Method: getAll (GET)

Usage: gets all the tasks from the Task table. This will not be used anywhere in our application but I have created it as it is one of the essential CRUD calls.

URL: http://localhost:8080/task/getAll

Params: None

Response:

```
[
   {
       "id": int,
       "text": "string",
       "isActive": true,
       "isComplete": false
   }
]
```

Permission: None

Method: getAllActive (GET)

Usage: gets all the active tasks from the Task table. This is used in our application to populated the grid when “All” option is selected.

URL: http://localhost:8080/task/getAllActive

Params: None

Response:

```
[
  {
    "id": int,
    "text": "string",
    "isActive": true,
    "isComplete": false
  }
]
```

Permission: None

Method: getAllActive/{isComplete} (GET)

Usage: gets the respective active tasks from the Task table. This is used in our application to populated the grid when “Pending” or “Completed” options are selected.

URL: http://localhost:8080/task/getAllActive/{isComplete}

Path Params:

isComplete:

type: Boolean

description: This is the flag which tells the API to pick completed or incompleted tasks.

Response:

```
[
  {
    "id": int,
    "text": "string",
    "isActive": true,
    "isComplete": false
  }
]
```

Permission: None

Method: getById/{id} (GET)

Usage: gets the task from the Task table with the respective task id. This will not be used anywhere in our application but I have created it as it is one of the essential CRUD calls.

URL: http://localhost:8080/task/getById/{id}

Path Params:

    id:

        type: Long

        description: This is the id for which the API will return the task.

Response:

```
{
   "id": int,
   "text": "string",
   "isActive": true,
   "isComplete": false
}
```

Permission: None

Method: addTask (POST)

Usage: adds a new task to the Task table. This is used by our input box when enter is pressed after adding a task.

URL: http://localhost:8080/task/addTask

Request Body:

```
{
   "text": "string"
}
```

Response:

```
{
   "id": int,
   "text": "string",
   "isActive": true,
   "isComplete": false
}
```

Permission: None

Method: updateTask (PUT)

Usage: updates a task in the Task table. This is used by our grid when a task is marked complete/incomplete or when a task is “deleted” by pressing the red cross button.

URL: http://localhost:8080/task/updateTask

Request Body:

```
{
   "id": int,
   "text": "string",
   "isActive": true,
   "isComplete": false
}
```

Response:

```
[
   {
      "id": int,
      "text": "string",
      "isActive": true,
      "isComplete": false
   }
]
```

Permission: None

Method: clearAllCompleted (PUT)

Usage: updates the tasks in the Task table. This is used by our grid when “Clear Completed” buttons is pressed. This removes all the completed task from our grid as if the red cross button for those tasks was clicked.

URL: http://localhost:8080/task/clearAllCompleted

Request Body: None

Response:

```
[
  {
    "id": int,
    "text": "string",
    "isActive": true,
    "isComplete": false
  }
]
```

Permission: None

Method: markAllCompleted (PUT)

Usage: updates the tasks in the Task table. This is used by our grid when “Mark all Completed” buttons is pressed. This mark all the incomplete tasks in our grid as if the checkbox for those tasks was clicked.

URL: http://localhost:8080/task/markAllCompleted

Request Body: None

Response:

```
[
  {
    "id": int,
    "text": "string",
    "isActive": true,
    "isComplete": false
  }
]
```

Permission: None

Method: deleteTask/{id} (GET)

Usage: deletes the task from the Task table with the respective task id. This will not be used anywhere in our application but I have created it as it is one of the essential CRUD calls.
To manage the delete functionality I have used a “isActive” flag.
My understanding is, if a repository is expected to get too big and old items may someday become relevant (for analytics, etc) then the items should not be deleted but rather handled in this way.

URL: http://localhost:8080/task/deleteTask/{id}

Path Params:

- id:
  - type: Long
  - description: This is the id for which the API will delete the task.

Response: None

Permission: None