

```
"INACTIVE",
"RESTRICTED",
"VERIFIED",
"UNVERIFIED"
]
}
}
}
}
}
```

crate the api documentation using this json

Here is a clean, structured **API Documentation** generated from your OpenAPI JSON.

EOD & Desktop User Management API

Version: v0

Base URL: <http://localhost:8080>

◆ EOD (End Of Day) APIs

1 Get All EOD Records

GET /api/eod

✓ **Response: 200 OK**

Returns a list of EOD records.

json

 Copy code

```
[
  {
    "desktopUserId": 1,
    "monthId": 2,
```

```
    "timeIn": { "hour": 9, "minute": 0, "second": 0, "nano": 0 },
    "timeOut": { "hour": 18, "minute": 0, "second": 0, "nano": 0 },
    "date": "2026-02-11",
    "attendanceStatus": "FULL_DAY",
    "todaysWork": "Completed API development",
    "assignedBy": "Manager",
    "remarks": "On track"
  }
}
```

2 Get EOD Record by ID

GET /api/eod/{id}

◆ Path Parameter

Name	Type	Required
id	integer (int64)	Yes

✔ Response: 200 OK

Returns a single EODInOutDto object.

3 Create EOD Record

POST /api/eod

📄 Request Body (application/json)

```
json
{
  "desktopUserId": 1,
  "monthId": 2,
  "timeIn": { "hour": 9, "minute": 0, "second": 0, "nano": 0 },
  "timeOut": { "hour": 18, "minute": 0, "second": 0, "nano": 0 },
  "date": "2026-02-11",
  "attendanceStatus": "FULL_DAY",
  "todaysWork": "Worked on backend APIs",
  "assignedBy": "Team Lead",
}
```

📄 Copy code

```
"remarks": "Completed successfully"
}
```

✓ Response: 200 OK

Returns created EODInOutDto .

4 Update EOD Record (Full Update)

PUT /api/eod/{id}

◆ Path Parameter

Name	Type	Required
id	integer (int64)	Yes

📄 Request Body

Same structure as EODInOutDto .

✓ Response: 200 OK

Returns updated record.

5 Patch EOD Record (Partial Update)

PATCH /api/eod/{id}

◆ Path Parameter

Name	Type	Required
id	integer (int64)	Yes

📄 Request Body

Partial EODInOutDto object.

✓ Response: 200 OK

Returns updated record.

6 Delete EOD Record

DELETE /api/eod/{id}

◆ Path Parameter

Name	Type	Required
id	integer (int64)	Yes

✓ Response: 200 OK

Returns confirmation message (string).

7 Get EOD Records by User

GET /api/eod/user/{userId}

◆ Path Parameter

Name	Type	Required
userId	integer (int64)	Yes

✓ Response: 200 OK

Returns list of EODInOutDto .

8 Get EOD Records by Date

GET /api/eod/date?date=YYYY-MM-DD

◆ Query Parameter

Name	Type	Format	Required
date	string	date (YYYY-MM-DD)	Yes

✓ Response: 200 OK

Returns list of EODInOutDto .

◆ Desktop User APIs

1 Get All Desktop Users

GET /api/desktop-user

✓ Response: 200 OK

Returns list of DesktopUserDto .

2 Get Desktop User by ID

GET /api/desktop-user/{id}

◆ Path Parameter

Name	Type	Required
id	integer (int64)	Yes

✓ Response: 200 OK

Returns DesktopUserDto .

3 Create Desktop User

POST /api/desktop-user

📎 Request Body

json

📋 Copy code

```
{
  "desktopId": "D-101",
  "desktopUserName": "John Doe",
  "desktopIp": "192.168.1.10",
  "currentActivity": "ACTIVE",
  "activityStatus": "VERIFIED"
}
```

✓ Response: 200 OK

Returns created `DesktopUserDto` .

Update Desktop User

PUT `/api/desktop-user/{id}`

◆ Path Parameter

Name	Type	Required
id	integer (int64)	Yes

Request Body

Same as `DesktopUserDto` .

✓ Response: 200 OK

Returns updated `DesktopUserDto` .

Delete Desktop User

DELETE `/api/desktop-user/{id}`

◆ Path Parameter

Name	Type	Required
id	integer (int64)	Yes

✓ Response: 200 OK

No content body.

Data Models

EODInOutDto

Field	Type	Description
desktopUserId	integer (int64)	Associated desktop user

Field	Type	Description
monthId	integer (int64)	Month reference
timeIn	LocalTime	Check-in time
timeOut	LocalTime	Check-out time
date	string (YYYY-MM-DD)	Work date
attendanceStatus	enum	FULL_DAY, HALF_DAY, LOP, ON_LEAVE
todaysWork	string	Work description
assignedBy	string	Assigned by
remarks	string	Additional notes



LocalTime

Field	Type
hour	integer (int32)
minute	integer (int32)
second	integer (int32)
nano	integer (int32)



DesktopUserDto

Field	Type	Enum Values
desktopId	string	—
desktopUserName	string	—
desktopIp	string	—

Field	Type	Enum Values
currentActivity	string	ACTIVE, INACTIVE, RESTRICTED, VERIFIED, UNVERIFIED
activityStatus	string	ACTIVE, INACTIVE, RESTRICTED, VERIFIED, UNVERIFIED






Attendance Status Enum

- FULL_DAY
- HALF_DAY
- LOP
- ON_LEAVE

Activity Status Enum

- ACTIVE
- INACTIVE
- RESTRICTED
- VERIFIED
- UNVERIFIED

If you want, I can also generate:

-  Swagger-ready Markdown file
-  Confluence documentation format
-  Postman collection
-  Sample cURL requests
-  PDF-ready professional API documentation

Just tell me the format you need.

