API Documentation -: Reqres Demo (Postman)

1. DEL DELETE User

- **Request Type:** DELETE
- **Endpoint:** /api/users/2
- **Description:** Deletes the user with ID 2. No response body is returned.
- **Response:**
    - **Status Code:** 204 No Content
    - **Body:** *(No Content)*

2. GET Delayed Response

Request Type: GET
Endpoint: /api/users?delay=3
Description: Fetches the list of users with a delayed response of 3 seconds. Useful for testing loading states.
Parameters:

- delay (integer) – Time in seconds to delay the response.

Response:

- **Success (200 OK)**

3. GET List Users

Request Type: GET
Endpoint: /api/users
Description: Fetches a paginated list of users.
Parameters:

- page (integer) – Specifies the page number of users to retrieve.

Response:

- **Success (200 OK)**

json
CopyEdit

```
{
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://reqres.in/img/faces/7-image.jpg"
    },
    ...
  ]
}
```

4. GET List

Request Type: GET
Endpoint: /api/unknown
Description: Fetches a list of resources (e.g., colors or other data types).
Parameters: None

Response:

- **Success (200 OK)**

json
CopyEdit

```
{
  "page": 1,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 1,
      "name": "cerulean",
      "year": 2000,
      "color": "#98B2D1",
      "pantone_value": "15-4020"
    },
    ...
  ]
}
```

5. GET Single List Not Found

Request Type: GET
Endpoint: /api/unknown/23
Description: Tries to fetch a resource that does not exist.

Response:

- **Error (404 Not Found)**

  ```
  json
  CopyEdit
  {
    "error": "Resource not found"
  }
  ```

6. GET Single List

Request Type: GET
Endpoint: /api/unknown/2
Description: Fetches a single resource based on its ID.
Parameters:

- {id} (integer) – The ID of the resource to fetch.

Response:

- **Success (200 OK)**

  ```
  json
  CopyEdit
  {
    "data": {
      "id": 2,
      "name": "fuchsia rose",
      "year": 2001,
      "color": "#C74375",
      "pantone_value": "17-2031"
    }
  }
  ```

- **Error (404 Not Found)**

  json
  CopyEdit

  ```
  {
    "error": "Resource not found"
  }
  ```

7. GET Single List Not Found

- **Request Type:** GET
- **Endpoint:** /api/unknown/23
- **Description:** Tries to fetch resource with ID 23, but it does not exist.
- **Response:**
  - **Status Code:** 404 Not Found
  - **Body:** {} (empty JSON object)

8. GET Single User

- **Request Type:** GET
- **Endpoint:** /api/users/2
- **Description:** Retrieves details of a user with ID 2.
- **Response:**
  - **Status Code:** 200 OK
  - **Body:**

    json
    CopyEdit

    ```
    {
      "data": {
        "id": 2,
        "email": "janet.weaver@reqres.in",
        "first_name": "Janet",
        "last_name": "Weaver",
        "avatar": "https://reqres.in/img/faces/2-image.jpg"
      },
      "support": {
        "url":
    "https://contentcaddy.io?utm_source=reqres&utm_medium=json&utm_campaign=referral",
        "text": "Tired of writing endless social media content? Let Content Caddy generate it for you."
      }
    }
    ```

9. PATCH Partial Update User

- **Request Type:** PATCH
- **Endpoint:** /api/users/2
- **Description:** Partially updates the user with ID 2.
- **Request Body:**

  ```
  json
  CopyEdit
  {
    "name": "morpheus",
    "job": "zion resident"
  }
  ```

- **Response:**
  - **Status Code:** 200 OK
  - **Body:**

    ```
    json
    CopyEdit
    {
      "name": "morpheus",
      "job": "zion resident",
      "updatedAt": "2024-01-30T10:15:30.123Z"
    }
    ```

10. POST Create User

Request Type: POST
Endpoint: /api/users
Description: Creates a new user with the provided details. This is used for adding new users to the system.
Parameters:
• name (string) – The name of the user.
• job (string) – The job title of the user.

Response:

```
json
CopyEdit
{
```

```
  "name": "John Doe",
  "job": "Software Engineer",
  "id": "123",
  "createdAt": "2024-01-30T12:34:56.789Z"
}
```

11. POST Login - Successful

Request Type: POST
Endpoint: /api/login
Description: Logs in a user with valid credentials. Returns an authentication token upon success.
Parameters:
• email (string) – User’s email address.
• password (string) – User’s password.

Response:

json
CopyEdit

```
{
  "token": "QpwL5tke4Pnpja7X4"
}
```

12. POST Login - Unsuccessful

Request Type: POST
Endpoint: /api/login
Description: Attempts to log in a user with incorrect or missing credentials. Returns an error message indicating the issue.
Parameters:
• email (string) – User’s email address.

Response:

json
CopyEdit

```
{
  "error": "Missing password"
}
```

13. POST Register - Successful

Request Type: POST
Endpoint: /api/register
Description: Registers a new user with valid credentials. Returns a success response with a unique user ID.
Parameters:
• email (string) – User’s email address.
• password (string) – User’s password.

Response:

json
CopyEdit

```
{
  "id": 4,
  "token": "QpwL5tke4Pnpja7X4"
}
```

14. POST Register - Unsuccessful

Request Type: POST
Endpoint: /api/register
Description: Attempts to register a user without providing required details (e.g., missing password). Returns an error message.
Parameters:
• email (string) – User’s email address.

Response:

json
CopyEdit

```
{
  "error": "Missing password"
}
```

15. PUT Full Update User

Request Type: PUT
Endpoint: /api/users/{id}
Description: Updates the details of an existing user by replacing all fields with the new data.
Parameters:

- name (string) – Updated name of the user.
- job (string) – Updated job title of the user.

Response:

json
CopyEdit

```
{
  "name": "Jane Doe",
  "job": "Product Manager",
  "updatedAt": "2024-01-30T12:34:56.789Z"
}
```