controllers.RegisterInput >

models.Blog >

models.Category >

## blog

**GET** /blogs Get All blog

**POST** /blogs Create blog

**GET** /blogs/{id} Get blog

**DELETE** /blogs/{id} Delete blog

**PATCH** /blogs/{id} Update blog

## Category

**GET** /categories Get All List of Category

**POST** /categories Create Category

**GET** /categories/{id} Get Category by Id

**DELETE** /categories/{id} Delete Category by Id

**PATCH** /categories/{id} Update Category by Id

**GET** /categories/{id}/blog Get blog by category by id

# AUTH USER

## 1. Register (POST)

**POST** /register Register as a new User

Register new user

Parameters Cancel

Name	Description
<b>Body</b> <span>required</span>	the body to Register
object (body)	<div>Edit Value   Model</div> <pre>{   "email": "ali@gmail.com",   "password": "123",   "username": "ali" }</pre>

Cancel

Parameter content type  
application/json

Execute Clear

Responses Response content type application/json

Curl

```
curl -X 'POST' \
  'http://localhost:8080/register' \
  -H 'accept: application/json' \
  -H 'Content-type: application/json' \
  -d '{
    "email": "ali@gmail.com",
    "password": "123",
    "username": "ali"
  }'
```

Request URL

```
http://localhost:8080/register
```

Server response

Code	Details
200	<div>Response body<pre>{   "message": "Registrasi berhasil",   "user": {     "email": "ali@gmail.com",     "username": "ali"   } }</pre><span>Download</span></div> <div>Response headers<pre>content-length: 83 content-type: application/json; charset=utf-8 date: Tue, 17 Oct 2023 03:56:21 GMT</pre></div>

Responses

Code	Description
200	OK

## 2. Login (POST)

**Auth**

**POST** /login Login as a User

this func used to login to sistem by user by get jwt token and by authorization

**Parameters** Cancel

Name	Description
<b>Body</b> * required	the body to login
object (body)	<div>Edit Value   Model</div> <pre>{   "password": "123",   "username": "ali" }</pre>

Cancel

Parameter content type  
application/json

**Responses** Response content type application/json

**Curl**

```
curl -X 'POST' \
  'http://localhost:8080/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "password": "123",
    "username": "ali"
  }'
```

**Request URL**  
http://localhost:8080/login

**Server response**

Code	Details
200	<div><b>Response body</b><pre>{   "message": "Login berhasil!",   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdGRob3JpemVkIjp0cnV1LCJ1eHAiOiJlMTg2NDYsInVzZXI6ImQ1OjN9.-Ou3UYXPD_XCbMBbXoGiU7qv4b1dHa0YH13NqeVXgEU",   "user": {     "username": "ali"   } }</pre><span>Download</span></div> <div><b>Response headers</b><pre>content-length: 210 content-type: application/json; charset=utf-8 date: Tue, 17 Oct 2023 03:56:46 GMT</pre></div>

**Responses**

Code	Description
200	OK

Example Value | Model

# CATEGORY

## 1. Create Category (POST)

POST

/categories

Create Category

^

Create new Category

Parameters

Try it out

Name	Description
<b>Body</b> <small>* required</small>	kategori berhasil dibuat
object	Example Value   Model
(body)	<pre>{   "cat_type": "news" }</pre>
Parameter content type	<div>application/json</div>

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
  'http://localhost:8080/categories' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "cat_type": "news"
  }'
```

Request URL

Parameter content type application/json

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
  'http://localhost:8080/categories' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "cat_type": "news"
  }'
```

Request URL

```
http://localhost:8080/categories
```

Server response

Code	Details
200	<div>Response body<pre>{   "data": {     "id": 3,     "cat_type": "news"   } }</pre></div> <div>Response headers<pre>content-length: 35 content-type: application/json; charset=utf-8 date: Tue, 17 Oct 2023 03:57:27 GMT</pre></div>

Responses

## 2. Get Category By ID (GET)

GET

/categories/{id} Get Category by Id

Get One Category by Id

Parameters

Try it out

Name	Description
id * required	Category Id
string (path)	<input type="text" value="3"/>

Responses

Response content type application/json

Curl

curl -X 'GET' \n 'http://localhost:8080/categories/3' \n -H 'accept: application/json'

Request URL

http://localhost:8080/categories/3

Server response

Code	Details
200	<div>Response body</div> <div>{\n  "data": {\n    "id": 3,\n    "cat_type": "news"\n  }\n}</div>

Responses

Response content type application/json

Curl

curl -X 'GET' \n 'http://localhost:8080/categories/3' \n -H 'accept: application/json'

Request URL

http://localhost:8080/categories/3

Server response

Code	Details
200	<div>Response body</div> <div>{\n  "data": {\n    "id": 3,\n    "cat_type": "news"\n  }\n}</div> <div>Response headers</div> <div>content-length: 35\ncontent-type: application/json; charset=utf-8\ndate: Tue,17 Oct 2023 03:57:42 GMT</div>

Responses

Code	Description
200	OK

Example Value | Model

{\n "cat\_type": "string",\n "id": 0\n}

### 3. Update Category by ID (PATCH)

**PATCH** /categories/{id} Update Category by Id

Update One Category by Id

Parameters

Try it out

Name	Description
<b>id</b> * required string (path)	Category Id

**Body** \* required  
object  
(body)  
the body to update new category

Example Value | Model

```
{
  "cat_type": "Headline News"
}
```

Parameter content type  
application/json

Responses

Response content type application/json

Curl

```
curl -X 'PATCH' \
  'http://localhost:8080/categories/3' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "cat_type": "Headline News"
  }'
```

Responses

Response content type application/json

Curl

```
curl -X 'PATCH' \
  'http://localhost:8080/categories/3' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "cat_type": "Headline News"
  }'
```

Request URL

http://localhost:8080/categories/3

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "data": {     "id": 3,     "cat_type": "Headline News"   } }</pre></div> <div>Download</div> <div>Response headers</div> <div><pre>content-length: 44 content-type: application/json; charset=utf-8 date: Tue, 17 Oct 2023 03:58:10 GMT</pre></div>

Responses

Code	Description
200	OK

Example Value | Model

```
{
  "cat_type": "string",
  "id": 3
}
```

## 4. Delete Category (DELETE)

**DELETE** /categories/{id} Delete Category by Id

Delete One Category by Id

Parameters

Try it out

Name	Description
<b>id</b> * required string (path)	Category Id <input type="text" value="3"/>

Responses

Response content type application/json

Curl

```
curl -X 'DELETE' \
  'http://localhost:8080/categories/3' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/categories/3
```

Server response

Code	Details
200	<div>Response body</div> <pre>{   "data": true }</pre> <div>Download</div>

## 5. Get All Category (GET)

**GET** /categories Get All List of Category

Get List of Category

Parameters

Try it out

No parameters

Responses

Response content type application/json

Curl

```
curl -X 'GET' \
  'http://localhost:8080/categories' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/categories
```

Server response

Code	Details
200	<div>Response body</div> <pre>{   "data": [     {       "id": 1,       "cat_type": "pendidikan"     },     {       "id": 2,       "cat_type": "teknologi"     }   ] }</pre> <div>Download</div>



## 6. Get Blog by Category (GET)

GET /categories/{id}/blog Get blog by category by id

Get all blog from category by id

Parameters

Try it out

Name	Description
id * required	Category Id
string (path)	<input type="text" value="2"/>

Responses

Response content type application/json

Curl

```
curl -X 'GET' \
'http://localhost:8080/categories/2/blog' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/categories/2/blog
```

Server response

Code	Details
200	<div>Response body<pre>{   "data": [     {       "id": 2,       "title": "Judul 2",       "text": "Text 2",       "created_at": "2023-10-17T10:59:17.547+07:00",       "updated_at": "2023-10-17T10:59:17.547+07:00",       "category_id": 2     }   ] }</pre></div> <div>Download</div>

Response headers

```
content-length: 159
content-type: application/json; charset=utf-8
date: Tue, 17 Oct 2023 04:00:10 GMT
```

Responses

Code	Description
200	OK

Example Value | Model

```
{
  "category_id": 0,
  "created_at": "string",
  "id": 0,
  "text": "string",
  "title": "string",
  "updated_at": "string"
}
```

Auth

Module

# BLOG

## 1. Create Blog (POST)

POST

/blogs

Create blog

Create new blog

Parameters

Try it out

Name	Description
<b>Body</b> <small>* required</small>	the body to create a new blog
object (body)	Example Value   Model
<pre>{   "category_id": 2,   "text": "text 2",   "title": "Judul 2" }</pre>	
Parameter content type application/json	

Responses

Response content type application/json

Curl

```
curl -X 'POST' \
  'http://localhost:8080/blogs' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "category_id": 2,
    "text": "text 2",
    "title": "Judul 2"
  }'
```

Request URL

Server response

Code	Details
200	<div><div>Response body</div><pre>{   "data": {     "id": 2,     "title": "Judul 2",     "text": "text 2",     "created_at": "2023-10-17T10:59:17.547+07:00",     "updated_at": "2023-10-17T10:59:17.547+07:00",     "category_id": 2   } }</pre><div>Download</div></div> <div><div>Response headers</div><pre>content-length: 157 content-type: application/json; charset=utf-8 date: Tue, 17 Oct 2023 03:59:17 GMT</pre></div>

Responses

Code

Description

| 200 | OK |

Example Value | Model

## 2. Get Blog By ID (GET)

GET

/blogs/{id} Get blog

Get One blog by Id

Parameters

Try it out

Name	Description
id * required	blog id
string (path)	<input type="text" value="1"/>

Responses

Response content type application/json

Curl

```
curl -X 'GET' \
'http://localhost:8080/blogs/1' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/blogs/1
```

Server response

Code	Details
200	<div>Response body<pre>{   "data": {     "id": 1,     "title": "judul 1",   } }</pre></div>

string (path)

Responses

Response content type application/json

Curl

```
curl -X 'GET' \
'http://localhost:8080/blogs/1' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/blogs/1
```

Server response

Code	Details
200	<div>Response body<pre>{   "data": {     "id": 1,     "title": "judul 1",     "text": "text 1",     "created_at": "2023-10-17T10:58:46.259+07:00",     "updated_at": "2023-10-17T10:58:46.259+07:00",     "category_id": 1   } }</pre></div> <div>Response headers<pre>content-length: 157 content-type: application/json; charset=utf-8 date: Tue, 17 Oct 2023 03:59:00 GMT</pre></div>

Responses

Code	Description
200	OK

### 3. Update Blog by ID (PATCH)

PATCH

/blogs/{id} Update blog

Update blog by id

Parameters

Try it out

Name	Description
<b>id</b> * required string (path)	blog Id <input type="text" value="1"/>

**Body** \* required  
object  
(body)  
the body to update new blog  
Example Value | Model

```
{
  "category_id": 1,
  "text": "text baru 1",
  "title": "judul baru 1"
}
```

Parameter content type

Responses

Response content type application/json

Curl

```
curl -X 'PATCH' \
  'http://localhost:8080/blogs/1' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "category_id": 1,
    "text": "text baru 1",
    "title": "judul baru 1"
  }'
```

```
{
  "category_id": 1,
  "text": "text baru 1",
  "title": "judul baru 1"
}
```

Parameter content type

Responses

Response content type application/json

Curl

```
curl -X 'PATCH' \
  'http://localhost:8080/blogs/1' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "category_id": 1,
    "text": "text baru 1",
    "title": "judul baru 1"
  }'
```

Request URL

Server response

Code

Details

200

Response body

```
{
  "data": {
    "id": 1,
    "title": "judul baru 1",
    "text": "text baru 1",
    "created_at": "2023-10-17T10:58:46.259+07:00",
    "updated_at": "2023-10-17T10:59:50.107+07:00",
    "category_id": 1
  }
}
```

Download

## 4. Get All Blog (GET)

GET

/blogs

Get All blog

⬆

Get List of blog

Parameters

Try it out

No parameters

Responses

Response content type

application/json

⌵

Curl

```
curl -X 'GET' \
  'http://localhost:8080/blogs' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/blogs
```

Server response

Code

Details

200

Response body

```
{
  "data": [
    {
      "id": 1,
      "title": "judul baru 1",
      "text": "text baru 1",
      "created_at": "2023-10-17T10:58:46.259+07:00",
      "updated_at": "2023-10-17T10:59:58.107+07:00",
      "category_id": 1
    },
    {
      "id": 2,
```

Responses

Response content type

application/json

⌵

Curl

```
curl -X 'GET' \
  'http://localhost:8080/blogs' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8080/blogs
```

Server response

Code

Details

200

Response body

```
{
  "data": [
    {
      "id": 1,
      "title": "judul baru 1",
      "text": "text baru 1",
      "created_at": "2023-10-17T10:58:46.259+07:00",
      "updated_at": "2023-10-17T10:59:58.107+07:00",
      "category_id": 1
    },
    {
      "id": 2,
      "title": "judul 2",
      "text": "text 2",
      "created_at": "2023-10-17T10:59:17.547+07:00",
      "updated_at": "2023-10-17T10:59:17.547+07:00",
      "category_id": 2
    }
  ]
}
```

Response headers

```
content-length: 318
content-type: application/json; charset=utf-8
date: Tue, 17 Oct 2023 04:00:58 GMT
```

Responses

## 5. Delete Blog (DELETE)

**DELETE** /blogs/{id} Delete blog

Delete One blog by Id

**Parameters** Try it out

Name	Description
<b>id</b> * required string (path)	blog Id

**Responses** Response content type application/json

Curl

```
curl -X 'DELETE' \
'http://localhost:8080/blogs/2' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8080/blogs/2
```

Server response

Code	Details
200	<div><p>Response body</p><pre>{   "data": true }</pre><span>Download</span></div>