

# /users route

1. GET /users/:userID :

Gets one user using their user id.

- Response body :

```
{
  "data": {
    "email": "example@email.com",
    "id": 1,
    "role": 1,
    "username": "example-username"
  }
}
```

2. GET /users?email={email}&username={username} :

Gets one user using their email and username.

- Response body :

```
{
  "email": "example@email.com",
  "id": 1,
  "role": 1,
  "username": "example-username"
}
```

3. POST /users :

Adds one user.

Parameter	Type	Description
Username	String	An username
Email	String	The user email
Role (optional)	Integer	The user role's id

\* Request **body** :

```
{
  "user": {
    "username": "example-username",
    "email": "example@email.com",
    "role": 1 (optional)
  }
}
```

\* Response **body** :

```
{
  "data": {
    "email": "example@email.com",
    "id": 1,
    "role": 1,
    "username": "example-username"
  }
}
```

#### 4. PUT /users/:userID :

*Editing an user using their user id.*

Parameter	Type	Description
Username	String	An username
Email	String	The user email
Role (optional)	Integer	The user role's id

\* Request **body** :

```
{
  "user": {
    "username": "edited-username", (optional)
    "email": "example@email.com", (optional)
    "role": 1 (optional)
  }
}
```

\* Response **body** :

```
{
  "data": {
    "email": "example@email.com",
    "id": 1,
    "role": 1,
    "username": "edited-username"
  }
}
```

#### 5. DELETE /users/:userID :

*Deletes an user using their user id.*

## /workingtimes route

#### 1. GET /workingtimes/:userID?start={datetime}&end={datetime} :

*Gets all the workingtimes of an user using their user id, within a start and an end date.*

- Response body :

```
[
  {
    "end": "2020-10-12T15:18:12.000000",
    "id": 1,
    "start": "2020-10-10T15:17:12.000000"
  }
]
```

## 2. GET /workingtimes/:userID/:workingtimeID :

*Gets one workingtime using its working time id and the user id.*

- Response body :

```
{
  "end": "2020-10-12T15:18:12.000000",
  "id": 1,
  "start": "2020-10-10T15:17:12.000000"
}
```

## 3. POST /workingtimes/:userID :

*Adds one workingtime for an user.*

Parameter	Type	Description
Start	UTC datetime	The workingtime date & time
End	UTC datetime	The workingtime date & time

\* Request **body** :

```
{
  "start": "2020-10-20T15:19:12Z",
  "end": "2020-10-20T15:19:12"
}
```

\* Response **body** :

```
{
  "end": "2020-10-12T15:18:12.000000",
  "id": 1,
  "start": "2020-10-10T15:17:12.000000",
  "user_id": 1
}
```

## 4. PUT /workingtimes/:workingtimeID :

*Modifies one workingtime using its id.*

Parameter	Type	Description
Start	UTC datetime	The workingtime date & time

Parameter	Type	Description
End	UTC datetime	The workingtime date & time
User id (optional)	Integer	The worktime user id

\* Request **body** :

```
{
  "workingtime": {
    "start": "2020-10-10T15:17:12Z",
    "end": "2020-10-20T16:19:12",
    "user_id": 1 (optional)
  }
}
```

\* Response **body** :

```
{
  "data": {
    "end": "2020-10-20T16:19:12Z",
    "id": 1,
    "start": "2020-10-10T15:17:12Z",
    "user_id": 1
  }
}
```

## 5. DELETE /workingtimes/:workingtimeID :

*Deletes a workingtime using its id*

# /clocks route

## 1. GET /clocks/:userID :

*Gets all the clocks of an user using its user id*

- Response body :

```
[
  {
    "id": 1,
    "status": true,
    "time": "2020-10-12T08:18:12.000000",
    "user_id": 1
  },
  {
    "id": 2,
    "status": false,
    "time": "2020-10-12T16:18:12.000000",
    "user_id": 1
  }
]
```

## 2. POST /clocks/:userID :

*Adds one clock for an user using its user id*

Parameter	Type	Description
Time	UTC datetime	The clock date & time
Status	Boolean	True clocking-in, False clocking-out

\* Request **body** :

```
{
  "time": "2020-10-13T08:30:12Z",
  "status": "true"
}
```

\* Response **body** :

```
{
  "data": {
    "id": 3,
    "status": true,
    "time": "2020-10-13T08:30:12.000000"
  }
}
```

### 3. GET /clocks/last/:userID

*Get the last clock for the user by its user id.*

- Response body :

```
{
  "id": 3,
  "status": true,
  "time": "2020-10-13T08:30:12.000000"
}
```

## /roles route

### 1. GET /roles :

*Get all the roles.*

- Response body :

```
{
  "data": [
    {
      "id": 1,
      "name": "Employee"
    },
    {
      "id": 2,
      "name": "Manager"
    },
    {
      "id": 3,
```

```
    "name": "General Manager"
  }
]
```

## 2. GET /roles/:roleID :

Get a role by id.

- Response body :

```
{
  "data": {
    "id": 2,
    "name": "Manager"
  }
}
```

## 3. GET /roles/:roleID/users :

Get all the users by role.

- Response body :

```
[
  {
    "email": "dscotcher9@sohu.com",
    "id": 10,
    "role": 2,
    "username": "mrunge9"
  },
  {
    "email": "manager@time.man",
    "id": 11,
    "role": 2,
    "username": "manager"
  }
]
```

## 4. POST /roles/ :

Create a new role.

Parameter	Type	Description
Role	String	The role's name

\* Request **body** :

```
{
  "role":{
    "name":"Manager"
  }
}
```

\* Response **body** :

```
{
  "data": {
    "id": 2,
    "name": "Manager"
  }
}
```

#### 5. PUT /roles/:roleID :

*Edit a role.*

Parameter	Type	Description
Role	String	The role's name

\* Request **body** :

```
{
  "role":{
    "name":"The Manager"
  }
}
```

\* Response **body** :

```
{
  "data": {
    "id": 2,
    "name": "The manager"
  }
}
```

#### 6. DELETE /roles/:roleID :

*Deletes a role using the role id.*

## /teams route

---

#### 1. GET /teams :

*Get all the teams.*

- Response body :

```
{
  "data": [
    {
      "id": 1,
      "members": [
        1,
        2,
        3,
        4,
        5
      ]
    },
  ],
}
```

```

{
  "id": 2,
  "members": [
    7,
    8,
    9,
    4,
    5
  ]
},
{
  "id": 3,
  "members": [
    1,
    2,
    3,
    7
  ]
},
{
  "id": 4,
  "members": [
    1,
    2,
    3
  ]
},
{
  "id": 5,
  "members": [
    4,
    5
  ]
},
{
  "id": 6,
  "members": [
    4,
    5,
    8
  ]
}
]
}

```

## 2. GET /teams/:managerID :

*Get all the teams using the manager's user id.*

- Response body :

```

[
  {
    "id": 1,
    "members": [
      1,
      2,
      3,
      4,
      5
    ],
    "name": "works",
    "user_id": 11
  },
  {
    "id": 4,
    "members": [
      1,
      2,
      3
    ],

```



```

    "name": "play",
    "user_id": 11
  },
  {
    "id": 5,
    "members": [
      4,
      5
    ],
    "name": "management",
    "user_id": 11
  },
  {
    "id": 6,
    "members": [
      4,
      5,
      8
    ],
    "name": "toto",
    "user_id": 11
  }
]

```

### 3. GET /teams/:managerID/:id :

*Get a team by the manager user id and the team id.*

- response body :

```

{
  "data": {
    "id": 1,
    "members": [
      1,
      2,
      3,
      4,
      5
    ]
  }
}

```

### 4. GET /member\_teams/:memberID :

*Get all the teams the member (by its user id) is assigned to.*

- Response body :

```

{
  "data": {
    "id": 1,
    "members": [
      1,
      2,
      3,
      4,
      5
    ]
  }
}

```

### 5. POST /teams/ :

*Create a new team.*

Parameter	Type	Description
User_id	Integer	The team's manager's user id
Name	String	The team's name
Members	Array	List of members's user id

\* Request **body** :

```
{
  "team":{
    "user_id":10,
    "name":"new team",
    "members":[3, 22]
  }
}
```

#### 6. PUT /teams/teamID :

*Edit a team.*

Parameter	Type	Description
User_id	Integer	The team's manager's user id
Name	String	The team's name
Members	Array	List of members's user id

\* Request **body** :

```
{
  "team":{
    "user_id":11,
    "name":"The team",
    "members":[3, 22]
  }
}
```

\* Response **body** :

```
{
  "data": {
    "id": 11,
    "name":"The team",
    "members": [
      3,
      22
    ]
  }
}
```

**7. DELETE /teams/:teamID :**

*Delete a team using its team id.*