**ToDo Application**

**End points**

/task/{userId}

/task/{userId}/{taskID}

**Verbs**

GET - to fetch all the tasks under the user when he or she logs in

POST - to create a new task

DELETE - to delete a task

PUT - to modify a task

**Request**

Request message will have a json attached to it depending on type of request eg: in case of POST and PUT the new and modified task will be passed along the request.

on the other hand DELET and GET will not have any data along with it.

Sample json:

in case of adding new task and modifying the task.

{ “task” : {

“id” : “if present”,

“name” : “task name”,

“description” : “description here”,

“priority” : “high”,

“date” : “date of completion”

}

}

**Response**

**Successful response**

incase of POST,PUT,DELETE

{

“code” : 200,

“content” : {

“id” : 123456

}

}

incase of GET (at the time of sign in we will need all the tasks stored for a user)

{

"code" : 200,

"tasks" : [

{

"id" : "task id",

"name" : "task name",

"description" : "description here",

"priority" : "high",

"date" : "date of completion"

},

{

"id" : "task id",

"name" : "task name",

"description" : "description here",

"priority" : "high",

"date" : "date of completion"

},

{

"id" : "task id",

"name" : "task name",

"description" : "description here",

"priority" : "high",

"date" : "date of completion"

}

]

}

**unsuccessful response**

{

“code” : 401,

“content” : {

“error” : “unauthorized”

}

}

or

{

“code” : 500,

“content” : {

“error” : “internal error try again”

}

}

**Error handling**

In case of the error the Api can send back error code and description of error eg :

{

“code” : 401, //error code, other error codes could be 500, 403

“content” : {

“error” : “unauthorized” //description of the error

}

}