**vue와 ajax 요청**

vue에서 ajax 요청을 하는 방법은 여러가지가 있습니다. 순수 자바스크립트, jQuery를 사용하여 작업할 수도 있고, [vue-resource](https://github.com/pagekit/vue-resource)와 [axios](https://github.com/mzabriskie/axios)도 있습니다. 이번에는 axios를 사용하는 방법을 소개 합니다.

**왜 vue-resource를 사용하지 않나요?**

vue-resource는 현재(2017년 1월) github에서 3600여개의 스타를 받은 성공적인 프로젝트 입니다. vue와 함께 잘 작동합니다. 하지만, 2016년 9월 이후 업데이트가 되지 않고 있습니다. 해결되지 않은 이슈가 무려 100여개가 넘습니다. 그리고 결정적으로 vue의 Evan You는 자신의 Medium에서 공식적으로 추천하는 프로젝트가 아니라고 말합니다. 자세한 내용은 [이 글](https://medium.com/the-vue-point/retiring-vue-resource-871a82880af4#.re69qas2z)을 확인하세요.

**axios를 사용하세요**

axios는 현재 가장 성공적인 HTTP 클라이언트 라이브러리 중 하나입니다. 아직 1.x 버전이 릴리즈 되지 않았지만 스타가 1만개가 넘을 정도로 인기가 좋습니다. 특별히 언급할만한 부분은 요청 취소와 TypeScript를 사용할 수 있는 것 입니다. 이 글에서는 기본적인 vue와 axios의 사용 방법을 알아봅니다. [axios의 github 프로젝트](https://github.com/mzabriskie/axios)를 살펴보시면 더 많은 내용을 익힐 수 있을 것입니다.

**axios 설치**

[vue-cli](https://github.com/vuejs/vue-cli)를 사용하고 계신다면 간단하게 추가할 수있습니다. vue-cli에 대한 간략한 설명은 vue.js 한국어 사용자 모임의 [한국어 번역](https://github.com/vuejs-kr/vue-cli)을 확인하세요.

터미널에서 **npm** 명령어를 이용하여 설치합니다.

npm install --save axios

이제 **main.js** 파일을 열어 axios를 추가하고 vue 앱의 전역으로 사용할 수 있도록 메소드를 추가하면 됩니다.



간단하게 추가하였습니다. 전역으로 사용하고 싶지 않고 **필요한 컴포넌트 안에서만** 사용하고 싶으면, 각 컴포넌트에서 import를 이용하여 사용하면 됩니다.

이를 이용해서 간단한 앱을 만들어 보겠습니다. [JSONPlaceholder](https://jsonplaceholder.typicode.com/)는 모의 REST API 테스트를 위한 서비스 입니다. 간단한 앱의 전체 코드 입니다.

<template>

<div id="app">

<div v-if="hasResult">

<div v-for="post in posts" v-bind:key="post.id">

<h1></h1>

<p></p>

</div>

</div>

<button v-else v-on:click="searchTerm">글 불러오기</button>

</div>

</template>

<script>

import Hello from './components/Hello'

export default {

name: 'app',

data: function () {

return {

posts: []

}

},

computed: {

hasResult: function () {

return this.posts.length > 0

}

},

methods: {

searchTerm: function () {

*// using JSONPlaceholder*

const baseURI = '[https://jsonplaceholder.typicode.com](https://jsonplaceholder.typicode.com/)';

this.$http.get(`${baseURI}/posts`)

.then((result) => {

console.log(result)

this.posts = result.data

})

}

}

}

</script>

<style>

#app {

font-family: 'Avenir', Helvetica, Arial, sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

color: #2c3e50;

max-width: 560px;

}

</style>

결과 입니다.



**axios와 Internet Explorer**

IE는 Promise를 지원하지 않습니다. [MSDN](https://msdn.microsoft.com/ko-kr/library/dn802826(v=vs.94).aspx)에서 확인하실 수 있습니다. 이를 해결하려면 약간의 변경이 필요합니다. [axios의 polyfill관련 문서](https://github.com/mzabriskie/axios/blob/master/UPGRADE_GUIDE.md" \l "es6-promise-polyfill) 에 따르면 **es6-promise** 의 polyfill 기능을 활성화 해야한다고 조언합니다. **es6-promise**를 설치하겠습니다.

npm install --save es6-promise

이후 webpack의 설정 파일인 webpack.base.conf.js파일을 열어 Promise를 사용할 수 있도록 추가해줍니다.

require('es6-promise').polyfill();

*// 또는*

require('es6-promise/auto');

이제 IE에서 Promise를 사용할 수 있습니다. [es6-promise 문서](https://github.com/stefanpenner/es6-promise)를 참조하세요.

감사합니다.

도와주신분