

1 Nuxt.js
2 ref : <https://vue-nuxt.gitbook.io/nuxt/>
3
4 1. Nuxt.js
5 1)Vue.js framework를 기반으로 SSR(Server Side Rendering) web page를 만들 수 있도록 해 주는 library이다.
6
7 -SSR : Vue Application을 Server에서 실행하고 Rendering된 HTML 문서를 Client로 넘겨 주는 기술
8
9 2)Vue.js Application 개발을 보다 강력하고 사용하기 쉽게 만들어 주는 Framework이다.
10 3)비동기 데이터, Middleware, Layout 등과 같이 Client와 Server 사이에서 개발하는데 도움이 되는 많은 기능을 제공한다.
11 4)SEO 등의 문제로 CSR이 아닌 SSR 웹을 구축해야 하는 경우에 유용하게 사용할 수 있다.
12 5)Project 전반적인 folder 구조가 조금 다르지만 대체로 비슷하게 동작한다.
13 6)Router setup이 달라진다.
14 7)Directory 구조에 따라 자동생성하는 것을 default로 한다.
15 8)개발 및 배포 환경이 조금 달라진다.
16 9)Layout을 component의 property로써 사용할 수 있다.
17
18 10)구조적 차이
19 -npx로 생성한 Nuxt project와 vue-cli3으로 생성한 Vue.js project의 구조를 비교해 보자.
20 -Nuxt의 경우에는 Universal(non-SPA) 앱이면서 vuex를 사용하도록 설정되어 있다.
21 -반면 Vue.js의 경우에는 vue-router와 vuex를 사용하도록 설정해야 한다.
22 -전반적으로 Vue.js project의 src folder 아래에 있던 code들이 root level로 올라왔다.
23 -이름이 그대로 유지되는 directory는 기능 또한 그대로 유지되는 것으로 보인다.(Asset, Component, Store 등)
24 -router의 경우 Vue.js에서는 선택적으로 사용하는 기능인 반면 nuxt에는 항상 포함되어 있는 기능으로 변화하였다.
25 -vue-cli(@vue/cli)로 project를 생성하면 router의 사용 여부를 묻고 사용하는 것으로 선택한 경우 router 및 views folder가 생성된다.
26 -반면, npx로 Nuxt project를 생성하는 경우에는 pages folder가 항상 존재하게 되며, router와 views 폴더를 대체한다.
27 -이 때, 기존에 router/index.js 에 직접 router를 등록해 주던 것과 달리 Nuxt에서는 build할 때 자동으로 pages folder의 구조대로 router를 생성해 준다.
28 -Project를 build할 때 dist folder 안에 생성되던 file들은 .nuxt 안으로 이동했다.
29 -물론 CSR이 아닌 SSR이 되었기에 build 결과물의 형태는 조금 달라졌다.
30
31 11)라우팅 방식의 차이
32 -가장 큰 변화중의 하나로써, router를 꼽을 수 있을 것 같다.
33 -기존의 Vue.js project에서는 router를 이용하여 직접 component를 path에 binding하고 fallback 등을 설정할 수 있었다.
34 -Nuxt에서는 views folder 안에 있는 .vue file들을 folder 내 구조에 맞게 자동으로 router에 추가 해 준다.
35
36 12)Nuxt.js로 서비스 가능한 유형
37 -Server Side Rendering(SSR, MPA)
38 --Server와의 연동이 필요한 서비스 개발
39
40 -Client Side Rendering(CSR, SPA)
41 --Server 연동 없이 1개의 Web page에서 실행되는 서비스 개발
42
43 -정적 Resource Generator(Static Generate)
44 --Server 연동 없이 다수의 정적 page로 실행되는 서비스 개발
45
46
47 2. 설치
48 1)create-nuxt-app 사용하기
49 -Nuxt.js 팀은 빠른 시작을 위해 create-nuxt-app라는 툴을 만들었다.
50 -우선 npx가 설치되어 있는 지를 확인한다.
51 -NPM 5.2.0부터 npx는 기본적으로 설치되어 있다.
52 -npx create-nuxt-app <project-name>
53
54 \$ npx create-nuxt-app my-test
55
56 -이 명령어는 아래의 질문을 묻는다.
57 --통합 서버 사이드 프레임 워크를 선택 : None (Nuxt default server)
58 --UI 프레임워크 선택:None (차후에 언제든 추가 가능)
59 --테스팅 프레임워크 선택:None (차후에 언제든 추가 가능)
60 --넥스트 모드(Universal 혹은 SPA) 선택.
61 --HTTP 통신을 위한 axios module 추가 여부.
62 --저장 시 코드 린트를 위한 ESLint 추가 여부.

--코드를 prettify하기 위한 Prettier 추가 여부.

create-nuxt-app v2.15.0

🌀 Generating Nuxt.js project in my-test

? Project name my-test

? Project description My bedazzling Nuxt.js project

? Author name

? Choose programming language JavaScript

? Choose the package manager Npm

? Choose UI framework None

? Choose custom server framework None (Recommended)

? Choose Nuxt.js modules (Press <space> to select, <a> to toggle all, <i> to invert selection)

? Choose linting tools (Press <space> to select, <a> to toggle all, <i> to invert selection)

? Choose test framework None

? Choose rendering mode Universal (SSR)

? Choose development tools (Press <space> to select, <a> to toggle all, <i> > to invert selection)

```

      :-:
      .==--+
      .==. :+- .-=-
      .==. :==++-+=.
      :==.  -*: :+=.
      :+-   :*+++. .++.
      :+-   -= .++: .+=.
      -+:   =*- .+*: .+=:
      -+:   .=-* .=-* =+:
      .==:   .+*:   -*- -+-
      .+=:.....+*-.....:=*..=-*
      .-=-=-=-=+++=-=-=-=-=-+++=-=-=:

```

Thanks for installing nuxtjs

💡 Successfully created project my-test

To get started:

```
cd my-test
npm run dev
```

To build & start for production:

```
cd my-test
npm run build
npm run start
```

-선택을 했다면, 모든 디펜던시들은 설치가 될 것이다.

-다음 단계는 프로젝트 폴더로 이동해 다음 명령어로 실행하는 것이다.

```
$ cd my-test
$ npm run dev
```

> my-test@1.0.0 dev D:\VueHome\my-test

> nuxt

```

      |
Nuxt.js v2.12.2      |
Running in development mode (universal) |
      |
Listening on: http://localhost:3000/    |
      |

```

```
130
131 i Preparing project for development
132 i Initial build may take a while
133 ✓ Builder initialized
134 ✓ Nuxt files generated
135
136 ✓ Client
137   Compiled successfully in 4.53s
138
139 ✓ Server
140   Compiled successfully in 4.49s
141
142 i Waiting for file changes
143 i Memory usage: 159 MB (RSS: 232 MB)
144 i Listening on: http://localhost:3000/
```

-이제 애플리케이션은 다음 주소에서 작동한다.

<http://localhost:3000>

-Nuxt.js는 page 디렉토리의 파일 변경을 지켜보고 있으므로, 새로운 페이지를 추가할 때 마다 애플리케이션을 다시 시작할 필요가 없다.

2)vue-cli를 통한 Nuxt.js 설치

-Nuxt.js 버전이 올라가기 전에는 nuxt starter-template을 vue-cli를 통해서 설치했다.

-CLI가 아닌 ZIP 파일로 받아 압축을 풀어도 된다.

-먼저 vue-cli를 설치한다.

```
$ npm i -g @vue/cli-init
```

-그다음으로 template를 설치하고 npm module을 설치하면 된다.

```
$ vue init nuxt-community/starter-template <project-name>
```

? Project name mysite

? Project description Nuxt.js project

? Author

vue-cli · Generated "mysite".

To get started:

```
cd mysite
```

```
npm install # Or yarn
```

```
npm run dev
```

```
$ cd <project-name>
```

```
$ npm i
```

3)처음부터 Nuxt 시작하기

-위의 create-nuxt-app 명령어를 사용하지 않고

-맨 처음부터 Nuxt.js 프로젝트를 생성하는 것은 쉽다.

-단지 파일 하나, 디렉토리 하나가 필요하다.

-우선 빈 디렉토리를 생성한다.

-mkdir <project-name>

```
$ mkdir mysite
```

```
$ cd mysite
```

-package.json

--nuxt를 시작하기 위해선 모든 프로젝트는 package.json가 필요하다.

--이 json 파일을 package.json에 복사해 붙여넣기하고 npm install을 실행하기 전에 저장한다.

```
{
```

```
"name": "mysite",
"scripts": {
  "dev": "nuxt"
}
```

--뒤에서 npm run dev를 실행하면 scripts가 Nuxt.js를 실행할 것이다.

-nuxt 설치하기

--package.json가 생성되었다면, npm으로 nuxt를 설치한다.

```
$ npm install --save nuxt
```

-pages 디렉토리

--pages 디렉토리에 있는 *.vue 파일들을 Nuxt.js는 애플리케이션의 라우트(route)로 사용할 것이다.

--pages 디렉토리 생성하기

```
$ mkdir pages
$ cd pages
```

--pages/index.vue에서 첫 번째 페이지를 생성한다.

```
<template>
  <h1>Hello world!</h1>
</template>
```

-그리고 프로젝트를 아래의 명령어로 실행한다.

```
$ cd ..
$ npm run dev
```

```
> mysite@ dev D:\VueHome\mysite
> nuxt
```

```
Nuxt.js v2.12.2
Running in development mode (universal)
Listening on: http://localhost:3000/
```

i Preparing project for development

i Initial build may take a while

✓ Builder initialized

✓ Nuxt files generated

✓ Client

Compiled successfully in 4.03s

✓ Server

Compiled successfully in 3.68s

i Waiting for file changes

i Memory usage: 171 MB (RSS: 235 MB)

i Listening on: http://localhost:3000/

-이제 애플리케이션은 다음 주소에서 작동한다.

<http://localhost:3000>

3. Folder 구조

-ref : <https://sites.google.com/site/yongheuicho/class/cloud/nuxtjssayongbangbeob>

1)pages: Vue file 저장

-index.vue가 기본 표시 page
-각종 routing은 해당 Vue file로 지향
--예를 들면 `http://localhost/about`은 `about.vue`를 표시함
--`<nuxt-link to="/about">About</nuxt-link>`

2)static: 고정적인 그림, 문서 등을 저장
-예를 들어 이 folder는 `'~static/img.png'` 혹은 `'/img.png'`로 접근함

3)assets: webpack으로 처리할 asset을 저장
-이 folder는 webpack에 의해 처리됨
-Webpack: JS module(C 언어에서 #include) 지원 system
-예를 들어 `"~assets/img.png"`로 접근
-`'~'`는 webpack으로 처리하여 source directory를 의미(~ 대신 @ 사용 가능)
-`'~~'`는 webpack으로 처리하여 root directory를 의미(~~ 대신 @@ 사용 가능)
-설정을 바꾸지 않으면 ~와 ~~는 동일

4)layouts: layout에 관련된 file 저장
-Main layout은 `default.vue`로 생성
-Nuxt.js가 생성하는 runtime page가 `<nuxt/>` 부분에 표시 예정
-`<nuxt/>` 위에 있는 내용이 header, 아래에 있으면 footer

5)components: Vue.js component 저장

6)middleware: middleware에 사용할 사용자 정의 함수 저장

7)plugins: Vue.js application이 생성되기 전에 실행되는 JavaScript plugin을 저장
-`nuxt.config.js`에 등록해야 함
-일반 예: `plugins: [{src: '~/plugins/myFun'}]`
--*.vue에서 사용할 때: `import myFun from '~/plugins/myFun';`
-SSR(server-side rendering) 하지 않는 예: `plugins: [{src: '~/plugins/myChart', ssr: false}]`

8)store: Vuex Store file 저장
-Vuex: state management pattern + library for Vue.js applications(중앙 집중형 상태 관리 library)
-Vuex Store: Vuex가 만드는 state, mutation 정보를 중앙 관리하는 핵심 object

9)자체 link 설정 방법
-HTML에서: `~assets/img.png` 등으로 설정; `nuxt-link` 사용
-Script에서: `~/components/compo.vue` 등으로 설정

4. JSON 처리 방법

1)Server에 있는 JSON은 import 이용해 string 형태로 읽어들이

2)test.json 읽는 방법

`import strTest from '~/assets/test.json'`

3)읽은 strTest를 parsing

`let objTest = JSON.parse(strTest);`

4)다른 server에 있는 JSON은 `axios.get()` 이용해 object 형태로 읽어들이

5. Hello, World

1)이번 예제를 통해 Nuxt와 Vue가 어떻게 rendering을 분담하는 지 이해하도록 한다.

2)이번 예제를 통해서 현재 페이지가 Nuxt에서 rendering되는지, 아니면 Client에서 rendering되는지 확인하는 예제이다.

3)Project 생성

`$ npx create-nuxt-app my-project`

...

`$ cd my-project`

`$ npm run dev`

`-http://localhost:3000`

4)/pages/index.vue

```

330 <template>
331   <div class="container">
332     <div>
333       <logo />
334       <h1 class="title">
335         Vue? Nuxt?
336       </h1>
337       <h2 class="subtitle">
338         Which one rendered this page?
339       </h2>
340       <div class="links">
341         <nuxt-link to="guess" class="button--green">Guess!</nuxt-link>
342       </div>
343     </div>
344   </div>
345 </template>
346
347 <script>
348 import Logo from '~/components/Logo.vue'
349
350 export default {
351   components: {
352     Logo
353   }
354 }
355 </script>
356
357 <style scoped>
358 .container {
359   margin: 0 auto;
360   min-height: 100vh;
361   display: flex;
362   justify-content: center;
363   align-items: center;
364   text-align: center;
365 }
366
367 .title {
368   font-family: 'Quicksand', 'Source Sans Pro', -apple-system, BlinkMacSystemFont,
369     'Segoe UI', Roboto, 'Helvetica Neue', Arial, sans-serif;
370   display: block;
371   font-weight: 300;
372   font-size: 100px;
373   color: #35495e;
374   letter-spacing: 1px;
375 }
376
377 .subtitle {
378   font-weight: 300;
379   font-size: 42px;
380   color: #526488;
381   word-spacing: 5px;
382   padding-bottom: 15px;
383 }
384
385 .links {
386   padding-top: 15px;
387 }
388 </style>
389

```

5)pages/guess.vue

```

391 <template>
392   <component :is="target"></component>
393 </template>
394
395 <script>

```

```

397 import NuxtPage from '~/components/nuxt.vue'
398 import VuePage from '~/components/vue.vue'
399
400 export default {
401   components : {
402     Nuxt: NuxtPage,
403     Vue: VuePage
404   },
405   asyncData({ req }) {
406     return {
407       target : (req) ? 'Nuxt' : 'Vue'
408     }
409   }
410 }
411 </script>
412
413 6)/components/nuxt.vue
414 <template>
415   <section class="container">
416     <div>
417       <logo />
418       <h1 class="title">
419         NUXT
420       </h1>
421       <h2 class="subtitle">
422         Universal Vue.js Application
423       </h2>
424       <div class="links">
425         <a href="https://nuxtjs.org/" target="_blank"
426           class="button--green">Documentation</a>
427         <a href="https://github.com/nuxt/nuxt.js" target="_blank"
428           class="button--gray">Github</a>
429       </div>
430     </div>
431   </section>
432 </template>
433 <script>
434 import Logo from '~/components/Logo.vue'
435
436 export default {
437   components:{
438     Logo
439   }
440 }
441 </script>
442 <style scoped>
443 .container {
444   margin: 0 auto;
445   min-height: 100vh;
446   display: flex;
447   justify-content: center;
448   align-items: center;
449   text-align: center;
450 }
451 .title {
452   font-family: 'Quicksand', 'Source Sans Pro', -apple-system, BlinkMacSystemFont,
453     'Segoe UI', Roboto, 'Helvetica Neue', Arial, sans-serif;
454   display: block;
455   font-weight: 300;
456   font-size: 100px;
457   color: #35495e;
458   letter-spacing: 1px;
459 }
460 .subtitle {
461   font-weight: 300;

```

```

462     font-size: 42px;
463     color: #526488;
464     word-spacing: 5px;
465     padding-bottom: 15px;
466 }
467
468 .links {
469     padding-top: 15px;
470 }
471 </style>
472
473 7)/components/vue.vue
474 <template>
475     <div class="container">
476         
477         <div class="hello">
478             <h1>{{ msg }}</h1>
479             <h2>Essential Links</h2>
480             <ul>
481                 <li><a href="https://vuejs.org" target="_blank">Core Docs</a></li>
482                 <li><a href="https://forum.vuejs.org" target="_blank">Forum</a></li>
483                 <li><a href="https://gitter.im/vuejs/vue" target="_blank">GitterChat</a></li>
484                 <li><a href="https://twitter.com/vuejs" target="_blank">Twitter</a></li>
485                 <br />
486                 <li><a href="https://vuejs-templates.github.io/webpack/" target="_blank">Docs for
This Template</a></li>
487                 <li><a href="http://router.vuejs.org" target="_blank">vurouter</a></li>
488                 <li><a href="http://vuex.vuejs.org" target="_blank">vuex</a></li>
489                 <li><a href="http://vue-loader.vuejs.org" target="_blank">vueloader</a></li>
490                 <li><a href="https://github.com/vuejs/awesome-vue"
target="_blank">awesome-vue</a></li>
491             </ul>
492         </div>
493     </div>
494 </template>
495 <script>
496 export default {
497     data(){
498         return{
499             msg: 'Welcome to Your Vue.js App'
500         }
501     }
502 }
503 </script>
504 <!--Add 'scope' attribute to limit CSS to this componen only -->
505 <style scoped>
506     .container{
507         font-family: 'Avenir', Helvetica, Arial, sans-serif;
508         -webkit-font-smoothing:antialiased;
509         -moz-osx-font-smoothing:grayscale;
510         text-align: center;
511         color: #2c3e50;
512         margin-top:60px;
513     }
514
515     h1, h2{
516         font-weight: normal;
517     }
518
519     ul{
520         list-style-type: none;
521         padding:0;
522     }
523
524     li {
525         display: line-block;
526 
```



```
527     margin:0 10px;
528   }
529
530   a {
531     color: #42b983;
532   }
533
534   img {
535     width:200px;
536     height:200px;
537   }
538 </style>
```

539
540 8)모두 저장 후 다음 명령을 실행한 다음, Browser에서 확인한다.

```
541
542 $ npm run dev
543
```

544 9)예제를 실행한 후, Web page에서 보면 Guess 버튼이 보인다.

545 10)이 button을 클릭하여 이동한 페이지가 만일 Nuxt로 rendering되었다면 Nuxt logo를, Vue로 rendering되었다면 Vue logo를 보여준다.

546 11)즉, Guess 버튼을 클릭해서 이동한 페이지가 vue이면 Client에서 실행된 것이다.

547 12)Nuxt logo 즉, 직접 localhost:3000/guess로 입력해서 들어가거나 새로 고침하면 Nuxt logo가 나올 것이다.

548 13)Nuxt는 처음 요청에 대해서만 rendering된 page를 전달한다.

549 14)Single Page Application이기 때문에 다른 page들은 처음 page의 resource loading 때 대부분이 download되어 나머지는 Client에서 rendering 된다.

550 15)그래서 이 예제에서 localhost:3000으로 접속하고 Guess 버튼을 누르면 이 action은 client 영역이 되는 것이고, localhost:3000/guess로 접속한다면 이것은 Nuxt에서 rendering해서 보내주게 되는 것이다.

551 552 553 6. 정적으로 Page build하기

554 1)정적 Page build는 Nuxt application을 미리 build해서 HTML 파일로 만들어 주는 기능이다.

555 2)다음 명령어를 사용한다.

```
556
557 mysite\node_modules\.bin\nuxt generate
558
```

559 3)만일 nuxt 명령어가 실행되지 않으면 위와 같이 nuxt 경로를 직접 입력해서 실행한다.

560 4)Build가 완료되면 /dist folder에 HTML 파일들이 생성되는 것을 볼 수 있다.

561 5)이 예제를 npm run dev로 build하면 nuxt logo를 볼 수 없다.

562 6)왜냐하면, build 된 결과물은 이미 rendering된 결과물이기 때문에 Server(Nuxt)의 영향 없이 오직 Client의 영향만 있기 때문이다.

563 564 565 7. 예제 실행하기

566 1)<https://ko.nuxtjs.org/examples>에서 hello-world.zip을 download 한다.

567 2)vuehome folder에 압축을 푼다.

568 3)다음의 명령을 수행한다.

```
569
570 $ vuehome/hello-world> npm install --save nuxt
571
```

572 4)그리고 \$ npm run dev로 build한다.

573
574 5)Web Browser에서 확인한다.