

# mapsense

Task: Location-based Service API

## javascript

```
//user.model.js
const userSchema = new Schema(
  {
    username: {
      type: String,
      required: true,
      unique: true
    },
  },
)
```

## User

## javascript

```
//Code Snippet from user.routes.js
import { Router } from "express";
import { loginUser, logoutUser, refreshAccessToken, registerUser } from "../controllers/user";
import { verifyJWT } from "../middlewares/auth.middleware.js";

const router = Router()
router.route("/register").post(registerUser)
router.route('/login').post(loginUser)
//secured routes
router.route("/logout").post(verifyJWT,logoutUser)
router.route("/refresh-token").post(refreshAccessToken)
```


## POST Register

http://localhost:8000/api/register

## javascript

```
// Code Snippet from user.controller.js
const registerUser = asyncHandler( async (req, res) => {
  const {username,password} = req.body
```

```
console.log(`The username and password ${username} ${password}`)  
if(  
    [username,password].some((field)=>field?.trim()=== "")
```

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this ...

```
}  
const existedUser = await User.findOne({
```

## Body raw (json)

json

```
{  
  "username": "sharma01ketan@gmail.com",  
  "password": "sharma01ketan"  
}
```

## POST Login

http://localhost:8000/api/login

javascript

```
// Code Scippet from user.controller.js  
const loginUser = asyncHandler(async (req, res) =>{  
    const {username, password} = req.body  
    if (!username) {  
        throw new ApiError(400, "username is required")  
    }  
    const user = await User.findOne({  
        $or: [{username}]  
    }).populate('locations')  
    if (!user) {  
        throw new ApiError(404, "User does not exist")  
    }  
})
```

## HEADERS


Authorization

Bearer

## Body raw (json)

json

```
{  
  "username": "sharma01ketan@gmail.com".
```

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this ...

## POST Logout

http://localhost:8000/api/logout

### javascript

```
//Code Snipper from user.controller.js
const logoutUser = asyncHandler(async(req, res) => {
  await User.findByIdAndUpdate(
    req.user._id,
    {
      $unset: {
        refreshToken: 1 // this removes the field from document
      }
    },
    {
      new: true
    }
  )
})
```

## HEADERS

accessToken	"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NjNIODRiMWQ2ZjE0MzFmNTRmOWE0MTgiLCJ1c2VybmFtZSI6InNoeXJtYTAXa2V0YW4iLCJpYXQiOiJlMzMTUzNzMyMzgsImV4cCI6MTcxNTQ1OTYzOH0.qpFAUZqEZ_R4t7rO-EwAWLTtKNFsLoFQle_rRkge-hA"
refreshToken	"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NjNIODRiMWQ2ZjE0MzFmNTRmOWE0MTgiLCJpYXQiOiJlMzMTUzNzMyMzgsImV4cCI6MTcxNjIzOH0.GLNdmnOTemveFMqF2Qil-CEAePc6rNH8_Z5-qxOSfHs"

## Body raw (json)

### json

```
//NOT FUNCTIONING
//FRONTEND REQUEST IS WORKING PROPERLR
{
  "username": "sharma01ketan@gmail.com",
  "password": "sharma01ketan"
}
```

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this ...

#### javascript

```
//Code Snippet from location.routes.js
import { Router } from "express";
import { createLocation, calculateDistance, closestDistance, addLocation, allLocations, get
from "../controllers/location.controller.js";
const router = Router();
router.route("/location").post(createLocation);
router.route('/distance').post(calculateDistance)
router.route('/closest').post(closestDistance)
router.route('/addLocation').post(addLocation)
router.route('/allLocations').post(allLocations)
router.route("/:locationId").get(getLocationById);
```

## POST Location

http://localhost:8000/api/location

#### javascript


```
//Code Snippet from location.routes.js
const createLocation = asyncHandler(async (req, res) => {
  const { latitude, longitude } = req.body;
  console.log(`${latitude} ${longitude}createLocation`)
  if(
    [latitude,longitude].some((field)=>field?.trim()=== "")
  ){
    throw new ApiError(400,"All Fields are Required")
  }
  const existedLocation = await Location.findOne({
    $or:[{latitude,longitude}]
```

#### Body raw (json)

#### json

```
{
  "latitude": "40.741895",
  "longitude": "-73.989308"
}
```

## POST Distance

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this ...

#### javascript

```
//Code Snippet from location.routes.js
const calculateDistance = asyncHandler(async (req, res) => {
  const { coordinate1, coordinate2 } = req.body;
  console.log("The request body is:", req.body);

  if (!coordinate1.latitude || !coordinate1.longitude || !coordinate2.longitude || !coordinate2.latitude) {
    throw new ApiError(400, "All Fields are Necessary. Value passed as empty.");
  }

  const distance = haversine(coordinate1, coordinate2);
```

#### Body raw (json)

#### json

```
{
  "coordinate1": {
    "latitude": "40.7128",
    "longitude": "-74.0060"
  },
  "coordinate2": {
    "latitude": "34.0522",
    "longitude": "-118.2437"
  }
}
```

## POST Closest

http://localhost:8000/api/closest

#### javascript


```
//Code Snippet from location.routes.js
const closestDistance = asyncHandler(async (req, res) => {
  const { latitude, longitude } = req.body;
  console.log("The request body is:", req.body);

  if (!latitude || !longitude) {
    throw new ApiError(400, "All Fields are Necessary. Value passed as empty.");
  }

  const locations = await Location.find().select('latitude longitude -_id');
  if (locations.length === 0) {
```

## Body raw (json)

---

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this ...

json

```
{
  "latitude": "48.8566",
  "longitude": "2.3522"
}
```

---

## POST AddLocation

http://localhost:8000/api/addLocation

Plain Text

```
//Code Snippet from location.routes.js
const addLocation = asyncHandler(async (req, res) => {
  const { latitude, longitude, username } = req.body;
  console.log(req.body)
  console.log(`${latitude} ${longitude} ${username} addLocation`);
  try {
    const user = await User.findOne({
      $or:[{username}]
    })
    console.log(`User id is ${user._id}`)
    if (!user) {
```

## Body raw (json)

---

json


```
{
  "latitude": 34.741891,
  "longitude": -13.989308,
  "username": "sharma01ketan@gmail.com"
}
```

---

## POST AllLocations

http://localhost:8000/api/allLocations

---

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this ...

```
try {  
  const { username } = req.body; // Assuming username is sent in the request body  
  
  const user = await User.findOne({ username }).populate('locations')  
  
  if (!user) {  
    throw new ApiError(404, "User not found");  
  }  
}
```

### Body raw (json)

---

json

```
{  
  "username": "sharma01ketan@gmail.com"  
}
```