

1. todo-html

- html 파일에 vue.js 코드를 작성하는 방식
- 프로그램 종료시, todo 데이터가 사라지는 단점
- 브라우저 Storage 를 사용하면, todo 데이터가 유지되지만,
- 내 노트북에서만 보이고, 다른 노트북에서는 볼 수 없음

2. todo-vue-only

- 장고에 (1.번)의 html 파일을 그대로 수용하는 방식
- 테이블을 사용하지 않아, todo 데이터 관련 단점은 (1.번)과 동일함

3. todo-django-only

- Vue.js 코드 없이, 장고 코드만으로 todo 앱을 개발하는 방식
- 테이블을 사용하므로, 다른 사람과 데이터 공유가 가능함
- 간단하고 쉬운, 클래스형 뷰를 사용함

4. todo-django-mixin

5. todo-vue-django

6. blog-vue-django

My Todo App !

서로 할 일이나 의견을 공유해 봅시다.

CreateView

name ..

type anything welcomed ...

ADD

김석훈:: Django 와 Vue.js 연동 프로그램을 만들고 있습니다.

×

홍길동:: 이름을 안쓰면 홍길동으로 나와요...

×

이순신:: 신에게는 아직 열두 척의 배가 있습니다.

×

성춘향:: 그네 타기

×

ListView

DeleteView

- ListView: DB 에서 레코드 목록을 가져와 보여주는 뷰
- CreateView: 폼에 입력한 내용으로, DB 에 레코드를 생성하는 뷰
- UpdateView: DB 에 있는 특정 레코드를 수정하는 뷰
- DeleteView: DB 에 있는 특정 레코드를 삭제하는 뷰

1. todo-html

- html 파일에 vue.js 코드를 작성하는 방식
- todo 데이터가 사라지는 단점이 있어, 불완전한 웹 프로그램 임

2. todo-vue-only

- 장고에 (1.번)의 html 파일을 그대로 수용하는 방식
- todo 데이터가 사라지는 단점이 있어, 불완전한 웹 프로그램 임

3. todo-django-only

- Vue.js 코드 없이, 장고 코드만으로 todo 앱을 개발하는 방식
- 완전한 웹 프로그램, 페이지 이동 자주 발생 (3개 html)

4. todo-django-mixin

- 장고 클래스형 뷰의 믹스인 기능을 활용하는 방식
- 페이지 이동을 축소 함

Client rendering

Server rendering

1. todo-html
 - html 파일에 vue.js 코드를 작성하는 방식
 - todo 데이터가 사라지는 단점이 있어, 불완전한 웹 프로그램 임
2. todo-vue-only
 - 장고에 (1.번)의 html 파일을 그대로 수용하는 방식
 - todo 데이터가 사라지는 단점이 있어, 불완전한 웹 프로그램 임
3. todo-django-only
 - Vue.js 코드 없이, 장고 코드만으로 todo 앱을 개발하는 방식
 - 완전한 웹 프로그램, 페이지 이동 자주 발생 (3개 html)
4. todo-django-mixin
 - 장고 클래스형 뷰의 믹스인 기능을 활용하는 방식
 - 페이지 이동을 축소 함

Client Rendering	Server Rendering
Vue.js 코드가, 즉 클라이언트에서 렌더링을 한다.	Django 코드가, 즉 서버에서 렌더링을 한다.
<p>Rendering 이란 ?</p> <ul style="list-style-type: none"> • DOM + CSSOM = Rendering Tree • Layout • Painting <p>이런 Rendering 은 항상 브라우저, 즉 클라이언트에서 처리되는 과정 임.</p>	
➔ DOM/CSSOM 에 필요한 HTML 및 CSS 를, 누가 생성하느냐 ?	
Vue.js 코드에서, <ul style="list-style-type: none"> • DOM 용: html tag 및 data • CSSOM 용: css 를 생성 함 	Django 코드에서, <ul style="list-style-type: none"> • DOM 용: html tag 및 data • CSSOM 용: css 를 생성 함

Client Rendering

- 첫 화면은 Server Rendering (Django 에서 HTML/CSS 를 생성)
- 그 이후에 Client Rendering (Vue.js에서 HTML/CSS 다시 생성)

첫 화면을 어떻게 ?

빈 화면	내용이 있는 화면
<ul style="list-style-type: none">• Vue CLI (webpack)• Vue directive, axios• Vue Router, Vuex• Vue.js 의 일반적인 개발 방식• SPA(Single Page Application) 방식	<p>Nuxt.js F/W</p> <ul style="list-style-type: none">• SSR(Server Side Rendering) 방식• SEO 문제 해결 가능• Client + Server Rendering 방식으로, Vue.js 애플리케이션을 개발할 수 있음• Client/Server 코드 모두 Vue.js
<p>SEO 문제</p> <ul style="list-style-type: none">• Search Engine Optimization• 검색엔진에 노출되지 않아, 페이지가 홍보가 안되는 문제	<p>우리 예제 (VueOnly 메뉴)</p> <ul style="list-style-type: none">• 첫 화면은 Django Rendering• 이후는 Vue.js Rendering• SEO 문제 해결 가능 <p>→ Nuxt.js 방식과 유사함</p>

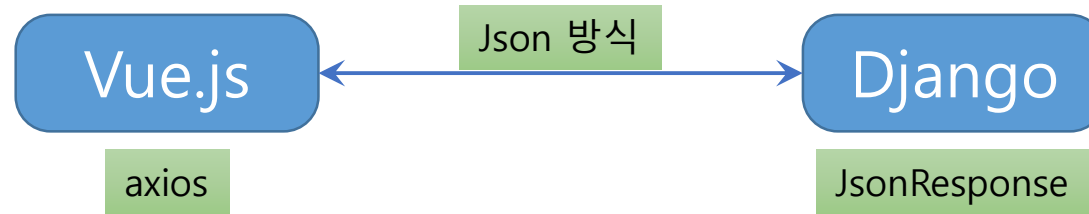
▶ 장고 개발자의 선택은 ?

	Client	Server	
1 안	Django	Django	DjangoMixin 메뉴
2 안	Vue.js	Django	VueOnly 메뉴
3 안	Nuxt.js	Nuxt.js	모두 Vue.js 코드

▶ 필자의 선택은

- Client 는 Client 전문, Server는 Server 전문 프레임워크
- Vue.js 는 React/Angular 등 다른 프레임워크 대비 쉬움
- Vue.js 는 기존 코드에 적용이 쉽고, 점진적으로 적용이 가능함

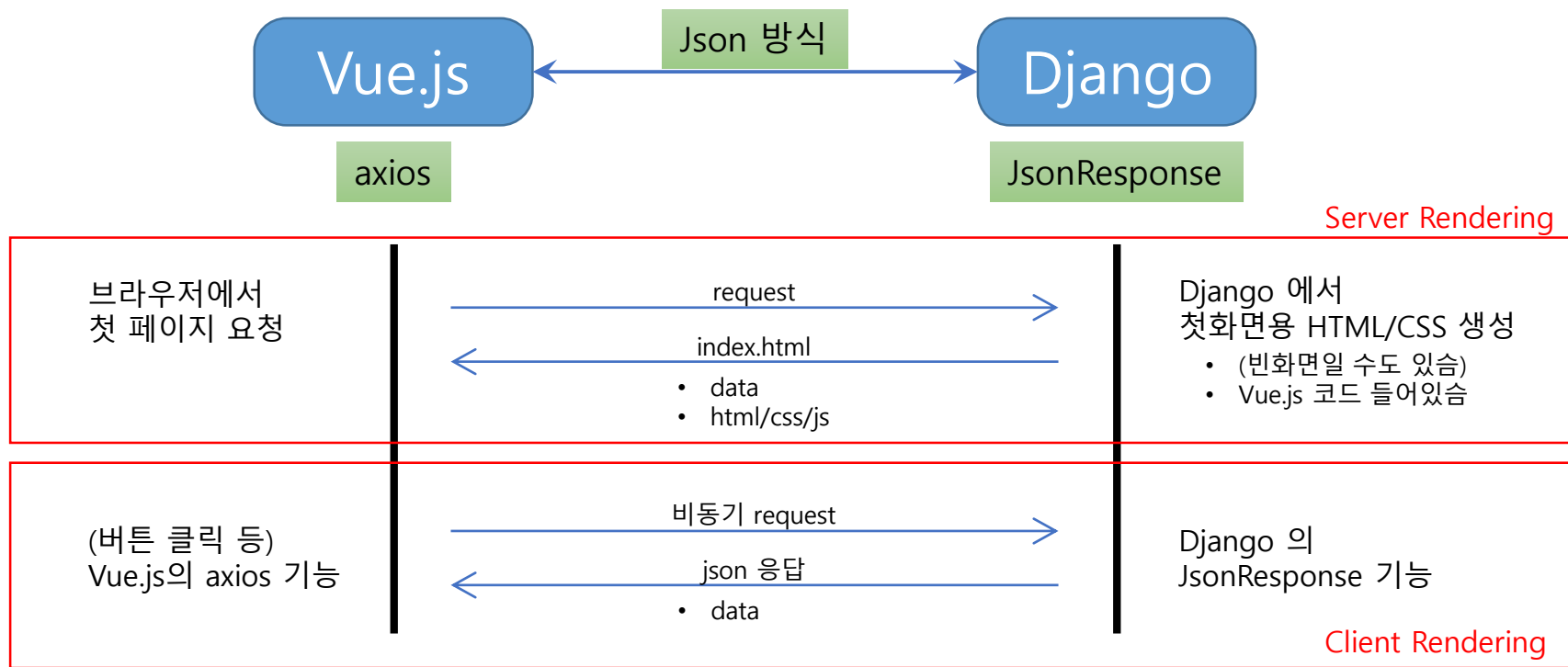
▶ Todo 앱 설계 (Vue-Django 연동 방식) - 1



VueOnly 메뉴

- ① 첫 페이지는 Django 에서 생성해서, 클라이언트에게 보내 줌
- ② 이후 화면 렌더링은, Vue.js 코드에서 수행
- ③ Data 저장은 서버 측의 DB 에 저장 (SQLite)
- ④ Client-Server 간 Data 연동은 JSON 포맷으로
- ⑤ Vue.js 의 directive/axios 기능 사용
- ⑥ DRF(Django Rest Framework) 대신에, JsonResponse 사용

▶ Todo 앱 설계 (Vue-Django 연동 방식) - 2



Virtual DOM 처리로, 화면 깜박임 없음

Data만, 트래픽 감소로 응답속도 빠름

1. todo-html

2. todo-vue-only

3. todo-django-only

4. todo-django-mixin

5. todo-vue-django

- Vue.js - Django 연동 방식 (JSON 포맷)
- VueOnly 소스에, 데이터 처리를 위한 JSON 연동기능 추가
- VueOnly + axios + JsonResponse

▶ Vue-Django 연동 처리 과정 (Json 방식)

