Ask Company

리액트와 함께 장고 시작하기 / 리액트

Axios 라이브러리를 활용한 HTTP 요청

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

axios 라이브러리

Promise based HTTP client for the browser and node.js

설치> yarn add axios

```
function getPost () {
  axios.get("http://example.com/posts/12345/")
    .then(response => {
      console.log(response);
    })
    .catch(error => {
      console.error(error);
    });
}
```

Django 서버로 Ajax 요청을 할 경우 //
장고가 서비스되는 호스트명과 리액트가 서비스되는 호스트명이
다르면, cors headers 설정이 필요합니다.

```
async function getPost () {
  try {
    const response = await axios.get("http://example.com/posts/12345/");
    console.log(response);
  }
  catch (error) {
    console.error(error);
  }
}
```

async/await는 ECMAScript2017 문법

https://github.com/axios/axios 공식문서를 체크하는 습관 !!!

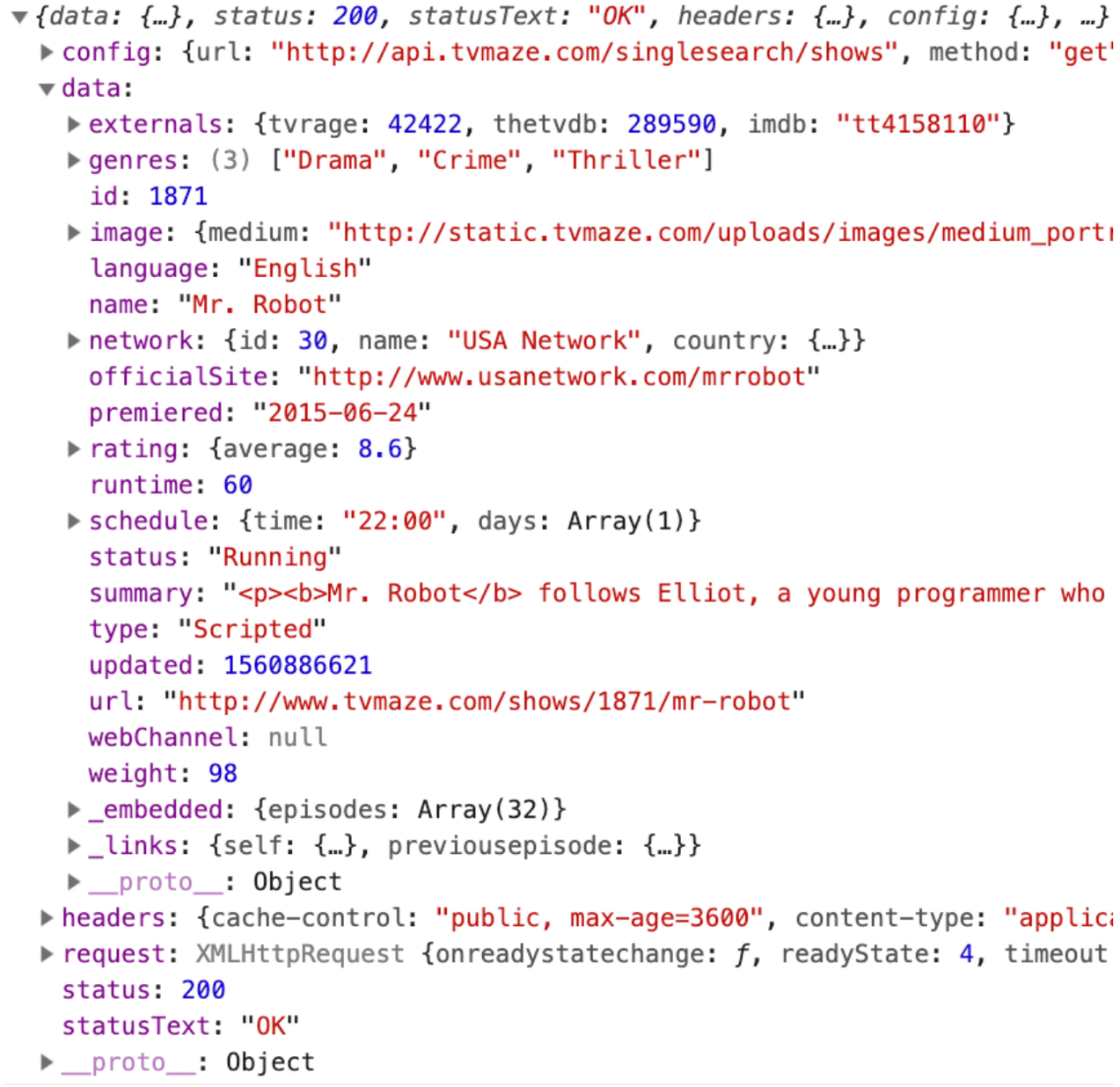
```jsx
import axios from 'axios';

const App = () => {
  const [episodeList, setEpisodeList] = React.useState([]);

  React.useEffect(
    () => {
      async function fetch() {
        const query = 'mr-robot';
        const params = { q: query, embed: 'episodes' };
        const response = await axios.get('http://api.tvmaze.com/singlesearch/shows', { params });
        setEpisodeList(response.data._embedded.episodes);
      }
      fetch();
    }, []);

  return (
    <ul>
      {episodeList.map(post => (
        <li key={post.id}>
          {post.name} <img src={post.image.medium} alt={post.name} />
        </li>
      ))}
    </ul>
  );
};
```

async 함수에서만 await를 사용할 수 있습니다.
useEffect에는 async 함수를 지정할 수 없습니다.

config 인자

Response Schema

```
{
  // 서버 응답 JSON 객체
  data: {},

  // 서버 응답의 HTTP 상태 코드
  status: 200,

  // 서버 응답의 HTTP 상태 메세지
  statusText: 'OK',

  // 서버 응답 헤더. 모든 헤더명은 소문자
  headers: {},

  // 요청에 사용된 axios 설정
  config: {},

  // 요청 객체
  request: {}
}
```



https://github.com/axios/axios#response-schema

axios API

요청 메소드 aliases

axios.request(config), axios.get(url[, config]), delete, head, options

axios.post(url[, data[, config]]), put, patch

Concurrency

axios.all(iterable)로 요청하고 axios.spread(callback)로 응답을 받아서 처리

config가 적용된 instance

```
const instance = axios.create({
  baseURL: 'https://some-domain.com/api/',
  timeout: 1000,
  headers: {'X-Custom-Header': 'foobar'}
});
```

요청 설정

```
url: '/user/'
method: 'get'
baseURL: 'https://some-domain.com/api/'
headers: { 'X-Requested-With': 'XMLHttpRequest' }
params: { q: 'Django' }
data: { firstName: 'Fred' }
timeout: 1000                  → 디폴트 : 0 (no timeout)
auth: { username: '', password: '' }
responseType: 'json' → arraybuffer, document, json (디폴트), text, stream
onUploadProgress = (progressEvent) => {}
onDownloadProgress: (progressEvent) => {}
validateStatus: (status) => (status >= 200 && status < 300)
cancelToken: new CancelToken(function(cancel) {})
```

https://github.com/axios/axios#request-config

Life is short.
You need Python and Django.

I will be your pacemaker.