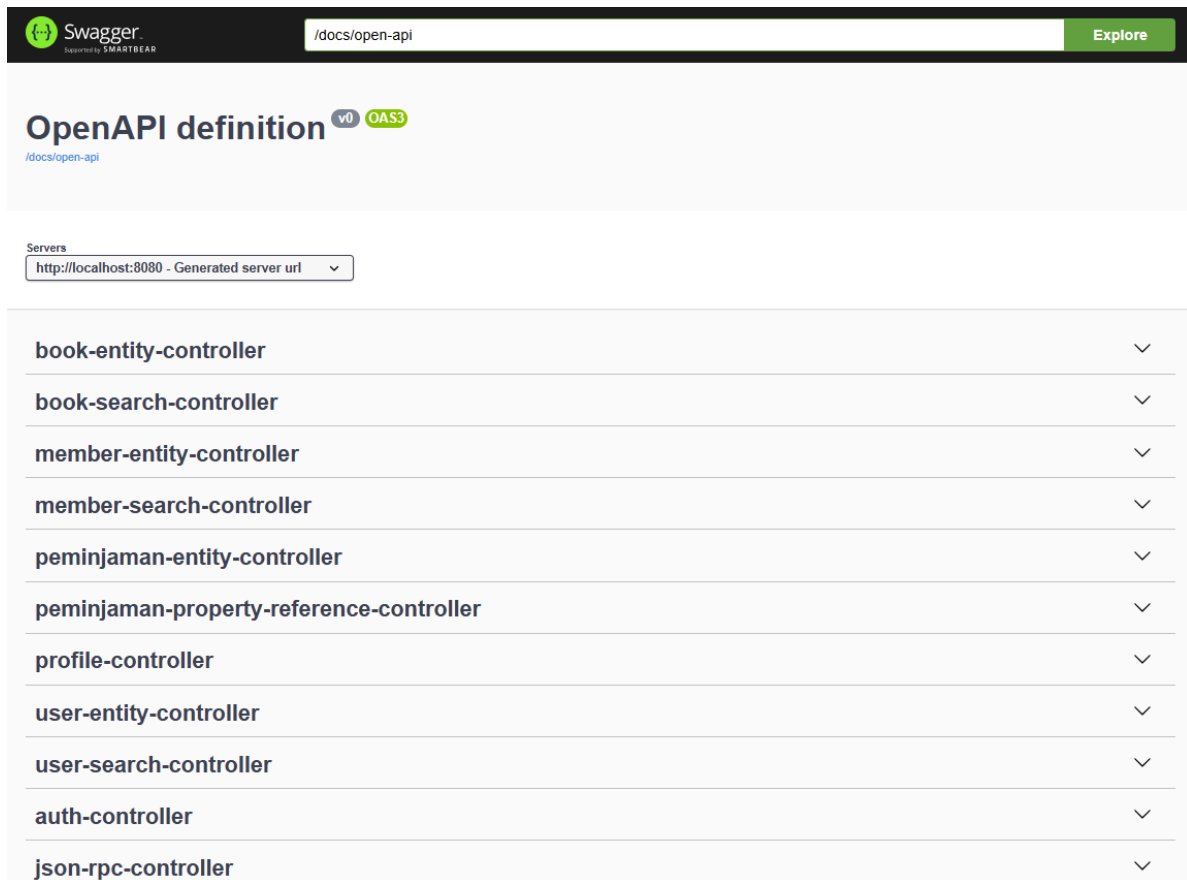


LAPORAN PEKERJAAN

Halaman utama pada swagger:



1. Authentication (Endpoint untuk registrasi dan login)

a. POST /register

- Deskripsi: Menerima data pengguna (nama, email, password), kemudian menyimpan ke database setelah mengenkripsi password
- Request Body :
 - o Name
 - o Email
 - o Password
- Responses
 - o 200 OK: Registrasi berhasil, mengembalikan data pengguna yang telah dibuat

- 400 Bad Request: Terjadi jika data yang dikirim tidak valid.

b. POST/login

- Deskripsi: Memvalidasi kredensial pengguna (email dan password). Jika valid, sistem akan memberikan accessToken yang harus digunakan pada header Authorization untuk mengakses endpoint yang terproteksi.
- Request Body :
 - Email
 - Password

Request body required application/json

Example Value | Schema

```
{
  "id": 0,
  "name": "string",
  "email": "string",
  "password": "string",
  "enabled": true,
  "authorities": [
    {
      "authority": "string"
    }
  ],
  "accountNonLocked": true,
  "username": "string",
  "accountNonExpired": true,
  "credentialsNonExpired": true
}
```

- Responses
 - 200 OK: Login berhasil, mengembalikan email dan accessToken.
 - 401 Unauthorized: Kredensial yang diberikan salah (email atau password tidak cocok).

Responses

Code	Description	Links
200	OK	No links

Media type
application/hal+json

Controls Accept header.

Example Value | Schema

```
{}
```

2. Books (Endpoint untuk manajemen buku)

a. GET/books

- Deskripsi: Menampilkan seluruh koleksi buku yang ada di perpustakaan..
- Parameters :
 - o Page: Menampilkan jumlah halaman dalam suatu buku
 - o Size: Menampilkan data yang ditampilkan per halaman (default: 20)
 - o Sort: Kriteria pengurutan dalam format properti, ascending, atau descending. Mendukung beberapa kriteria pengurutan.

Parameters	
Name	Description
page integer (query)	Zero-based page index (0..N) Default value : 0
<input type="text" value="0"/>	
size integer (query)	The size of the page to be returned Default value : 20
<input type="text" value="20"/>	
sort array[string] (query)	Sorting criteria in the format: property,(asc desc). Default sort order is ascending. Multiple sort criteria are supported.

- Responses
 - o 200 OK: Berhasil menampilkan seluruh koleksi buku.

Code	Description	Links
200	OK	No links

Media type
application/hal+json

Controls Accept header

Example Value | Schema

```
{
  "_embedded": {
    "books": [
      {
        "title": "string",
        "author": "string",
        "description": "string",
        "links": [
          {
            "additionalProp1": {
              "href": "string",
              "hreflang": "string",
              "title": "string",
              "type": "string",
              "deprecation": "string",
              "profile": "string",
              "name": "string",
              "templated": true
            },
            "additionalProp2": {
              "href": "string",
              "hreflang": "string",
              "title": "string",
              "type": "string",
              "deprecation": "string",
              "profile": "string",
              "name": "string",
              "templated": true
            }
          }
        ]
      }
    ]
  }
}
```

b. POST/books

- Deskripsi: Menambahkan buku kedalam database perpustakaan dengan atribut yang telah ditentukan, yaitu title, author, dan description
- Request Body :
 - o title
 - o author
 - o description

Request body required application/json

Example Value | Schema

```
{
  "id": 0,
  "title": "string",
  "author": "string",
  "description": "string"
}
```

- Request URL: <http://localhost:8080/books>
- Responses
 - o 403 Forbidden: Tidak bisa melakukan penambahan buku karena belum diberi akses (Belum login)

403
Undocumented Error: response status is 403

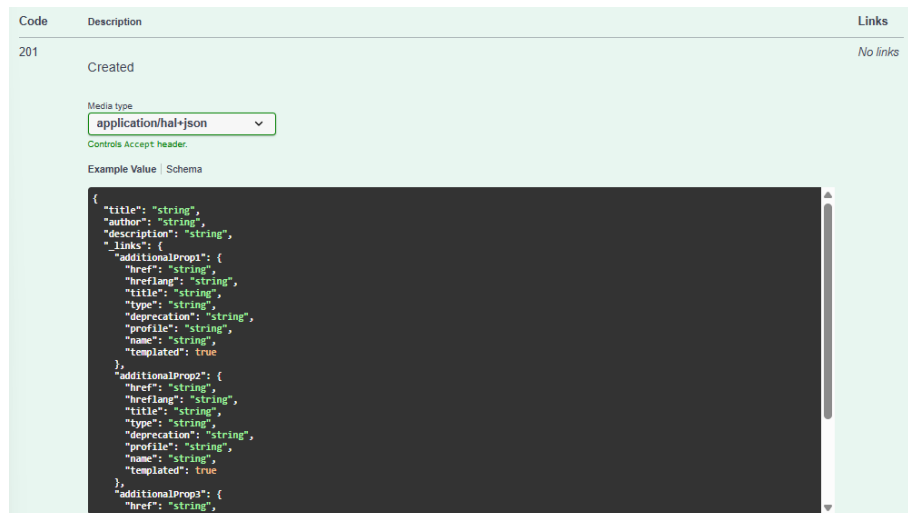
Response body

```
{
  "timestamp": "2025-10-13T10:40:03.666+00:00",
  "status": 403,
  "error": "Forbidden",
  "path": "/books"
}
```

Response headers

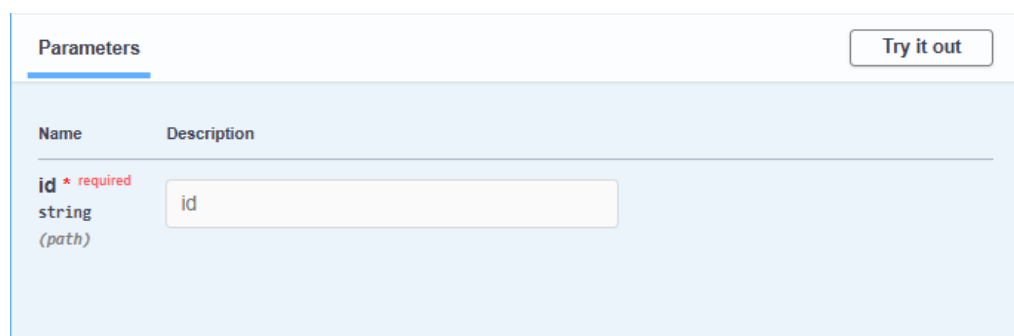
```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-type: application/hal+json
date: Mon, 13 Oct 2025 10:40:03 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 0
```

- o 201 Created: Buku telah berhasil dibuat atau ditambahkan ke database.



c. GET/books/{id}

- Deskripsi: Mengambil detail satu buku berdasarkan ID.
- Parameter: id



- Responses
 - o 200 OK, mnegembalikan data buku dengan id yang dijadikan sebagai parameter
 - o 404 Not Found, tidak dapat menemukan buku dengan id yang dijadikan parameter

Responses

Code	Description	Links
200	OK	No links

Media type

Controls Accept header.

Example Value | Schema

```

{
  "title": "string",
  "author": "string",
  "description": "string",
  "_links": {
    "additionalProp1": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "hreflang": "string",
      "title": "string"
    }
  }
}

```

404 Not Found

No links

d. PUT/books/{id}

- Deskripsi: Memperbarui **seluruh** data buku. Semua field wajib diisi.
- Parameter: id

Parameters

Name	Description
id * required string (path)	id

- Request Body: Book (title, author, description)

Request body *required*

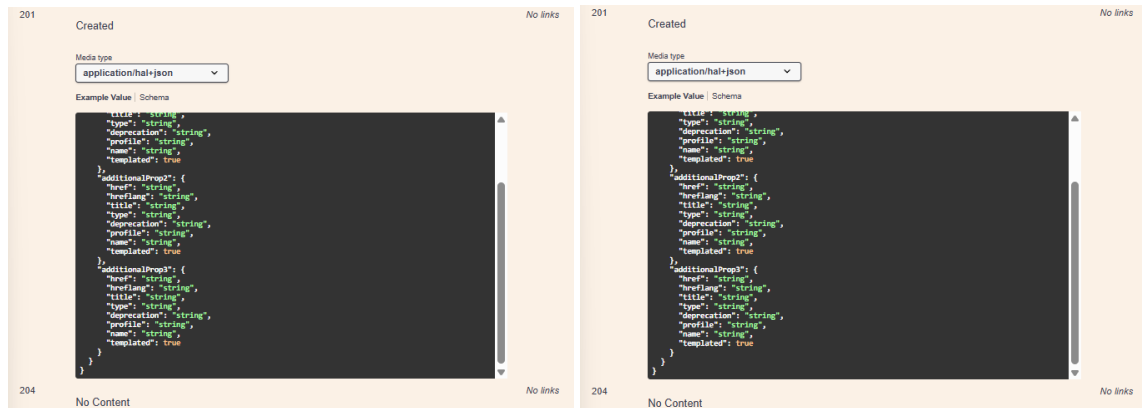
Example Value | Schema

```

{
  "id": 0,
  "title": "string",
  "author": "string",
  "description": "string"
}

```

- Responses
 - o 200 OK, mengembalikan data buku dengan id yang dijadikan parameter
 - o 201 Created, mengembalikan data buku dengan id yang dijadikan sebagai parameter
 - o 204, No content



e. PATCH/book/{id}

- Deskripsi: Memperbarui **sebagian** data buku. Hanya field yang dikirim yang akan diubah.
- Parameter: id

Parameters	
Name	Description
id * required string (path)	id

- Request Body: Book (hanya field yang ingin diubah)

Request body required	
<div> <div>Media type</div> <div>application/json</div> </div> <div> <div>Example Value Schema</div> <pre>{ "id": 0, "title": "string", "author": "string", "description": "string" }</pre> </div>	

- Responses
 - 200 OK, mengembalikan data buku dengan id yang dijadikan sebagai parameter
 - 204 No Content

Responses		
Code	Description	Links
200	OK	No links
Media type <input type="text" value="application/hal+json"/> <small>Controls Accept header.</small> Example Value Schema <div> <pre> { "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp2": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp3": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true } } </pre> </div>		
204	No Content	No links

f. DELETE/books/{id}

- Deskripsi: Menghapus data buku dari sistem.
- Parameter: id

Parameters	
Name	Description
id * required	
string (path)	id

- Responses
 - 204 No Content
 - 404 Not Found, tidak dapat menemukan buku dengan id yang dijadikan parameter

Responses		
Code	Description	Links
204	No Content	No links
404	Not Found	No links

g. GET/books/search/findByTitleContainingIgnoreCase

- Deskripsi: Mencari buku berdasarkan kata kunci tertentu.
- Parameter: keyword(string)

Parameters
Try it out

Name	Description
keyword string (query)	<input type="text" value="keyword"/>

- Responses
 - o 200OK, menampilkan buku yang dicari atau sesuai dengan keyword
 - o 404 Not Found, tidak dapat menemukan buku dengan keyword= yang dijadikan parameter

Responses		
Code	Description	Links
200	OK	No links
<div> Media type application/hal+json Controls Accept header. </div> <div> Example Value Schema </div> <pre> { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp2": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp3": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true } } </pre>		
404	Not Found	No links

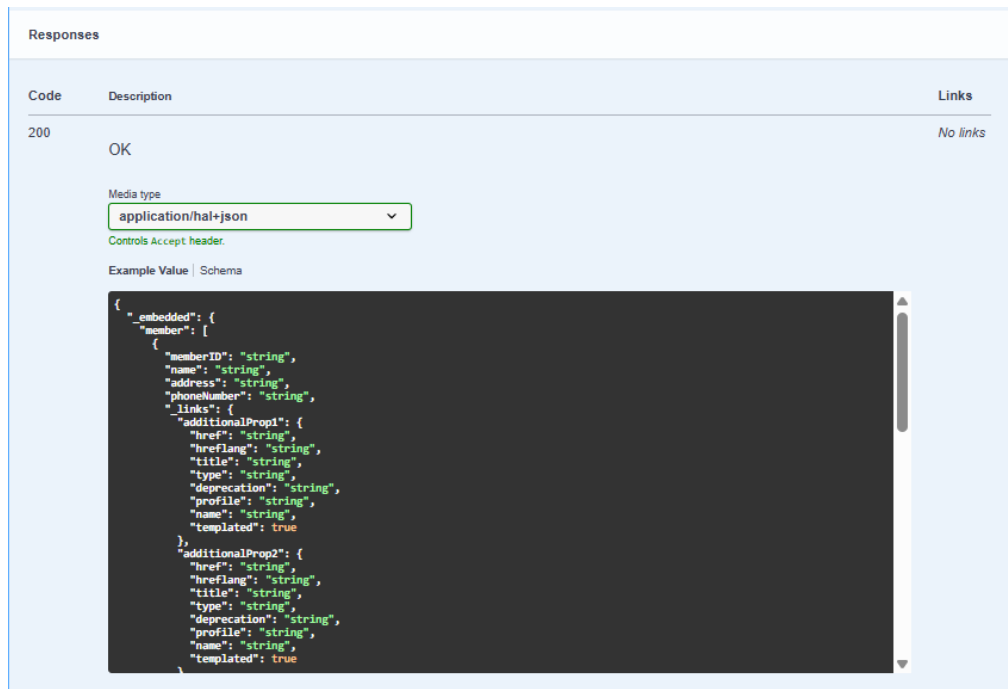
3. Member (Endpoint untuk mengelola data anggota)

a. GET /member

Fungsi: Mengambil daftar semua member yang terdaftar.

Metode HTTP: GET

Response Sukses: 200 OK, dengan *body* berisi *array* dari objek Member.



Responses

Code	Description	Links
200	OK	No links

Media type: application/hal+json

Controls: Accept header

Example Value | Schema

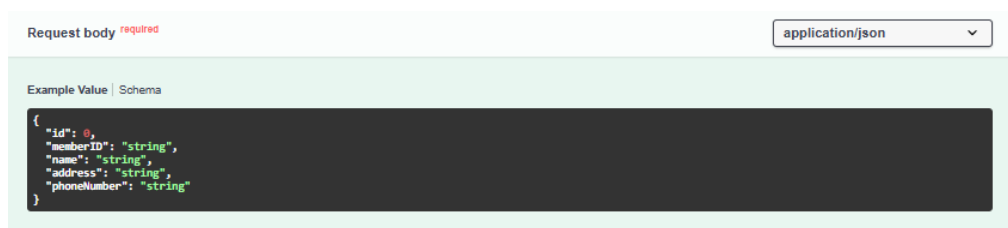
```
{
  "_embedded": {
    "member": [
      {
        "memberID": "string",
        "name": "string",
        "address": "string",
        "phoneNumber": "string",
        "_links": {
          "additionalProp1": {
            "href": "string",
            "hreflang": "string",
            "title": "string",
            "type": "string",
            "deprecation": "string",
            "profile": "string",
            "name": "string",
            "templated": true
          },
          "additionalProp2": {
            "href": "string",
            "hreflang": "string",
            "title": "string",
            "type": "string",
            "deprecation": "string",
            "profile": "string",
            "name": "string",
            "templated": true
          }
        }
      }
    ]
  }
}
```

b. POST /member

Fungsi: Menambahkan atau mendaftarkan seorang member baru ke dalam sistem.

Metode HTTP: POST

Request Body: Objek JSON yang merepresentasikan data member baru (misal: name, memberID, address).



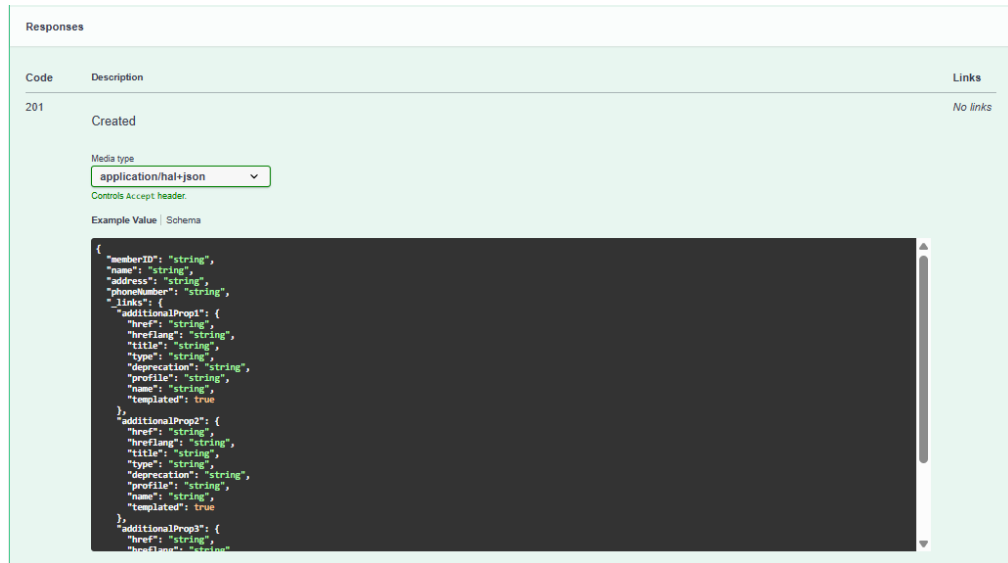
Request body required

application/json

Example Value | Schema

```
{
  "id": 0,
  "memberID": "string",
  "name": "string",
  "address": "string",
  "phoneNumber": "string"
}
```

Response Sukses: 201 Created, dengan *body* berisi data member yang baru saja dibuat, termasuk ID yang digenerate oleh sistem.

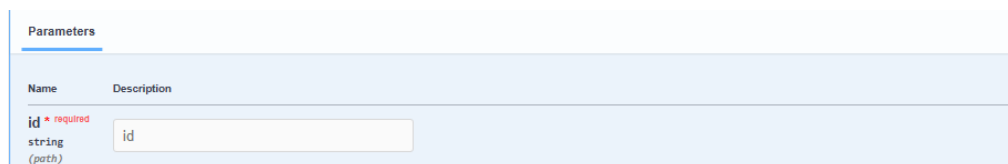


c. **GET /member/{id}**

Fungsi: Mengambil detail data seorang member spesifik berdasarkan ID uniknya.

Metode HTTP: GET

Parameter: id (Path variable) - ID dari member yang ingin dicari.



Response Sukses: 200 OK, dengan *body* berisi objek Member.

Response Gagal: 404 Not Found jika member dengan ID tersebut tidak ditemukan.

Responses		
Code	Description	Links
200	OK	No links
<div>Media type</div> <div>application/hal+json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <pre> { "memberID": "string", "name": "string", "address": "string", "phoneNumber": "string", "links": { "additionalProp1": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp2": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp3": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true } } } </pre>		
404	Not Found	No links

d. PUT /member/{id}

Fungsi: Memperbarui **seluruh** data member yang sudah ada. Metode ini mengharuskan klien mengirimkan semua atribut dari member, bahkan yang tidak berubah sekalipun.

Metode HTTP: PUT

Parameter: id (Path variable) - ID dari member yang akan diperbarui.

Parameters	
Name	Description
id * required string (path)	id

Request Body: Objek JSON dengan data member yang lengkap.

Request body required	application/json
Example Value Schema	
<pre> { "id": 0, "memberID": "string", "name": "string", "address": "string", "phoneNumber": "string" } </pre>	

Response Sukses: 200 OK.

Responses

Code	Description	Links
200	OK	No links

Media type
application/hal+json

Controls Accept header

Example Value | Schema

```

{
  "title": "string",
  "type": "string",
  "deprecation": "string",
  "profile": "string",
  "name": "string",
  "templated": true
},
"additionalProp1": {
  "href": "string",
  "hreflang": "string",
  "title": "string",
  "type": "string",
  "deprecation": "string",
  "profile": "string",
  "name": "string",
  "templated": true
},
"additionalProp2": {
  "href": "string",
  "hreflang": "string",
  "title": "string",
  "type": "string",
  "deprecation": "string",
  "profile": "string",
  "name": "string",
  "templated": true
},
"additionalProp3": {
  "href": "string",
  "hreflang": "string",
  "title": "string",
  "type": "string",
  "deprecation": "string",
  "profile": "string",
  "name": "string",
  "templated": true
},
"additionalProp4": {
  "href": "string",
  "hreflang": "string",
  "title": "string",
  "type": "string",
  "deprecation": "string",
  "profile": "string",
  "name": "string",
  "templated": true
}
}

```

201 Created
No links

Media type
application/hal+json

Example Value | Schema

```

{
  "memberID": "string",
  "name": "string",
  "address": "string",
  "phoneNumber": "string",
  "_links": {
    "additionalProp1": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp4": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    }
  }
}

```

204 No Content
No links

e. DELETE /member/{id}

Fungsi: Menghapus data member dari sistem.

Metode HTTP: DELETE

Parameter: id (Path variable) - ID dari member yang akan dihapus.

Parameters

Name	Description
id * required string (path)	id

Response Sukses: 204 No Content, menandakan data berhasil dihapus tanpa ada konten yang perlu dikembalikan.

Responses		
Code	Description	Links
204	No Content	No links
404	Not Found	No links

f. PATCH /member/{id}

Fungsi: Memperbarui **sebagian** data member. Klien hanya perlu mengirimkan atribut yang ingin diubah.

Metode HTTP: PATCH

Parameter: id (Path variable) - ID dari member yang akan diperbarui.

Parameters Try it out

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

Request Body: Objek JSON yang berisi hanya field yang ingin diubah.

Request body required application/json

Example Value | Schema

```
{
  "id": 0,
  "memberID": "string",
  "name": "string",
  "address": "string",
  "phoneNumber": "string"
}
```

Response Sukses: 200 OK.

Responses

Code	Description	Links
200	OK	No links

Media type
application/hal+json

Controls Accept header.

Example Value | Schema

```
{
  "memberID": "string",
  "name": "string",
  "address": "string",
  "phoneNumber": "string",
  "_links": {
    "additionalProp1": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    }
  }
}
```

204
No Content

No links

g. GET /member/search/findByName dan GET /member/search/findByMemberID

Fungsi: Melakukan pencarian member berdasarkan kriteria spesifik, yaitu nama atau ID Member.

Metode HTTP: GET

Parameter: name atau memberID (Request parameter).

Parameters	
Name	Description
member_id string (query)	<input type="text" value="member_id"/>

Parameters	
Name	Description
name string (query)	<input type="text" value="name"/>

Response Sukses: 200 OK, dengan *body* berisi *array* dari Member yang cocok dengan kriteria pencarian atau 404 Not Found

Responses		
Code	Description	Links
200	OK	No links
<div>Media type: <input type="text" value="application/hal+json"/></div> <div>Controls Accept header:</div> <div>Example Value Schema</div> <pre> { "embedded": { "member": [{ "memberID": "string", "name": "string", "address": "string", "phoneNumber": "string", "_links": { "additionalProp1": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "specification": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp2": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "specification": "string", "profile": "string", "name": "string", "templated": true } } }] } } </pre>		
404	Not Found	No links

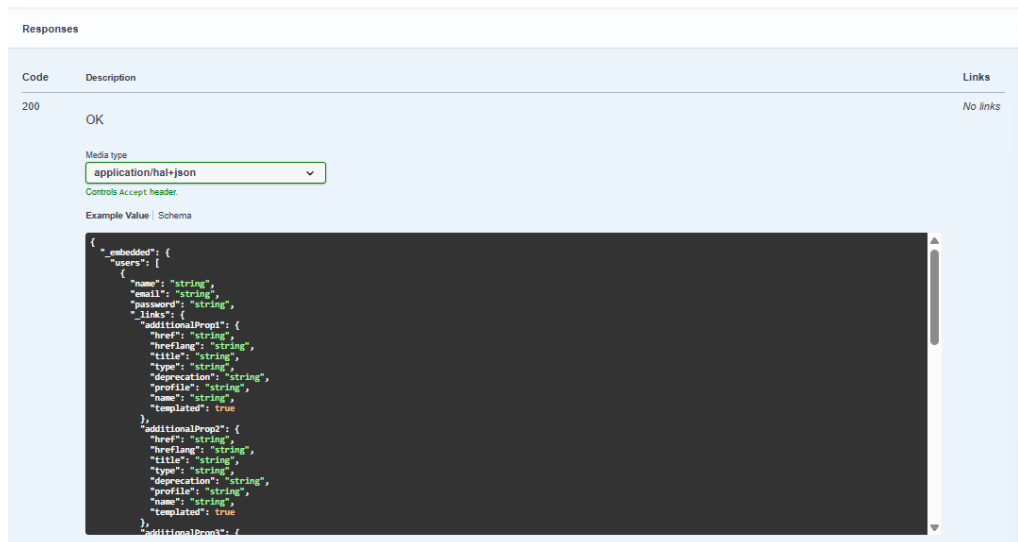
4. User (Endpoint untuk mengelola data user)

a. GET /User

Fungsi: Mengambil daftar semua User yang terdaftar.

Metode HTTP: GET

Response Sukses: 200 OK, dengan *body* berisi *array* dari objek User.

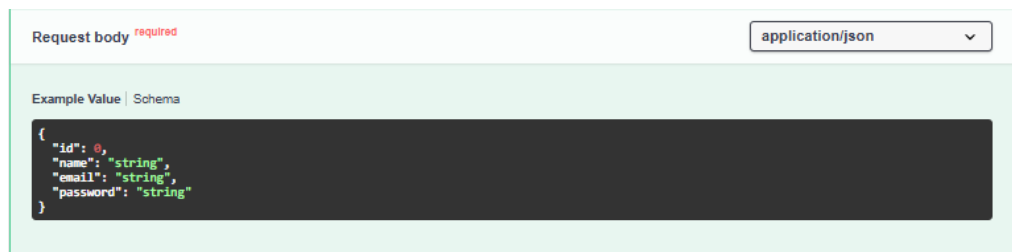


b. POST /User

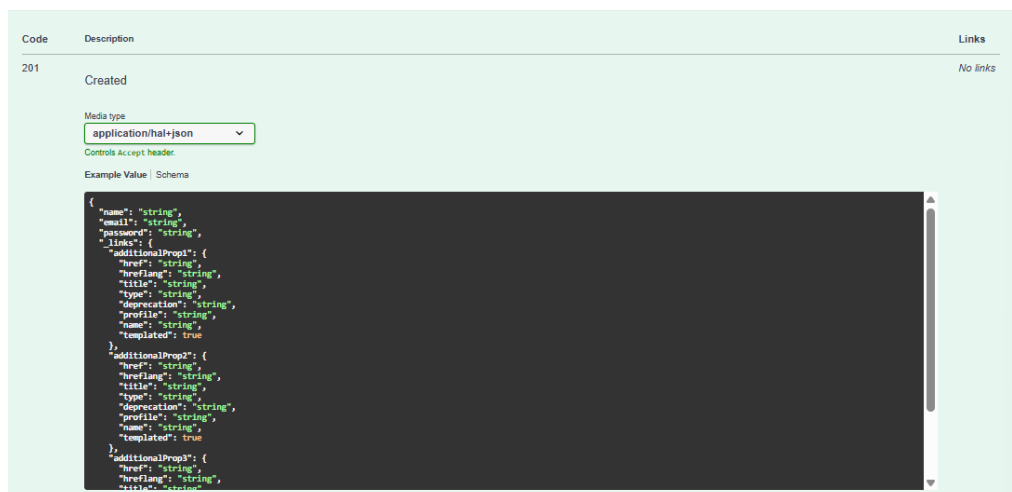
Fungsi: Menambahkan atau mendaftarkan seorang User baru ke dalam sistem.

Metode HTTP: POST

Request Body: Objek JSON yang merepresentasikan data User baru (misal: name, UserID, address).



Response Sukses: 201 Created, dengan *body* berisi data User yang baru saja dibuat, termasuk ID yang digenerate oleh sistem.



c. GET /User/{id}

Fungsi: Mengambil detail data seorang User spesifik berdasarkan ID uniknya.

Metode HTTP: GET

Parameter: id (Path variable) - ID dari User yang ingin dicari.

Parameters	
Name	Description
id * required string (path)	id

Response Sukses: 200 OK, dengan *body* berisi objek User.

Response Gagal: 404 Not Found jika User dengan ID tersebut tidak ditemukan.

Responses		
Code	Description	Links
200	OK	No links
<div>Media type application/hal+json</div> <div>Example Value Schema</div> <pre>{ "name": "string", "email": "string", "password": "string", "links": { "additionalProp1": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp2": { "href": "string", "hreflang": "string", "title": "string", "type": "string", "deprecation": "string", "profile": "string", "name": "string", "templated": true }, "additionalProp3": { "href": "string", "hreflang": "string", "title": "string" } } }</pre>		
404	Not Found	No links

d. PUT /User/{id}

Fungsi: Memperbarui seluruh data User yang sudah ada. Metode ini mengharuskan klien mengirimkan semua atribut dari User, bahkan yang tidak berubah sekalipun.

Metode HTTP: PUT

Parameter: id (Path variable) - ID dari User yang akan diperbarui.

Name	Description
id * required string (path)	id

Request Body: Objek JSON dengan data User yang lengkap.

Request body required application/json

Example Value | Schema

```
{
  "id": 0,
  "name": "string",
  "email": "string",
  "password": "string"
}
```

Response Sukses: 200 OK, 201 Createad, 204 No Content

Responses

Code	Description	Links
200	OK	No links

Media type
application/hal+json

Controls Accept header

Example Value | Schema

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "links": {
    "additionalProp1": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    }
  }
}
```

201 Created

No links

Media type
application/hal+json

Example Value | Schema

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "links": {
    "additionalProp1": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    }
  }
}
```

204 No Content

No links

e. **DELETE /User/{id}**

Fungsi: Menghapus data User dari sistem.

Metode HTTP: DELETE

Parameter: id (Path variable) - ID dari User yang akan dihapus.

Name	Description
id * required string (path)	id

Response Sukses: 204 No Content, menandakan data berhasil dihapus tanpa ada konten yang perlu dikembalikan.

Responses		
Code	Description	Links
204	No Content	No links
404	Not Found	No links

f. PATCH /User/{id}

Fungsi: Memperbarui sebagian data User. Klien hanya perlu mengirimkan atribut yang ingin diubah.

Metode HTTP: PATCH

Parameter: id (Path variable) - ID dari User yang akan diperbarui.

Parameters

Name	Description
id * required string (path)	id

Request Body: Objek JSON yang berisi hanya field yang ingin diubah.

Request body required application/json

Example Value | Schema

```
{
  "id": 0,
  "name": "string",
  "email": "string",
  "password": "string"
}
```

Response Sukses: 200 OK, 204 No Content

Responses

Code	Description	Links
200	OK	No links

Media type
application/hal+json

Controls Accept header

Example Value | Schema

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "links": {
    "additionalProp1": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "deprecation": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "hreflang": "string",
      "title": "string"
    }
  }
}
```

204	No Content	No links
-----	------------	----------

g. GET /User/search/findByEmail

Fungsi: Melakukan pencarian User berdasarkan email USER

Metode HTTP: GET

Parameter: email (Request parameter).

Parameters

Name	Description
email string (query)	<input type="text" value="email"/>

Response Sukses: 200 OK, dengan *body* berisi *array* dari User yang cocok dengan kriteria pencarian.

Responses

Code	Description	Links
200	OK	No links

Media type
application/hal+json

Controls Accept header:
Example Value | Schema

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "links": {
    "additionalProp1": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "description": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp2": {
      "href": "string",
      "hreflang": "string",
      "title": "string",
      "type": "string",
      "description": "string",
      "profile": "string",
      "name": "string",
      "templated": true
    },
    "additionalProp3": {
      "href": "string",
      "hreflang": "string",
      "title": "string",

```

404 Not Found

No links

5. JSON-RPC (Endpoint untuk JSON-RPC untuk operasi buku)

a. POST/jsonrpc

Fungsi: Sebagai gateway untuk semua operasi yang berkaitan dengan manajemen buku.

Metode HTTP: POST

Request Body:

- Jsonrpc
- Method

- Params
- Id

Request body required application/json

Example Value | Schema

```
{  "jsonrpc": "string",  "method": "string",  "params": {},  "id": "string"}
```

Response Sukses: 200 OK. Body dari respons akan mengikuti format JSON-RPC, berisi result dari operasi yang dijalankan dan id yang sama dengan permintaan.

Response Gagal: 400 Bad Request, jika method tidak ditemukan atau jika ada kesalahan dalam memproses permintaan.

Responses

Code	Description	Links
200	OK	No links

Media type application/json

Controls Accept header.

Example Value | Schema

```
{}
```