

# Nuxt, Vue, Express로 프론트엔드와 백엔드를 하나의 서버 앱에서 사용하기

2018-10-28 · 5 min to read

Nuxt   Vue.js   Express.js



## SPA(Single Page Application)의 서버 렌더링

웹사이트의 내용을 페이지가 바뀔 때마다 완전히 새로 구성하지 않고 필요한 부분만 동적으로 변경하는 방식의 앱을 SPA라고 한다. 서버는 클라이언트에게 웹사이트에 필요한 필수적인 자원만 넘겨준 후 그 다음은 모두 브라우저가 처리하도록 한다. WWW가 생긴 후 오랜 기간동안 일반적인 웹사이트는 페이지가 변경될 때마다 서버에서 모든 콘텐츠를 다시 내려받는 방식이었다. 하지만 SPA는 그 전통적인 방식을 벗어난 것이며, 예전과는 비교할 수 없을 정도의 빠르고 쾌적한 사용자 경험을 제공할 수 있게 되었다.

그런데 SPA에게는 사용자에게는 보이지 않는 문제가 있다. 검색엔진같은 소프트웨어에게는 SPA로 만들어진 웹사이트는 아마 아무런 내용이 없는 빈 페이지로 보일 것이며, 심지어 모든 페이지의 head 태그 안에는 같은 내용이 들어가 있을 것이기 때문에 페이지의 구분도 불가능하다. 즉 검색 엔진 최적화(SEO, Search Engine Optimization)가 어려우며 검색 엔진에 웹사이트가 가지고 있는 다양한 콘텐츠가 노출되지 않는다. 더 많은 사용자 유입을 위해 검색 엔진의 힘을 빌려야 하는 언론, 쇼핑몰, 블로그 같은 웹사이트라면 SPA를 도입하기 망설여질 것이다.

그래서 개발자들은 이를 극복하기 위해 서버 기반 웹앱과 SPA를 결합하는 방법을 고안했다. 서버가 브라우저에 아무것도 없는 HTML 페이지를 전달하는 대신, 최초로 표시될 내용만큼은 모두 서버에서 확실히 구성한 후 브라우저에 보내자는 것이다. 그리고 그 이후부터는 SPA처럼 동작하게 한다. 그러면 검색 엔진에게도 제대로 된 웹페이지가 보여질 것이고, 사용자에게도 여전히 빠른 웹사이트를 제공할 수 있을 것이다. 이 방식을 서버 사이드 렌더링, 또는 Universal Web App이라고 부른다. 서버 사이드 렌더링 아이디어가 없었다면 SPA의 활용 범위는 지금보다 훨씬 더 좁았을 것이다.

# 프레임워크

Nuxt  는 Vue.js를 프론트엔드 레이어로 하는 서버 사이드 렌더링 프레임워크다. React 서버 사이드 렌더링 프레임워크인 Next.js와 이름이 유사하다는 점에서 추측할 수 있겠지만, 둘 다 Zeit  팀에서 시작한 오픈소스 프레임워크다.


서버 사이드 렌더링 아이디어 자체는 간단하지만, 그걸 구현하는 것은 그리 녹록치 않은 일이다. 앱의 시작 지점이 서버니까 일단 서버 앱이 있어야 하며, 클라이언트 스크립트도 이 페이지가 서버에서 시작하는지, 웹브라우저에서 시작하는지를 구분해서 처리해야 하고, 라우트별로 다른 API를 호출해서 HTML의 head와 body를 구성해야 하는 등 할 일이 많다. Vue가 등장하기 전이며 React는 0.1x였던 시기에는 프로젝트 초기 구성 자체가 큰 hurdle이었던 걸로 기억한다.

하지만 이제는 Next.js, Nuxt.js라는 멋진 프레임워크가 있기에 처음부터 끝까지 모든 것을 제어하고 싶은 사람이 아니라면, 아주 간단히 서버 사이드 렌더링을 구현할 수 있다. 특히 프레임워크 차원에서 개발자에게 필요한 각종 유용한 기능을 제공하기 때문에 더욱 좋다.

그런데 문득 이런 생각이 들었다. 프론트엔드 앱의 시작 지점이 서버라면, 굳이 API 서버 앱을 별도로 둘 필요 없이 예전처럼 하나의 앱으로 개발하면 되지 않을까? Express.js라는 훌륭한 자바스크립트 서버 프레임워크도 있고 말이다. 하지만 자바스크립트가 아닌 다른 언어를 사용하는 백엔드 개발자가 더 많은데다, SPA 프레임워크를 사용하면서 서버 개발자가 마크업을 건드릴 일이 사라졌으니 반드시 같이 있어야 할 필요가 없기도 하다. 최근에는 서버리스 컴퓨팅이나 마이크로 서비스 아키텍처까지 트렌드를 타고 있어서 더욱 그렇다. 하지만 공부나 사이드 프로젝트를 위해 혼자 앱을 개발할 때는 안성맞춤일 것 같고, 풀 스택 개발자들이 함께 앱을 만들어 나갈때도 좋을 것 같다.

## Nuxt로 만든 앱에 API 라우트를 연결하는 과정

### Nuxt.js 앱 설치

Nuxt는 간편한 설치 과정  을 제공한다. 설치 과정에서 여러가지 옵션을 선택할 수 있는데, Custom Server 옵션으로 express를 선택하도록 하자.

```
> npx create-nuxt-app uni-febe
npx: 401개의 패키지를 11.523초 만에 설치했습니다.
```

```
? Project description My world-class Nuxt.js project
? Use a custom server framework
  none
> express
  koa
  adonis
  hapi
  feathers
```

## nuxt.config.js에서 미들웨어 설정

설치가 완료되고 `npm run dev` 명령어를 실행하면 앱이 잘 실행되는 것을 확인할 수 있다. 옵션 선택 과정에서 `prettier`나 `eslint`를 선택했으면 문법 오류 때문에 빌드가 안될 수 있는데, 그럴 땐 일단 `webpack`에서 관련 설정을 임시로 제거하도록 하자.

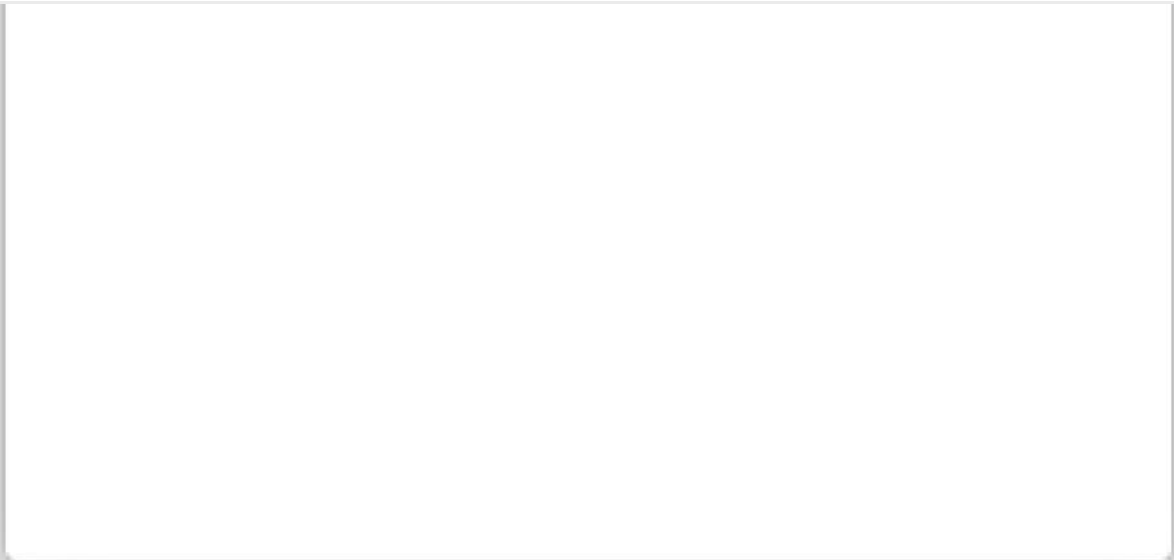
```
// nuxt.config.js
...

build: {
  /*
  ** You can extend webpack config here
  */
  extend(config, ctx) {
    // Run ESLint on save
    if (ctx.isDev && ctx.isClient) {
      // 이 부분을 주석 처리하고 코드는 나중에 정리하자
      config.module.rules.push({
        enforce: 'pre',
        test: /\.?(js|vue)$/ ,
        loader: 'eslint-loader',
        exclude: /(node_modules)/
      })
    }
  },
},
...

```

그리고 `nuxt.config.js` 파일에서 `serverMiddleware` 속성에 새로운 값을 할당한다.





이제 express가 미들웨어로 연결되었으니 여기서부터는 일반적인 express 앱 개발을 진행하면 된다. 앱에 라우트를 추가하고, 미들웨어를 추가하고, 데이터베이스를 연결하면 API 서버가 되는 것이다. 조금 더 상세한 예제는 템플릿 프로젝트 [🔗](#)를 참조하도록 하자.

## 프론트엔드에서 API 호출

이제 Vue 페이지에서 API 라우트를 호출해서 컴포넌트에 데이터를 연결하도록 하자. Nuxt는 Vue 컴포넌트에서 페이지 로딩 전에 비동기 데이터를 호출하기 위한 메소드, head 태그를 커스터마이징하기 위한 메소드를 제공하기 때문에 무척 편리하다. 아래 예제 코드는 `/api/user/:id` 라우트가 `name` 속성을 가진 JSON 데이터를 제공한다고 가정하고 작성한 컴포넌트다.

```
<template>
  <div>
    <!-- asyncData 메소드에서 전달한 데이터를 마크업에 연결 -->
    이름 : {{user.name}}
  </div>
</template>

<script>
export default {
  // Nuxt.js에서 제공하는 페이지 로딩 전에 비동기 데이터를 호출하기 위한 메소드
  // Vue.js의 data 메소드와 유사한 역할을 한다
  async asyncData({ app, query }) {
    // 설치 과정에서 axios 플러그인을 선택했다면 axios를 아래와 같은 형태로 사용 가능하다.
    const user = await app.$axios.$get(`/api/user/${query.id}`)
    return { user }
  },

  // Nuxt.js에서 제공하는 head 태그 수정용 메소드
  head() {
    return {
      title: `User | ${this.user.name}`, // asyncData에서 리턴한 user 데이터를 사용할 수 있다.
    }
  },
}
</script>
```

이렇게 Nuxt 프레임워크의 도움으로 간단하게 프론트, 백엔드가 합쳐진 서버 사이드 렌더링 SPA 앱이 만들어졌다.

## 참고 자료

- Installation - Nuxt.js [↗](#)
- Starter template for Nuxt.js with Express. [↗](#)
- Express routing [↗](#)



## [번역] 2018년에 배워야 할 최고의 ...

2년 전 · 댓글 9건

이 글은 Eric Elliot 의 Top JavaScript Libraries & Tech to Learn in 2018 을 ...

## [번역] 웹 아키텍처 입문

2년 전 · 댓글 7건

내가 웹 개발을 시작했을 때 알았으면 했던 기본 아키텍처 개념들 - 이 글은 ...

## [번역] 개발자를 위한 면접 지침

일년 전 · 댓글 한 건

또 다른 제목: 면접에서 회사를 상대로 질문하는 방법 - 이 글은 A developer's guide ...

## 블로그 정

2년 전 ·

블로그를 다 되어: 도 어느

댓글 2건 [blog.rhostem.com](#) [Disqus' Privacy Policy](#)

[로그인](#)

[추천](#) 2

[Tweet](#)

[공유](#)

[인기순](#)



## 토론 참여하기

[다음으로 로그인](#)

또는 디스커스에 가입하세요. [?](#)

이름



**sung wook Kim** · 4달 전

혹시 이렇게 구현하고 sequelize 연동해도 문제가 없을 까요?  
nuxt의 webpack server bundling이 불안하게 느껴집니다.

[^](#) | [v](#) · [답글](#) · [공유](#)



**rhostem** 관리자 → **sung wook Kim** · 3달 전

컨셉 수준으로 구현한 것이고 이걸 바탕으로 실제로 서비스를 운영해보진 않아서 장담은 못드릴 것 같습니다. 😊

[^](#) | [v](#) · [답글](#) · [공유](#)

[구독](#)

[당신의 사이트에 Disqus 추가하기](#) [Disqus 추가하기](#) [Do Not Sell My Data](#)



Copyright © [rhostem](#) All rights reserved

powered by [Gatsby](#), [Contentful](#), [CircleCI](#)