# Api Design Document

- Api Design : Reference Data Maintenance
- Technologies Used :
  - Backend - Nodejs, expressjs
  - Database- mongoDb
- Purpose of the api :
  - To maintain the userdata
  - To perform the create, read, edit operations of the users data
- Possibility of failure scenarios and conditions :
  - Incorrect database configuration
- Postman documentation link :

https://documenter.getpostman.com/view/23556069/2s83zpK1be

# HTTP methods and status codes

## ➢ 1 ) GET /api/fields

- **Request :** List all the fields contained within the collection resource

```
GET /api/fields    HTTP/1.1

Host : http://localhost:8000/api/fields

Accept: application/json
```

- **Response :** A response of “200 OK” indicating that the request has succeeded.

```
HTTP/1.1 200 OK

Vary: Accept

Content-Type: application/json
```


```
"success": true,
    "statusCode": 200,
    "data": [
        {
            "_id": "63483009fdea3c69d164ddb7",
            "userName": "steve",
            "fieldName": "I am coming to chennai",
            "createdAt": "2022-10-13T15:34:33.456Z",
            "updatedAt": "2022-10-13T15:34:33.456Z",
            "__v": 0
        },
        {
            "_id": "6348c89109a944e8e5b05b07",
            "userName": "Binny",
            "fieldName": "please come to my cabin",
            "createdAt": "2022-10-14T02:25:21.331Z",
            "updatedAt": "2022-10-14T02:25:21.331Z",
            "__v": 0
        }
    ],
    "message": "Successfully fetched data from the database"
}
```

- Success response :
  - Status Code : 200 OK – no error
- Error response :
  - Status Code : 404 Not Found – the “fieds” resource does not exist

## ➢ 2 ) POST /api/create

- **Request :** To create a document the collections with required field

  **.Validations :** The validation has been done by the mongoose schema validation

```
{
    userName: {
      type: String,
      required: true,
    },
    fieldName: {
      type: String,
      required: true,
      maxLength: 300, //only 300 characters are accepted
    },
  },
  {
    timestamps: true,
  },
```

**Request object**

HTTP/1.1 201 Created

Vary: Accept

Content-Type: application/json

```
{
      "userName    "Binny",
      "fieldName": "please come to my cabin"
}
```

- A response of “201 Created” indicates that the request has been fulfilled.
- The URI of the new resource is provided in the response

- **Response**

HTTP/1.1 201 Created

Vary: Accept

Content-Type: application/json


```
   {
 "success": true,
    "statusCode": 201,
    "message": "Successfully Added Data",
    "data": {
        "userName": "Binny",
        "fieldName": "please come to my cabin",
        "_id": "6348c89109a944e8e5b05b07",
        "createdAt": "2022-10-14T02:25:21.331Z",
        "updatedAt": "2022-10-14T02:25:21.331Z",
        "__v": 0
    }
```


- Success response :
  - Status Code : 201 OK – no error
- Error response :
  - Status Code : 512 -Validation error

## ➢ 3 ) PUT /api/edit/:id

- **Query params –** http://localhost:8000/api/edit/6348d553d529e9c7f94e2155
- **Validations -** mongoose schema validation
- **Request :** Update the particular field in the collection based on the id that are getting from params as id

- **Request object :**

PUT /api/edit/:id

HTTP/1.1

Host: http://localhost:8000/api/edit/:1223233

Accept: application/json

```
{
    "userName":"will world",
    "fieldName" : "This is updated field"
}
```

- Response
    - A response of “204 updated” indicates that the request has been fulfilled

HTTP/1.1 200 OK

Vary: Accept

Content-Type: text/javascript

```
  {
         "username”  : "Will world",
        "fieldName": " This is updated field”
        "_id": "6348c89109a944e8e5b05b07",
        "createdAt": "2022-10-14T02:25:21.331Z",
        "updatedAt": "2022-10-14T02:25:34.331Z",
        "__v": 0
}
```

- Success response :
    - Status Code : 204 OK – successfully updated
- Error response :
    - Status Code : 512 -Validation error

- Database : Mongodb

    - Collection name – epikindifi
    - Fields: userName(user input), fieldName (user input) , _id(default , createdAt(default), updatedAt(default) updatedAt(default)

- Purpose of Each keys

| S.no | Field Name | Type | Description | Purpose |
|---|---|---|---|---|
| 1 | _id | String | Unique id for individual document | To identify the document uniquely |
| 2 | userName | String | Name of the user | To idenitify the document and to perform search,update and etc opeations. |
| 3 | fieldName | String | Name of the field | To idenitify the document and to perform search,update and etc opeations. |
| 4 | createdAt | Date | document creation date and time | To get the document creation date and time |
| 5 | updatedAt | Date | document updation date and time | To get the document updation date and time |