

# Task-API

---

## Create a Task

This route is used to create a new Task in the database.

- **URL**

/api/tasks/create

- **Method:**

POST

- **URL Params**

**Required:**

None

- **Data Params**

- **Required:**

a JSON object containing at least a title for the new Task. Other fields can be omitted.

```
{
  "title": "Task-title",
  "description": "a Task"
}
```

- **Success Response:**

- **Code:** 200

**Content:**

```
{
  "__v": 0,
  "title": "Implement a cool Feature",
  "_id": "5a3254dafbe56821a4d3957e",
  "time": 0,
  "done": false,
  "active": false,
  "visible": true,
  "updatedAt": "2017-12-14T10:39:21.264Z",
  "createdAt": "2017-12-14T10:39:21.264Z",
  "interval": {
    "unit": null,
    "value": 0,
    "hasInterval": false
  },
  "notes": [],
  "description": "a Task"
}
```

- **Error Response:**

- **Code:** 400

**Content:**

```
{
  "errors": {
    "title": {
      "message": "Path `title` is required.",
      "name": "ValidatorError",
      "properties": {
        "type": "required",
        "message": "Path `{PATH}` is required.",
        "path": "title"
      },
      "kind": "required",
      "path": "title",
      "$isValidatorError": true
    }
  },
  "_message": "Task validation failed",
  "message": "Task validation failed: title: Path `title` is required.",
  "name": "ValidationError"
}
```

## Get all Tasks

This route is used to get all Task objects from the database.

- **URL**

/api/tasks/find/all

- **Method:**

GET

- **URL Params**

- **Required:**

None

- **Data Params**

- **Required:**

None

- **Success Response:**

- **Code:** 200

**Content:** { TODO insert response here }

- **Error Response:**

- **Code:** 400

**Content:** { TODO insert response here }

- **Sample Call:**

```
//TODO insert sample call here
```

## Get a Task by \_id

This route is used to get a single Task object from the database.

- **URL**

/api/tasks/find/:id

- **Method:**

GET

- **URL Params**

- **Required:**

:id - the \_id of the requested Task

- **Data Params**

- **Required:**

None

- **Success Response:**

- **Code:** 200

**Content:** { TODO insert response here }

- **Error Response:**

- **Code:** 400

**Content:** { TODO insert response here }

- **Sample Call:**

```
//TODO insert sample call here
```

## Update a Task

This route is used to update Task objects.

- **URL**

/api/tasks/update/:id

- **Method:**

PUT

- **URL Params**

- **Required:**

:id - the \_id of the requested Task

- **Data Params**

- **Required:**

a json object containing all fields that should be updated

```
{  
  "description": "the new description"  
}
```

- **Success Response:**

- **Code:** 200  
**Content:** { TODO insert response here }

- **Error Response:**

- **Code:** 400  
**Content:** { TODO insert response here }

- **Sample Call:**

```
//TODO insert sample call here
```

## Delete a Task

This route is used to delete Task objects from the database.

- **URL**

/api/tasks/delete/:id

- **Method:**

DELETE

- **URL Params**

- **Required:**

:id - the \_id of the requested Task

- **Data Params**

- **Required:**

None

- **Success Response:**

- **Code:** 200  
**Content:** { TODO insert response here }

- **Error Response:**

- **Code:** 400  
**Content:** { TODO insert response here }

- **Sample Call:**

```
//TODO insert sample call here
```

## Mail-API

---

### Send a Mail

This route is used to send a mail via smtp.

- **URL**

/api/emails/send

- **Method:**

POST

- **URL Params**

- **Required:**

None

- **Data Params**

- **Required:**

mailReceiver, mailSubject and mailContent

```
{
  "mailReceiver": "YOURMAILADDRESS@PROVIDER",
  "mailSubject": "Test",
  "mailContent": "<p>Test <b style=\"color: green;\">successful</b> if you can read this
in your mail app.</p>"
}
```

- **Optional:**

cclist: list of cc addresses delimited by , (no spaces)

- **Success Response:**

- **Code:** 200

**Content:**

```
{
  "status": "success",
  "mail": {
    "from": "mailservice.tms@gmail.com",
    "to": "YOURMAILADDRESS@PROVIDER",
    "subject": "Test",
    "html": "<p>Test <b style=\"color: green;\">successful</b> if you can read this in
your mail app.</p>"
  }
}
```

- **Error Response:**

- **Code:** 400

**Content:**

```
{
  "status": "failure",
  "mail": {
    "from": "mailservice.tms@gmail.com",
    "to": "YOURMAILADDRESS@PROVIDER",
    "subject": "Test",
    "html": "<p>Test <b style=\"color: green;\">successful</b> if you can read this in
your mail app.</p>",
    "cc": "CCMAIL1,CCMAIL2"
  }
}
```

OR

- **Code:** 400

**Content:**

```
{  
  "status": "invalid request syntax"  
}
```

- **Sample Call:**

```
//TODO insert sample call here
```