# Host : 127.0.0.1:9090(172.16.2.154:9090)

172.16.2.154—deployment machine IP

# Registration

|  |  |
| --- | --- |
| **URL** | /register |
| **Method** | POST |
| **Params** | name, email, password |

Upon the successful registration the following json response will be issued.

|  |
| --- |
| {      "error": false,      "message": "You are successfully registered"  } |

If the request is missing mandatory parameters the following json will be issued.

|  |
| --- |
| {      "error": true,      "message": "Required field(s) email, password is missing or empty"  } |

# Login

|  |  |
| --- | --- |
| **URL** | /login |
| **Method** | POST |
| **Params** | email, password |

On successful login the following json will be issued.

|  |
| --- |
| {      "error": false,      "name": "DharmaKrish",      "email": "dharmakrish6@gmail.com",      "apiKey": "940bb12af8d7b040876f60f965c5be6d",      "createdAt": "2016-05-10 12:10:18"  } |

If the credentials are wrong, you can expect the following json.

|  |
| --- |
| {      "error": true,      "message": "Login failed. Incorrect credentials"  } |

# **Authenticate**

While dealing with task data, we need to identify the user using the API key in the request header by reading **Authorization** field. Basically we’ll look into database for matched API key and get the appropriate user. If the API key not present in users table, then we’ll stop the execution and echo the error json.

If the api key is missing in the request header, the following json will be echoed with 400 status code.

|  |
| --- |
| {      "error": true,      "message": "Api key is misssing"  } |

If the api key is not valid following json will echoed with 401 status code.

|  |
| --- |
| {      "error": true,      "message": "Access Denied. Invalid Api key"  } |

**Api Calls with Authentication (Including API key in the request)**  
Following are the API calls should have an Api Key in the request header. These api calls primarily deals the user’s task data like creating, reading, updating and deleting.

# Creating New Task

|  |  |
| --- | --- |
| **URL** | /tasks |
| **Method** | POST |
| **Params** | task |

On successful creation of new task following json will be issued. If you got this json, you can see new row inserted in **tasks** and **user\_tasks** tables.

|  |
| --- |
| {      "error": false,      "message": "Task created successfully",      "task\_id": 1  } |

# Getting All Tasks

|  |  |
| --- | --- |
| **URL** | /tasks |
| **Method** | GET |
| **Params** | – |

Following json will be issued for list of tasks. The **“tasks”** represents list of tasks as an array. Also if the **“status”** is**0**, that means the task is not done yet.

|  |
| --- |
| {      "error": false,      "tasks": [          {              "id": 1,              "task": "Complete REST article by Sunday",              "status": 0,              "createdAt": "2014-01-08 23:35:45"          },          {              "id": 2,              "task": "Book bus tickets!",              "status": 0,              "createdAt": "2014-01-08 23:56:52"          }      ]  } |

# Getting Single Task

|  |  |
| --- | --- |
| **URL** | /tasks/id (id should be replaced with task id) |
| **Method** | GET |
| **Params** | – |

The details of a single task will be in following json format.

|  |
| --- |
| {      "error": false,      "id": 2,      "task": "Book bus tickets!",      "status": 0,      "createdAt": "2014-01-08 23:56:52"  } |

If you pass a task id which is not there in the database, you will get **404** not found error.

# Updating Task

|  |  |
| --- | --- |
| **URL** | /tasks/id (id should be replaced with task id) |
| **Method** | PUT |
| **Params** | task, status (0 or 1) |

Upon successful updation you will get following json.

|  |
| --- |
| {      "error": false,      "message": "Task updated successfully"  } |

# Deleting Task

|  |  |
| --- | --- |
| **URL** | /tasks/id (id should be replaced with task id) |
| **Method** | DELETE |
| **Params** | – |

You will get following json if the task is deleted successfully.

|  |
| --- |
| {      "error": false,      "message": "Task deleted succesfully"  } |

# ALL API

|  |  |  |  |
| --- | --- | --- | --- |
| **URL** | **Method** | **Parameters** | **Description** |
| http://localhost/task\_manager/v1/register | POST | name, email, password | User registration |
| http://localhost/task\_manager/v1/login | POST | email, password | User login |
| http://localhost/task\_manager/v1/tasks | POST | task | To create new task |
| http://localhost/task\_manager/v1/tasks | GET |  | Fetching all tasks |
| http://localhost/task\_manager/v1/tasks/:id | GET |  | Fetching single task |
| http://localhost/task\_manager/v1/tasks/:id | PUT |  | Updating single task |
| http://localhost/task\_manager/v1/tasks/:id | DELETE | task, status | Deleting single task |