



Project UAS pemrograman  
berbasis platform

- MUHAMAD ILHAM FAUZY – 20240040035
- MOHAMAD IBNU NIZAR HUSAINI – 20240040034
- EXZHA FAUZAN SOPIAN - 2024004168

## Contents

Latar belakang dan overview .....	3
Tech stack, library, dan framework.....	3
Rancangan sistem .....	4
Entity Relational Database .....	4
Dokumentasi endpoint API .....	5
Daftar daftar endpoint API.....	5
Daftar Tabel Endpoint API.....	6
Pengujian endpoint API.....	19
Account .....	19
Notes.....	21
Pokemon .....	22

## Latar belakang dan overview



Figure 1 Character name: Haru Urara

Project UAS kami memiliki 2 tema, yaitu Simple Note App dan Pokémon Database. Data Pokémon akan diambil dari API eksternal, yaitu PokeAPI, dengan total 20 API endpoint yang terbagi menjadi 4 jenis, yaitu "/account", "/notes", "/pokemon", dan "/". Pengujian akan dilakukan menggunakan Postman, dan dokumentasi lengkap akan dibuat menggunakan Swagger untuk kemudahan penggunaan.

## Tech stack, library, dan framework

kami menggunakan beberapa library maupun framework diantara lainnya:

- nodeJS
- expressJS
- Jsonwebtoken
- Bcrypt
- Pg dan mysql2 (database)
- Morgan (logging)
- Swagger (documentation)
- Uuid (uuid generator)
- Dotenv



Figure 2 Character name: Cheval Grand

## Rancangan sistem



Figure 3 Character name: Oguri Cap

Tema project: simple note app dan pokemon database

Database: postgresql dan mysql

External API: [pokeapi](https://pokeapi.co/)

Fitur: auth, clean architecture, standardize error handling and respond handling, swagger documentation

## Entity Relational Database

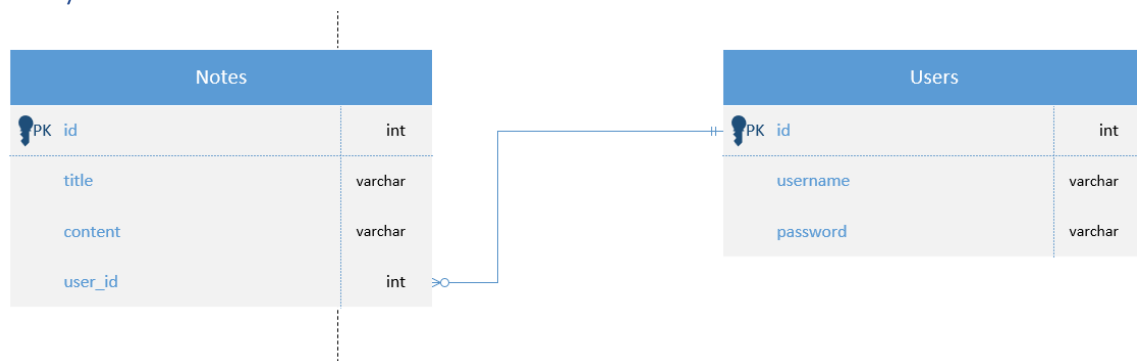


Table 1 ERD

### 1. Users

1.1.Id: primary key untuk users

1.2.Username

1.3.Password

### 2. Notes

2.1.Id: primary key untuk notes

2.2.Title: judul note

2.3.Content: isi note

2.4.User\_id: menentukan note kepemilikan note(foreign key one to many)

# Dokumentasi endpoint API

Menggunakan REST api dan http methods



Figure 4 Character name: Tamamo Cross

## Daftar daftar endpoint API

/account:

- POST /signup
- POST /login
- POST /getAccounts
- GET /getAllAccounts

/notes:

- GET
- POST
- PUT
- DELETE

/:

- GET

/pokemon:

## Daftar Tabel Endpoint API

No	Endpoint	Parameter or Body	Respond
1	POST /login	<pre>{   "username": "string",   "password": "string" }</pre>	<pre>{   "status": "success",   "statusCode": 200,   "message": "Login successful",   "data": {     "token": "YOUR_TOKEN"   },   "timeStamp": "TIME_STAMP",   "requestId": "YOUR_REQID" }</pre>
2	POST /signup	<pre>{   "username": "string",   "password": "string" }</pre>	<pre>{   "status": "success",   "statusCode": 201,   "message": "Signup completed successfully",   "data": {},   "timeStamp": "TIME_STAMP",   "requestId": "YOUR_REQID" }</pre>

			}
3	POST /getAccounts	{ "username": "string" }	{ "status": "success", "statusCode": 200, "message": "Login successful", "data": { "id": ID, "username": "USERNAME", "password": "PASSWORD" }, "timeStamp": "TIME_STAMP", "requestId": "YOUR_REQID" }
4	GET /getAllAccounts	-	{ "status": "success", "statusCode": 200, "message": "Login successful", "data": { "id": ID, "username": "USERNAME",

			<pre> "password": "PASSWORD" }, "timeStamp": "TIME_STAMP", "requestId": "YOUR_REQID" } </pre>
5	GET /notes	-	<pre> { "status": "success", "statusCode": 200, "message": "Login successful", "data": "timeStamp": "TIME_STAMP", "requestId": "YOUR_REQID" } </pre>
6	POST /notes	<pre> { "title": "string", "content": "string" } </pre>	<pre> { "status": "success", "statusCode": 200, "message": "Login successful", "data": "timeStamp": "TIME_STAMP", </pre>



			<pre>"requestId": "YOUR_REQID" }</pre>
7	PUT /notes	<pre>Query.id: string {   "title": "string",   "content": "string" }</pre>	<pre>{   "status": "success",   "statusCode": 200,   "message": "Login successful",   "data":   "timeStamp": "TIME_STAMP",   "requestId": "YOUR_REQID" }</pre>

8	DELETE /notes	Query.id: string	<pre>{   "status": "success",   "statusCode": 200,   "message": "Note has been deleted successfully",   "data":   "imestamp": "TIME_STAMP",   "requestId": "YOUR_REQID" }</pre>
9	GET /	-	<pre>{   "status": "success",   "statusCode": 200,   "message": "Login successful",   "data": "&lt;html&gt;&lt;h1&gt;Selamat datang di page utama!&lt;/h1&gt;  &lt;p&gt;ini adalah project uas matkul pemrograman berbasis platform&lt;/p&gt;  &lt;p&gt;Anggota: &lt;p&gt;</pre>

			<pre> &lt;ul&gt;   &lt;li&gt;Muhamad Ilham Fauzy&lt;/li&gt;&lt;li&gt;Mohamad Ibnu Nizar Husaini&lt;/li&gt;&lt;li&gt;Exza Fauzan Sofyan&lt;/li&gt; &lt;/li&gt;&lt;/html&gt;" "timeStamp": "TIME_STAMP", "requestId": "YOUR_REQID" } </pre>
10	GET /pokemon	Query.pokemon: string	<pre> {   "status": "success",   "statusCode": 200,   "message": "Login successful",   "data":     "timeStamp": "TIME_STAMP",     "requestId": "YOUR_REQID" } </pre>

11	GET /pokemon/abilities	Query.ability: string	<pre>{   "status": "success",   "statusCode": 200,   "message": "Data successfully retrieved",   "data":   "timeStamp": "2026-01-25T13:33:15.445Z",   "requestId": "019bf55b-d735-7030-a6fc-f04c6bd1c680" }</pre>
12	GET /pokemon/types	Query.type: string	<pre>{   "status": "success",   "statusCode": 200,   "message": "Data successfully retrieved",   "data": {   "timeStamp": "2026-01-25T13:42:07.559Z",   "requestId": "019bf563-f5c7-725d-9407-0c87a19e94bc" }</pre>

			}
13	GET /pokemon/genders	Query.gender: string	{ "status": "success", "statusCode": 200, "message": "Data successfully retrieved", "data": { }, "timestamp": "2026-01-25T13:43:06.433Z", "requestId": "019bf564-dbc1-726e-99ee-e14158c5fe4b" }

14	GET /pokemon/stats	Query.stat: string	{ "status": "success", "statusCode": 200, "message": "Data successfully retrieved", "data": { "timestamp": "2026-01-25T13:43:43.539Z", "requestId": "019bf565-6cb3-7689-b891-bff59b0367a1" } }
15	GET /pokemon/characteristics	Query.characteristic: string	{ "status": "success", "statusCode": 200, "message": "Data successfully retrieved", "data": { "timestamp": "2026-01-25T13:44:15.524Z", "requestId": "019bf565-e9a4-76fd-b602-f423035363df" } }

16	GET /pokemon/natures	Query.nature: string	{ "status": "success", "statusCode": 200, "message": "Data successfully retrieved", "data": { "timeStamp": "2026-01-25T13:44:55.413Z", "requestId": "019bf566-8575-75d9-b58a-b3cc7b0263b0" } }

17	GET /pokemon/genetrations	Query.generation: string	{ "status": "success", "statusCode": 200, "message": "Data successfully retrieved", "data": { "timestamp": "2026-01-25T13:45:34.390Z", "requestId": "019bf567-1db6-77cc-83a4-73c5c4921629" } }
18	GET /pokemon/version	Query.version: string	{ "status": "success", "statusCode": 200, "message": "Data successfully retrieved", "data": { "timestamp": "2026-01-25T13:46:09.768Z", "requestId": "019bf567-a7e8-723d-9779-39f884732f4c" } }



19	GET /pokemon/items	Query.item: string	{ "status": "success", "statusCode": 200, "message": "Data successfully retrieved", "data": { "timeStamp": "2026-01-25T13:46:35.922Z", "requestId": "019bf568-0e12-7230-b93b-ce2e5dbd68cd" } }

20	GET /pokemon/location	Query.localtion: string	{ "status": "success", "statusCode": 200, "message": "Data successfully retrieved", "data": { "timeStamp": "2026-01-25T13:48:05.895Z", "requestId": "019bf569-6d87-7633-afd7-cdf6fb1c9708" } }
----	-----------------------	-------------------------	--

# Pengujian endpoint API

## Account

### /Account/login

The screenshot shows a REST client interface with a POST request to `localhost:3000/account/login`. The request body is a JSON object: `{ "username": "kermit", "password": "123" }`. The response is a 200 OK status with a response time of 113 ms and a body size of 560 B. The response body is a JSON object: `{ "status": "success", "statusCode": 200, "message": "Login successful", "data": { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ3ZCI6Im1pdCIsIm1hdiI6IjI0MTM1ODU3MCw1ZXhwIjoxNzY2MzU0MTcwFQ.UPC70U-ac2U2k532W_-W-7jQdEpm7kRRVcGe7boxPXo", "timestamp": "2026-01-25T14:16:10.937Z", "requestId": "019bf583-23b9-756b-bebc-4939b4fc2983" } }`.

### /account/signup

The screenshot shows a REST client interface with a POST request to `localhost:3000/account/signup`. The request body is a JSON object: `{ "username": "kermit5", "password": "123" }`. The response is a 201 Created status with a response time of 60 ms and a body size of 420 B. The response body is a JSON object: `{ "status": "success", "statusCode": 201, "message": "Signup completed successfully", "data": {}, "timestamp": "2026-01-25T14:17:36.663Z", "requestId": "019bf584-7297-73ed-9901-8f2e78411d82" }`.

### /account/getaccounts

The screenshot shows a REST client interface with a POST request to `localhost:3000/account/getAccounts`. The request body is a JSON object: `{ "username": "kermit5" }`. The response is a 200 OK status with a response time of 37 ms and a body size of 518 B. The response body is a JSON object: `{ "status": "success", "statusCode": 200, "message": "Data retrieved successfully", "data": [ { "id": 11, "username": "kermit5", "password": "$2b$8$FroY51g15umYK4xa.L.cu831VGxztA3na2L8jM1kcq@WuECKn1W" } ], "timestamp": "2026-01-25T14:19:51.642Z", "requestId": "019bf586-61da-746a-a7b7-8ae08521f159" }`.

/account/getallaccounts

localhost:3000/account/getallAccounts

GET localhost:3000/account/getallAccounts

Body

JSON

```
1 {
2   "status": "success",
3   "statusCode": 200,
4   "message": "Successfully fetch data",
5   "data": [
6     {
7       "id": 4,
8       "username": "kermit",
9       "password": "$2b$8$SymCHZYSQZVLsHwQ3M/7pE./0a2d7zL37Wfz1fgt.as5YvaoiVcf.m"
10    },
11    {
12      "id": 10,
13      "username": "kermit2",
14      "password": "$2b$8$8xy4ElytEfZJ5B4Q.8yV33e2Z.bawz/XttrZth.syFaltPAaSzcz6i"
15    },
16    {
17      "id": 11,
18      "username": "kermit5",
19      "password": "$2b$8$8FroY51g15umYK4xa.L.cu831VGxztA3na2L6jMIkcq8wuECKnlw"
20    }
21  ],
22   "timeStamp": "2826-81-25T14:29:33.779Z",
23   "requestId": "619b5b7-2673-7bde-b5a8-b7ac8bf4f81e"
24 }
```

localhost:3000/

GET localhost:3000/

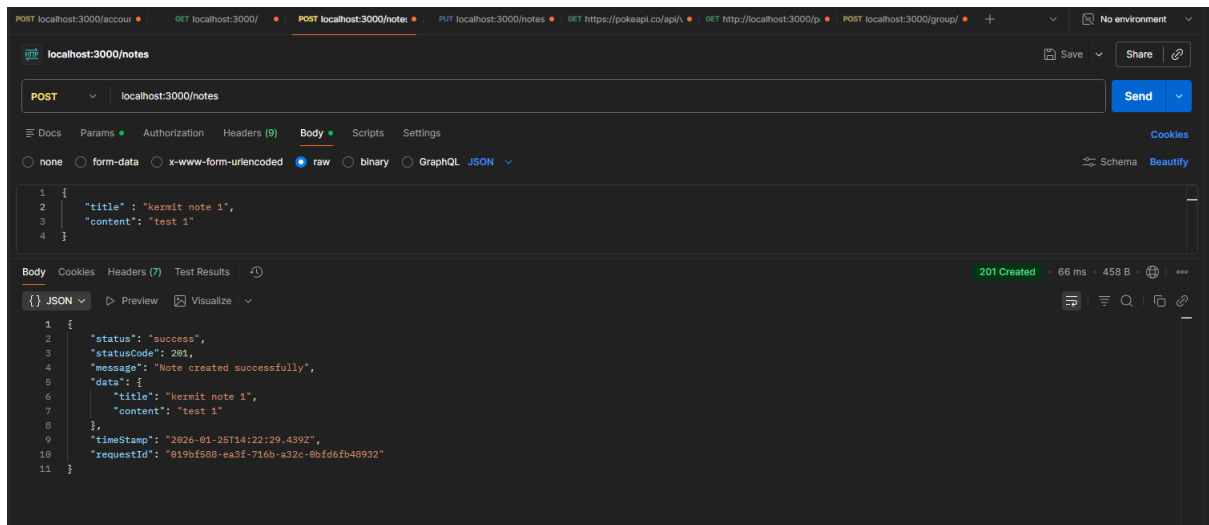
Body

HTML

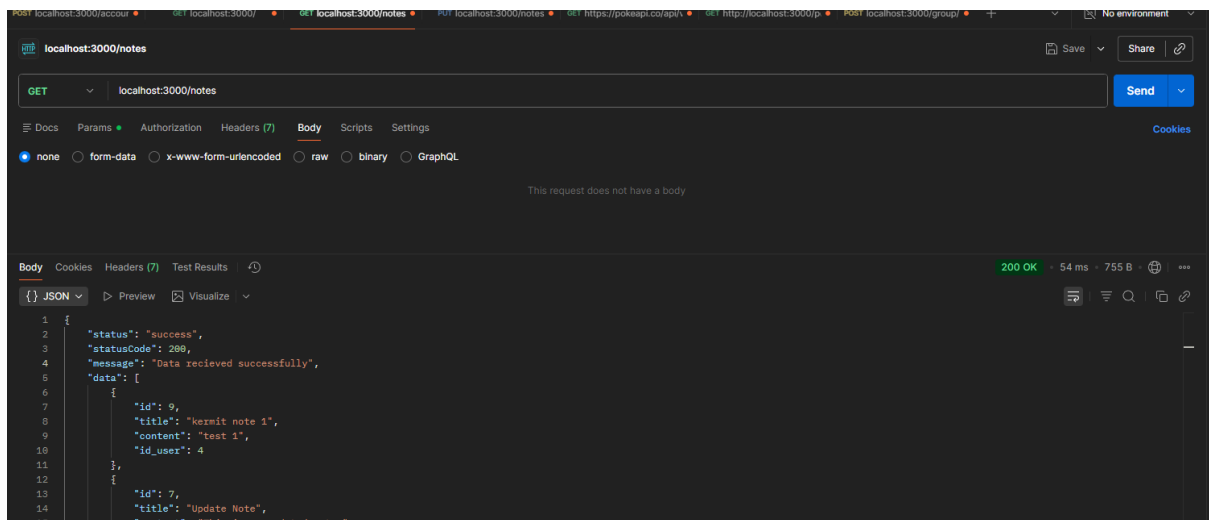
```
1 <html>
2 <h1>Selamat datang di page utama!</h1>
3 <p>ini adalah project uas matkul pemrograman berbasis platform</p>
4 <p>Anggota:
5 <p>
6 <ul>
7   <li>Muhamad Ilham Fauzy
8   <li>
9   <li>Mohamad Ibnu Nizar Husaini
10  <li>
11  <li>Exza Fauzan Sofyan
12  <li>
13 </li>
14 </ul>
15 </p>
16 </html>
```

# Notes

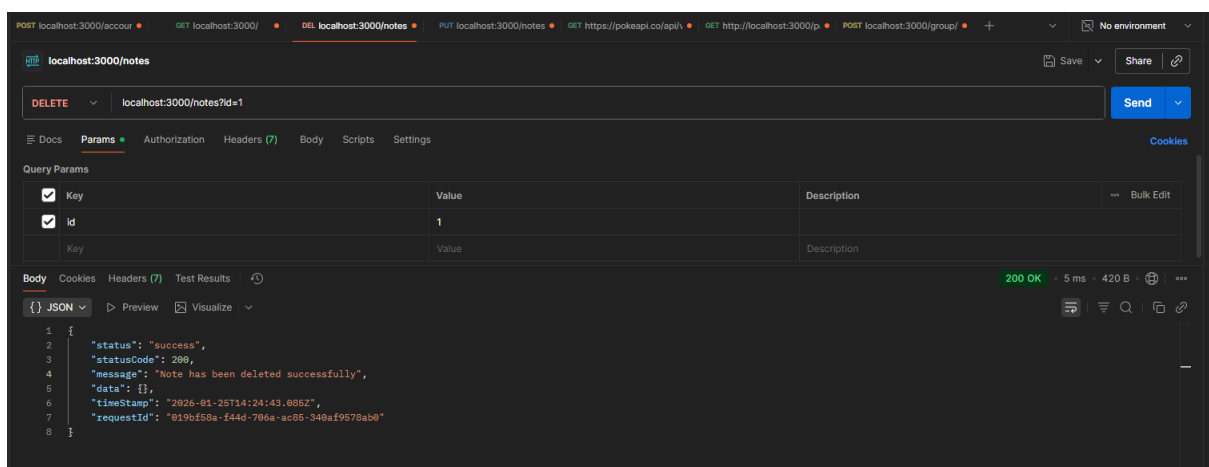
## POST /note



## GET /notes



## DELETE /notes



## PUT /notes

localhost:3000/notes

PUT localhost:3000/notes?id=8

Body

```
1 {
2   "title": "kermit note 1",
3   "content": "test 1"
4 }
```

Body

```
1 {
2   "status": "success",
3   "statusCode": 200,
4   "message": "Note successfully edited",
5   "data": {
6     "title": "kermit note 1",
7     "content": "test 1"
8   },
9   "timeStamp": "2026-01-25T14:25:21.386Z",
10  "requestId": "019bf58b-899a-7449-9b15-24862daee8f6"
11 }
```

200 OK · 47 ms · 452 B

## Pokemon

Abilities pokemon contoh: pikachu with electro ball

GET http://localhost:3000/pokemon/abilities?ability=static

Headers (6)

Key	Value	Description
6 hidden		

Body

```
1 {
2   "status": "success",
3   "statusCode": 200,
4   "message": "Data successfully retrieved",
5   "data": {
6     "effect_changes": [
7       {
8         "effect_entries": [
9           {
10            "effect": "N'a pas d'effet en dehors du combat.",
11            "language": {
12              "name": "fr",
```

200 OK · 99 ms · 28.03 KB

Types pokemon contoh: Fire

GET <http://localhost:3000/pokemon/types?type=fire> Send

Docs Params Authorization Headers (6) Body Scripts Settings Cookies

Headers 6 hidden

Key	Value	Description
-----	-------	-------------

Body Cookies Headers (7) Test Results 200 OK • 109 ms • 19.57 KB

JSON Preview Visualize

```
1 {
2   "status": "success",
3   "statusCode": 200,
4   "message": "Data successfully retrieved",
5   "data": {
6     "damage_relations": {
7       "double_damage_from": [
8         {
9           "name": "ground",
10          "url": "https://pokeapi.co/api/v2/type/5/"
11        },
12        {
13          "name": "steel",
14          "url": "https://pokeapi.co/api/v2/type/9/"
15        }
16      ]
17    }
18  }
```

Gender pokemon

GET <http://localhost:3000/pokemon/genders?gender=female> Send

Docs Params Authorization Headers (6) Body Scripts Settings Cookies

Headers 6 hidden

Key	Value	Description
-----	-------	-------------

Body Cookies Headers (7) Test Results 200 OK • 101 ms • 85.77 KB

JSON Preview Visualize

```
1 {
2   "status": "success",
3   "statusCode": 200,
4   "message": "Data successfully retrieved",
5   "data": {
6     "id": 1,
7     "name": "female",
8     "pokemon_species_details": [
9       {
10        "pokemon_species": {
11          "name": "bulbasaur",
12          "url": "https://pokeapi.co/api/v2/pokemon-species/1/"
13        }
14      }
15    ]
16  }
```

Stats pokemon

GET <http://localhost:3000/pokemon/stats?stat=attack> Send

Docs Params Authorization Headers (6) Body Scripts Settings Cookies

Headers 6 hidden

Key	Value	Description
-----	-------	-------------

Body Cookies Headers (7) Test Results 200 OK • 88 ms • 6.59 KB

JSON Preview Visualize

```
1 {
2   "status": "success",
3   "statusCode": 200,
4   "message": "Data successfully retrieved",
5   "data": {
6     "affecting_items": [
7       {
8         "name": "protein",
9         "url": "https://pokeapi.co/api/v2/item/46/"
10      },
11      {
12        "name": "x-attack",
13        "url": "https://pokeapi.co/api/v2/item/57/"
14      }
15    ]
16  }
```

### Characteristic pokemon

URL: `http://localhost:3000/pokemon/characteristics`

Method: **GET**

URL: `http://localhost:3000/pokemon/characteristics?characteristic=1`

Status: **200 OK** (89 ms, 1.64 KB)

Body (JSON):

```
1 {
2   "status": "success",
3   "statusCode": 200,
4   "message": "Data successfully retrieved",
5   "data": {
6     "descriptions": [
7       {
8         "description": "たべるのが だいすき",
9         "language": {
10          "name": "ja-Hrkt",
11          "url": "https://pokeapi.co/api/v2/language/1/"
12        }
13      }
14     ]
15   }
16 }
```

### natures contoh: daerah tempat pokemon berasal

URL: `http://localhost:3000/pokemon/natures`

Method: **GET**

URL: `http://localhost:3000/pokemon/natures?nature=bold`

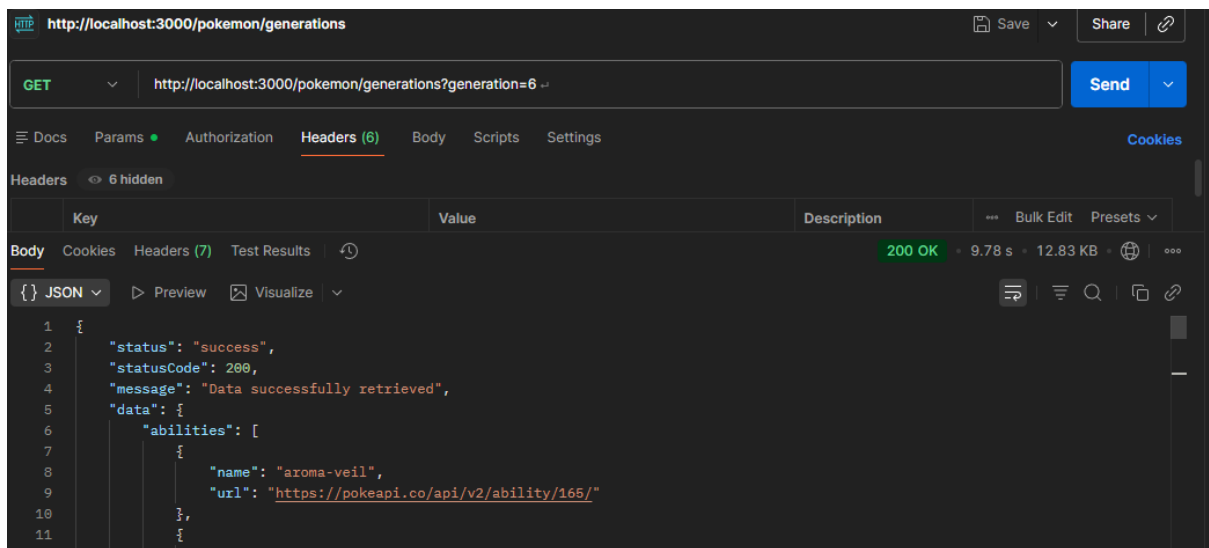
Status: **200 OK** (354 ms, 2.32 KB)

Body (JSON):

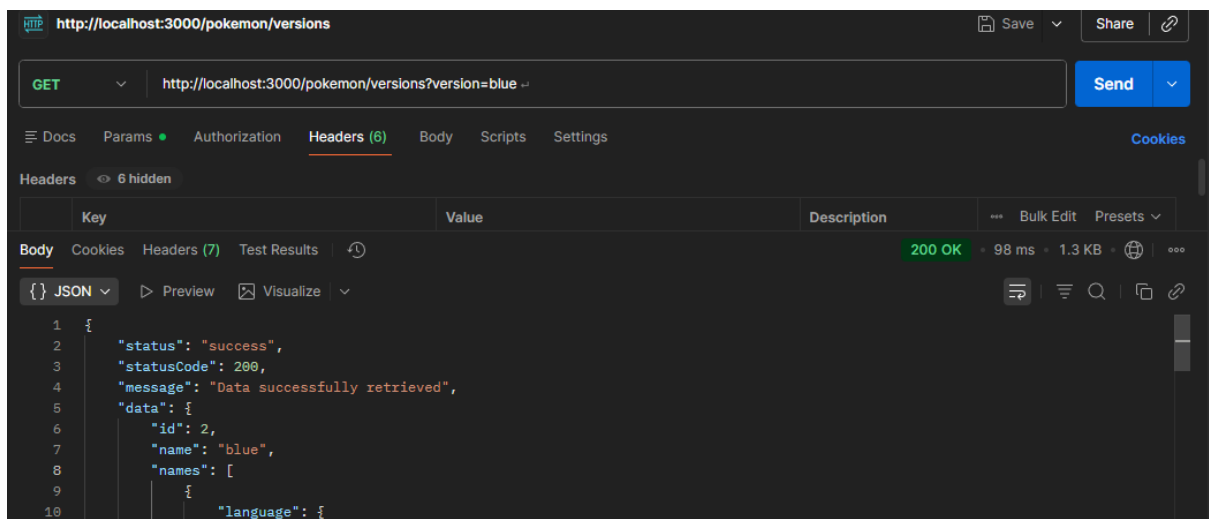
```
82   "url": "https://pokeapi.co/api/v2/language/6/"
83   },
84   "name": "Kühn"
85 },
86 {
87   "language": {
88     "name": "es",
89     "url": "https://pokeapi.co/api/v2/language/7/"
90   },
91   "name": "Osada"
92 },
93 }
```



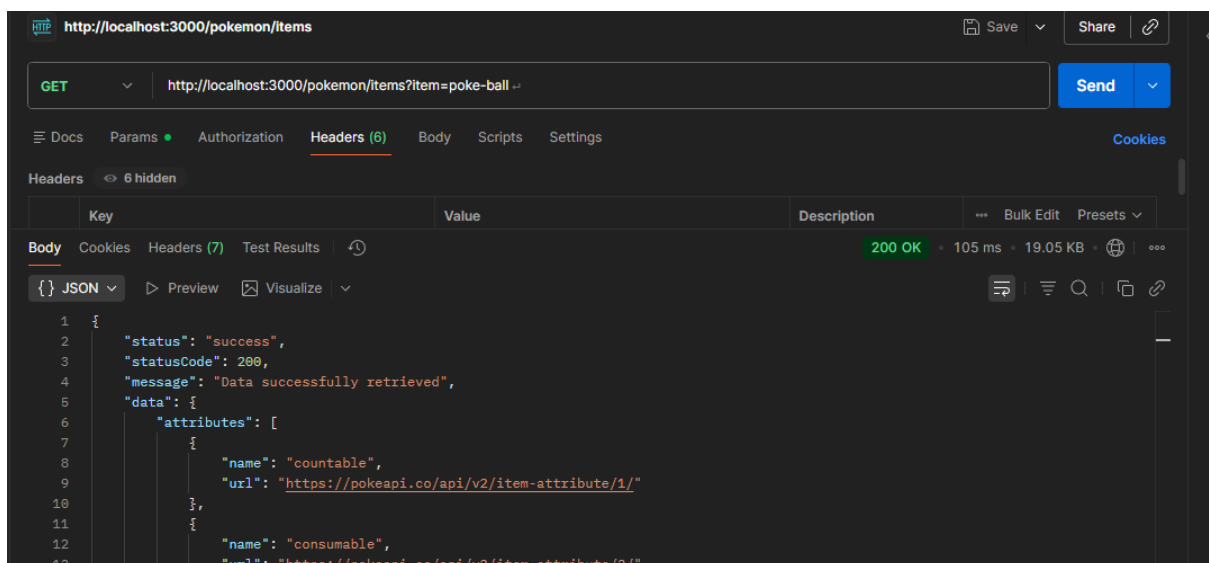
generation contoh: pokemon generasi berapa



version contoh: versi berapa game pokemon ini



items contoh: pokeball



location contoh: daerah pokemon

The screenshot shows a web browser interface with the URL `http://localhost:3000/pokemon/locations`. The request method is `GET` and the full URL is `http://localhost:3000/pokemon/locations?location=sinnoh`. The response status is `200 OK` with a response time of `3.22 s` and a size of `863 B`. The response body is in JSON format, showing a successful status and data for the Sinnoh region.

Headers (6)

Key	Value	Description

Body

```
{
  "status": "success",
  "statusCode": 200,
  "message": "Data successfully retrieved",
  "data": {
    "areas": [],
    "game_indices": [
      {
        "game_index": 2006,

```