React

CALLING BACKEND SERVICES

Code Base

https://1drv.ms/f/s!AtGKdbMmNBGdg_MrgNlXzocRVwfieA

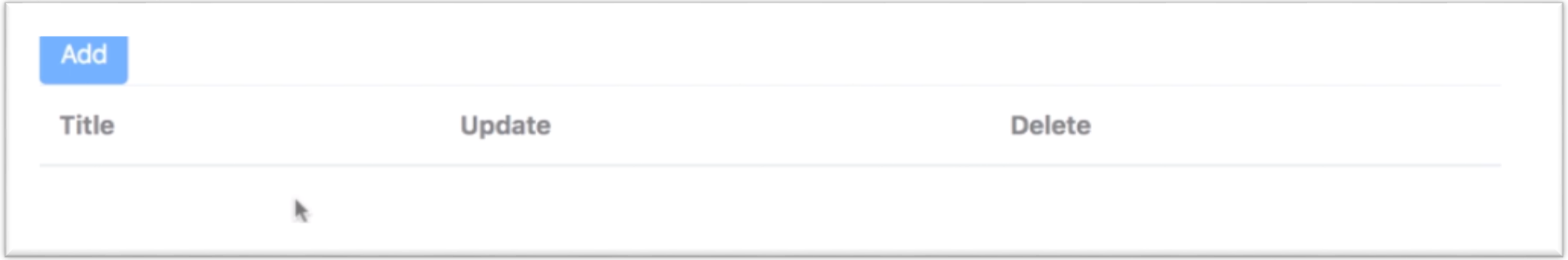
As Usual

Start and End

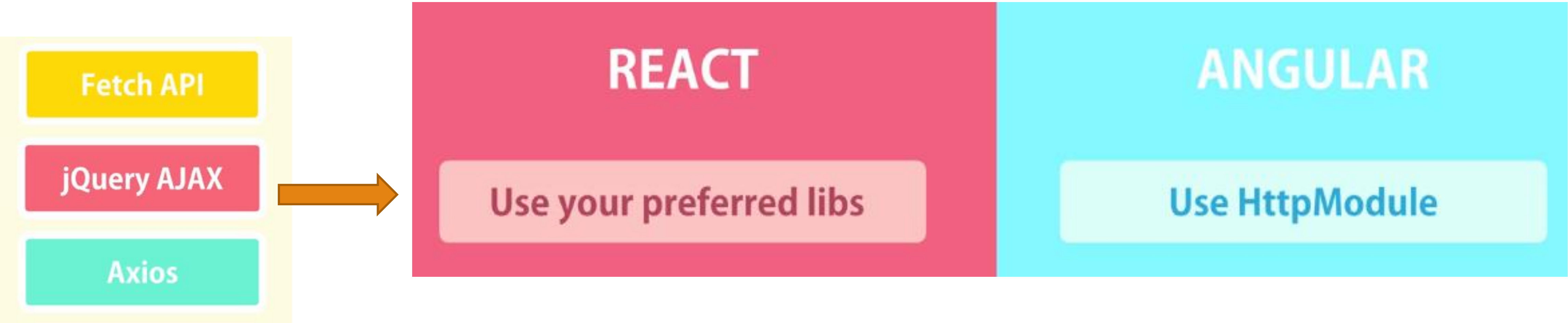
http-app

```
>npm install
```

```
>npm start
```



React Vs Angular



Fetch API

jQuery AJAX

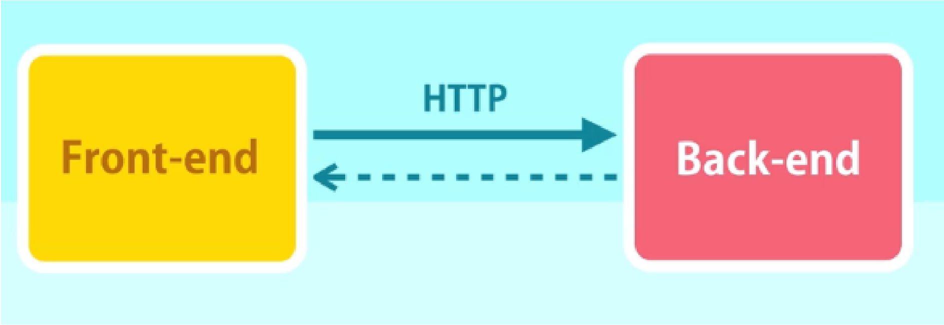
Axios

REACT

Use your preferred libs

ANGULAR

Use HttpModule

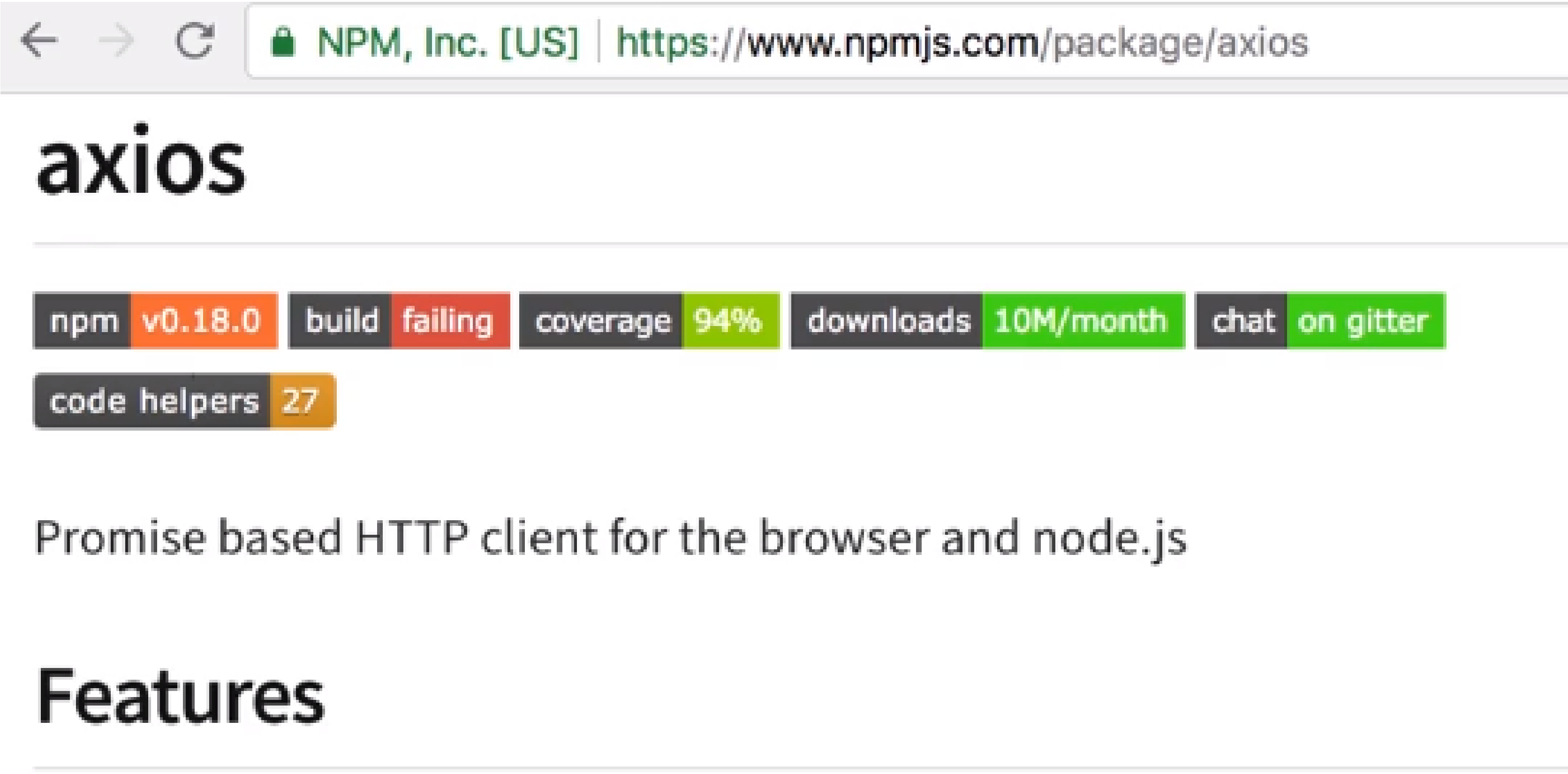


Front-end

HTTP

Back-end

>npm i axios



NPM, Inc. [US] https://www.npmjs.com/package/axios

axios

npm v0.18.0 build failing coverage 94% downloads 10M/month chat on gitter

code helpers 27

Promise based HTTP client for the browser and node.js

Features

state = { persons: [] }

```
componentDidMount() {
  axios.get(`https://jsonplaceholder.typicode.com/users`)
   .then(res => {
    const persons = res.data;
    this.setState({ persons });
   })
 }
```

Post Request

```
axios.post(`https://jsonplaceholder.typicode.com/users`, { user })
    .then(res => {
      console.log(res);
      console.log(res.data);
    })
```

Delete Request

```
axios.delete(`https://jsonplaceholder.typicode.com/users/${this.state.id}`)
      .then(res => {
        console.log(res);
        console.log(res.data);
      })
```

Base Instance

```
import axios from 'axios';
export default axios.create({
  baseURL: `http://jsonplaceholder.typicode.com/`
});

//api.js
```

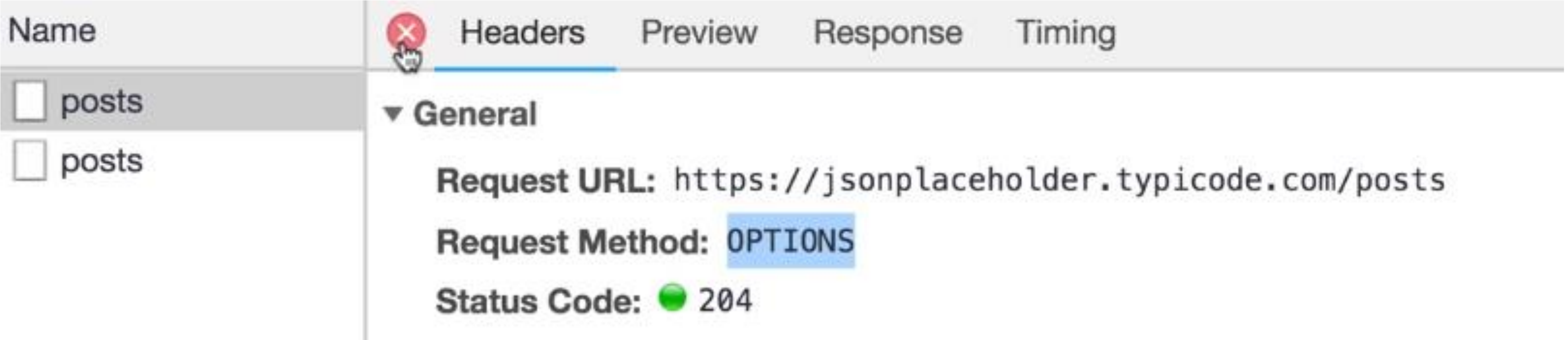
```
import API from '../api';
export default class PersonList extends React.Component {
  handleSubmit = event => {
    event.preventDefault();
    API.delete(`users/${this.state.id}`)
      .then(res => {
        console.log(res);
        console.log(res.data);
      })
  }
}
```

Using Async Wait

```
handleSubmit = async event => {
  event.preventDefault();
  // Promise is resolved and value is inside of the response const.
  const response = await API.delete(`users/${this.state.id}`);
  console.log(response);
  console.log(response.data);
};
```

Request Life Cycle

Browser send an options request in addition to actual request to cross domain api call



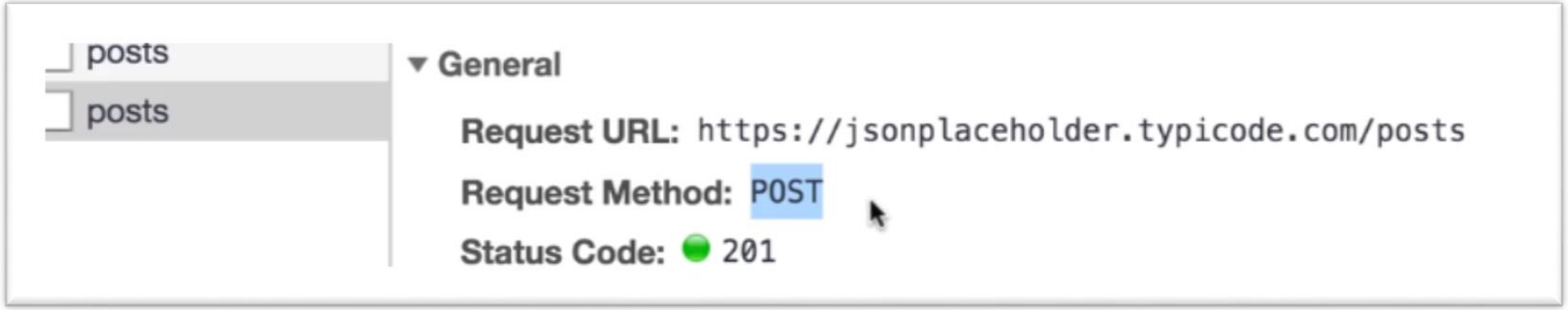
Request URL: https://jsonplaceholder.typicode.com/posts

Request Method: OPTIONS

Status Code: ● 204

Request Life Cycle

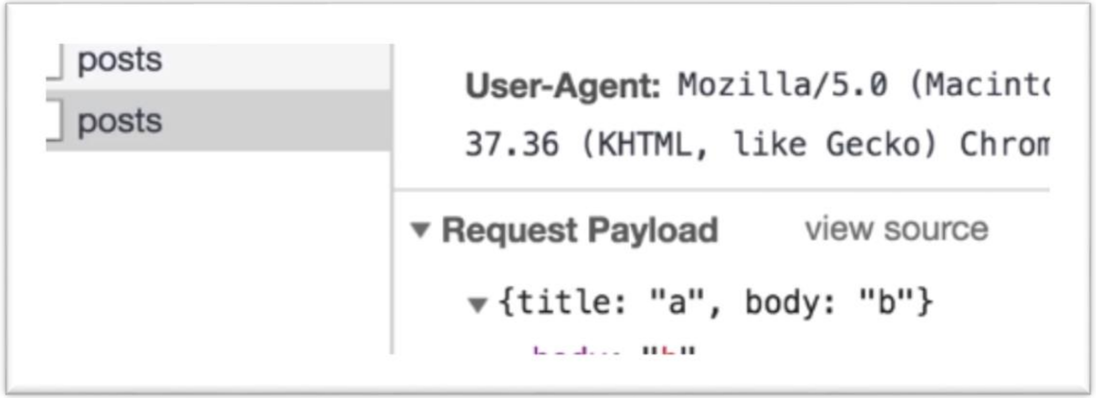
Actual Request with 201 status. Successful post



Request URL: https://jsonplaceholder.typicode.com/posts

Request Method: POST

Status Code: ● 201



User-Agent: Mozilla/5.0 (Macint
37.36 (KHTML, like Gecko) Chrom

▾ Request Payload view source

```
▾{title: "a", body: "b"}
```

More Cleaner approach

```
import axios from "axios";
import logger from "./logService";
import { toast } from "react-toastify";

axios.interceptors.response.use(null, error => {
const expectedError =
error.response &&
error.response.status >= 400 &&
error.response.status < 500;

if (!expectedError) {
```

More Cleaner approach

```
logger.log(error);
toast.error("An unexpected error occurrred.");
}
return Promise.reject(error);});

export default {
get: axios.get,
post: axios.post,
put: axios.put,
delete: axios.delete
};
```

React Application Wide Notifications

```
import { ToastContainer, toast } from 'react-toastify';
  import 'react-toastify/dist/ReactToastify.css';

__________________________________________

<ToastContainer />
```

Optimistic Vs Pessimistic approach

Update State Then Send Request (What if req fails?)

Vs

Send Request Then Update State

Optimistic Delete Approach

```
handleDelete = async post => {
    const originalPosts = this.state.posts;
    const posts = this.state.posts.filter(p => p.id !== post.id);    this.setState({ posts });
    try {       await http.delete(config.apiEndpoint + "/" + post.id);     }
catch (ex) {
      if (ex.response && ex.response.status === 404)
        alert("This post has already been deleted.");      this.setState({ posts: originalPosts
});
    }
  };
```

Expected Vs Unexpected Errors

EXPECTED: (404: not found: bad request) – CLIENT ERRORS

- Display a specific Errors

UNEXPECTED: (network down, server down, db down, bug)

- Log Them
- Display a generic or friendly error

```
const expectedError =
error.response &&
error.response.status >= 400 &&
error.response.status < 500;
```

Log Service

```
import Raven from "raven-js";
function init() {
// Raven.config("ADD YOUR OWN API KEY", {
// release: "1-0-0",
// environment: "development-test"
// }).install();
}
```

Log Service

```
function log(error) {
// Raven.captureException(error);
}
export default {
init,
log
};
```

Vidly Backend API

https://github.com/mosh-hamedani/vidly-api-node

Same code is also attached in start section of

https://1drv.ms/f/s!AtGKdbMmNBGdg_MrgNlXzocRVwfieA

Or if git is installed

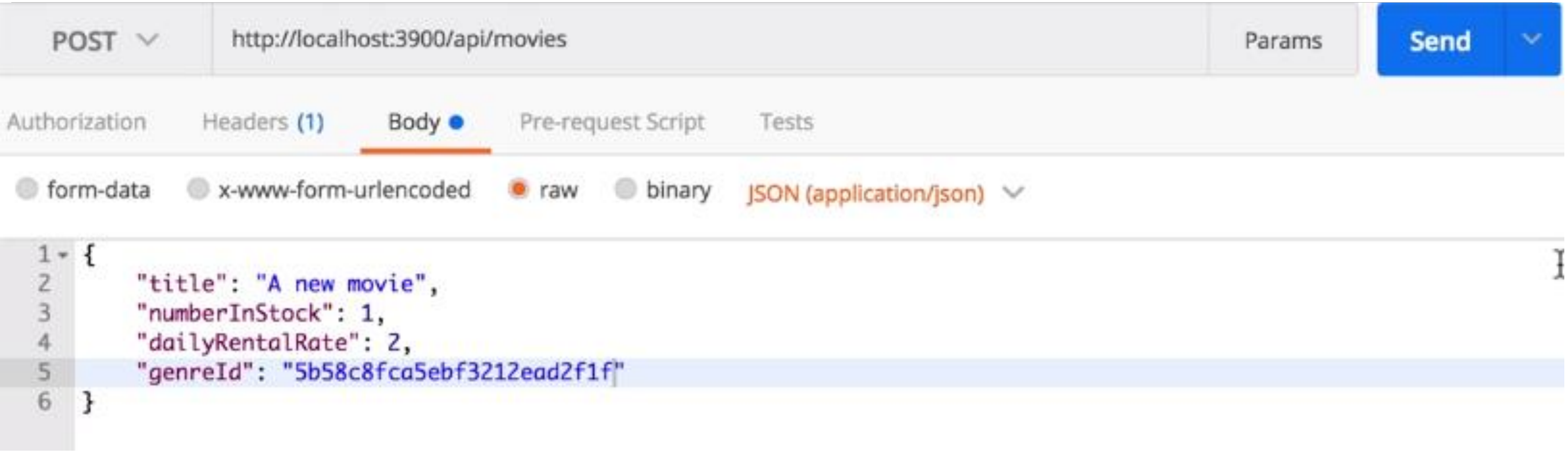
>git clone https://github.com/mosh-hamedani/vidly-api-node.git

>cd vidly-api-node

Access genres @ http://localhost:3900/api/genres

Disable auth in configs

Send a POST Request



POST http://localhost:3900/api/movies

● raw JSON (application/json)

```
{
    "title": "A new movie",
    "numberInStock": 1,
    "dailyRentalRate": 2,
    "genreId": "5b58c8fca5ebf3212ead2f1f"
}
```

Code Sample

https://1drv.ms/f/s!AtGKdbMmNBGdg_MrgNlXzocRVwfieA

Contains an API built in express with Mongo DB

API is consumed in React

MERN Stack: DONE