

# Socialization API

**All endpoints are protected and must be called with the authorization token passed in.**

`add_like()`:

Endpoint Route: `social/add_like`

Like a particular stock for the user. Expected input: `{"tickr": <valid_tickr>}`. Returns

`{"msg": "like added"}` on success.

`remove_like()`:

Endpoint Route: `social/remove_like`

Remove a like from a particular stock for the user. Expected input: `{"tickr": <valid_tickr>}`. Returns

`{"msg": "like removed"}` on success.

`user_likes()`:

Endpoint Route: `social/user_likes`

Shows all the stocks like by the user. Returns list of stocks likes by user

`total_likes()`:

Endpoint Route: `social/total_likes`

Gives the number of likes on a particular stock. Expected input: `{"tickr": <valid_tickr>}`. Returns numbers of likes on success.

`all_likes()`:

Endpoint Route: `social/all_likes`

Gives all the users that like a particular stock. Expected input: `{"tickr": <valid_tickr>}`. Returns list of users of success.

`add_comment()`:

Endpoint Route: `social/add_comment`

Adds comment on a particular stock for user. Expected input: `{"tickr": <valid_tickr>, "comment": "user_comment"}`. Returns `{"msg": "comment added"}` on success.

`edit_comment()`:

Endpoint Route: `social/edit_comment`

Edits a comment on a particular stock made by the user. Expected input: `{"time_stmp": <time stamp for the original comment>, "comment": <edited comment>}`. Returns `{"msg": "comment edited"}` on success. The original timestamp can be obtained by calling `userComments()` to get the comments on a particular stock made by the user.

`remove_comment()`:

Endpoint Route: `social/remove_comment`

Edits a comment on a particular stock made by the user. Expected input: `{"time_stmp": <time stamp for the original comment>}`. Returns `{"msg": "comment removed"}` on success. The original timestamp can be obtained by calling `userComments()` to get the comments on a particular stock made by the user.

`total_comments()`:

Endpoint Route: `social/total_comments`

Gives total number of comments for a stock. Expected input: `{"tickr": <valid_tickr>}`. Returns number of comments on success.

`user_comments()`:

Endpoint Route: `social/user_comments`

Gives all comments made on a particular stock by the user. Expected input: `{"tickr": <valid_tickr>}`. Returns list of comments on success.

`fetch_latest_comments()`:

Endpoint Route: `social/fetch_latest_comments`

Gives the 5 most recent comments on a stock . Expected input: `{"tickr": <valid_tickr>}`. Returns list of comments on success.

`all_comments()`:

Endpoint Route: `social/all_comments`

Gives all comments made on a particular stock . Expected input: `{"tickr": <valid_tickr>}`. Returns list of comments on success.

To test the above endpoints by hand, do the following:

1. Copy and run the following code in App.js. to log in as the test user

```
const [token, setToken] = useState([])

//To log in as test user
useEffect(() => {
  fetch('/auth/login', {
    method: 'POST',
    body: JSON.stringify({
      "username": "john.doe", "password": "John@12345"
    })
  }).then(
    res => res.json()
  ).then(
    token => {
      setToken(token);
      console.log(token)
    }
  )
}, [])
```

2. The browser console should spit out an authorization token. Copy the following code into App.js. Then, replace `<JWT_Token>` with that token; replace `<desired_endpoint>` with the endpoint that needs testing; and replace `<arguments>` with the appropriate JSON of inputs for the desired endpoint.

```

const [data, setData] = useState([])
//To test protected endpoints. Use JWT token outputted by code above
//Prepend all portfolio endpoints with '/portfolio'
//prepend all user authentication endpoints with '/auth'
useEffect(() => {
  fetch(<desired_endpoint>, {
    method: 'POST',
    headers: new Headers({
      'Authorization': 'Bearer ' + <JWT_token>
    }),
    body: JSON.stringify({
      <arguments>
    })
  } ).then(
    res => res.json()
  ).then(
    data => {
      setToken(data);
      console.log(data)
    }
  )
}, [])

```