

| 기타 강좌 | >

Vercel로 빠르게 서버리스 API를 배포해 보자 (1) - 개요



주노 챗봇 중수 🍌 1:1 채팅

2020.07.25. 18:06 조회 131

💬 댓글 14 URL 복사 ⋮

안녕하세요 주노입니다 ☺

카카오톡 봇을 만들다 보면 데이터를 다른 곳에서 끌어다 와서 가공해야 한다던가(ex> 게임 전적 뽑아내기, 시험 본 다음 날의 한강 수온 구하기 등등 크롤링 및 API 접근), 디바이스 외부의 데이터베이스(ex> 캐싱, 원격 로깅, 학습 등)에 접근해야 한다던가 하는 일들이 많은데요. API 서버가 필요한 순간들이죠.

PHP로 만들면 좀 뭔가 낱고럽긴 하지만 따로 설정하고 뭐하고 할 거 없이 빠르게 배포가 가능하니까 많이 쓰시는 것도 같고요.

에이 나는 자스나 파이썬으로 하고 싶다 하시구 익스프레스나 플라스크 등등으로 따로 서버를 만드셔서 EC2나 로컬에 돌려 쓰시는 분들이 계실 거예요.

하지만 뭔가 불안하지 않습니까...!! 언제 뭔가 꺼질지도 모르고, 에러 핸들링이 제대로 작동하는지도 모르니까 걱정이 조금 들더라고요.

그게 아니더라도 초기에 인스턴스를 만들고 보안 그룹 만지고 CI/CD로 자동 배포하고 등등 여러가지 설정하기가 귀찮아질 겁니다.

그래서 저는 간단한 서버를 만들 때는 Vercel이라는 녀석을 사용하는데요!

한번 간단하게 알아보겠습니다.



짠 요녀석입니다. 애네 이름이 원래는 ZEIT이었는데 몇달 전에 Vercel로 이름을 바꾸더라고요.

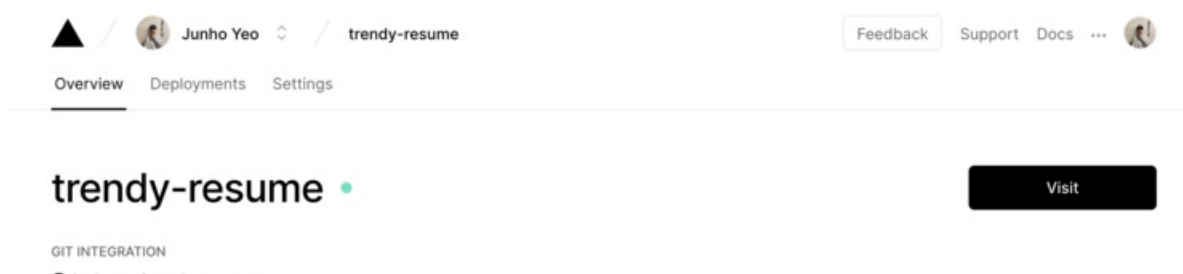
혹시 리액트를 사용하시는 분이 계시다면 한번쯤은 들어보셨을지도 모르는 Next.js, useSWR 같은 애들도 여기서 만든 것이랍니다.

애네의 사명은 'We enable developers to build and publish wonderful things(개발자 친구들이 멋진 거 만들고 배포하게 도와줌)'이고, Vercel(ZEIT now)의 소개는 'Develop. Preview. Ship. For the best frontend teams' 입니다.

약간 아.. 저건 프엔하는 애들 위한 거구나,, 하는 느낌이 드시죠?

실제로 그렇습니다..

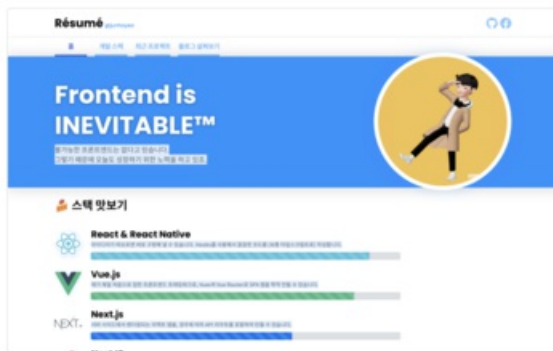
빠르게 리액트 등의 프론트엔드 프로젝트를 배포할 수 있게 시작된 게 바로 Vercel입니다. 영타치기 생각보다 힘들어서 이하 버셀이라고 부르겠습니다. 근데 구글링할 땐 버셀이라고 치지 마세요.. 잘 안나와요,,



Production Deployment

[View Build Logs](#)
[View Function Logs](#)
[View Domains](#)


The deployment that is available to your visitors.



DEPLOYMENT

trendy-resume-ety783jbu.vercel.app

DOMAINS

trendy-resume.junhoyeo.now.sh trendy-resume.now.sh
trendy-resume.junhoyeo.vercel.app

STATE

Ready

CREATED

4m ago

BRANCH

-

To update your Production Deployment, [push to the default branch](#).

메인 기능은 하나! 깃허브에 있는 프로젝트를 연동하면 알아서 어떤 프레임워크로 만들어졌는지, 빌드 커맨드는 뭔지 프로젝트 설정을 읽고 그걸로 바로 배포해주는 것이죠. **now.sh**로 끝나는 이쁜 도메인도 줍니다(요즘은 **vercel.app**을 선호하는 것 같음,, 물론 now.sh가 더 끌리면 그걸로 바꾸거나 커스텀 도메인으로 설정할 수도 있습니다).

Preview Deployments

Deployments that are currently being worked on.

DEPLOYMENT	STATE	BRANCH	AGE	CREATOR
trendy-resume-git-dependabot-np...	Ready	dependabot/npm_and_yarn/lodash-... chore(deps): bump lodash from 4.17...	6d ago	
trendy-resume-git-new-data-temp...	Ready	new-data-templates Merge branch 'master' of https://git...	80d ago	
trendy-resume-gbec7kt4h.now.sh	Ready	new-data-templates Merge branch 'master' of https://git...	80d ago	
trendy-resume-h7q3h4mlo.now.sh	Ready	new-data-templates Merge branch 'master' of https://git...	81d ago	
trendy-resume-nt1j0hz44.now.sh	Ready	new-data-templates refactor: render header information ...	166d ago	
trendy-resume-ol03qz0t7.now.sh	Ready	imgbot [imgBot] Optimize images *Total -- ...	169d ago	
trendy-resume-9nu4m4r2i.now.sh	Ready	imgbot [imgBot] Optimize images *Total -- ...	178d ago	
trendy-resume-59a65lpnj.now.sh	Ready	imgbot [imgBot] Optimize images *Total -- ...	179d ago	
trendy-resume-98nrkbgvj.now.sh	Ready	imgbot [imgBot] Optimize images *Total -- ...	179d ago	

그리고 다른 브랜치나 최근 커밋에 대해서는 이렇게 따로 배포해서 링크를 만들어 주기 때문에 다른 사람들과 작업하거나, 누가 PR을 올렸을 때 따로 체크아웃해서 돌려볼 필요 없이 바로 미리 확인해 볼 수 있습니다.

이쯤 되면 **우와 개꿀네 근대 이걸로 어떻게 서버를 만든다는 거야?** 하실 수도 있는데요.
버셀에서는 (프엔 개발자들을 위해서) **서버리스 API**를 만들 수 있게 해줍니다.
서버리스(Serverless), 이름 그대로 서버가 없따...?? 서버 없는 서버!!!

서버가 없으면 뭐를 짤까...요? 그래서 함수(Serverless Functions)라는 단위가 있습니다.
서버 아키텍처 전체를 만들어서 올리는 대신, 각각의 요청에 대한 처리 함수만 만들어 올리는 것이죠.
간단한 예를 들어보기 위해서 플라스크 문서에 있는 헬로월드 예제를 가져와 볼게요.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

썬,, 물론 현실 세계의 앱을 보면 저 위에 다른 미들웨어도 등록하고, 라우트 주소에 따라 이것저것 설정이 들어가겠지만요,,
어쨌든 다른 코드랑 무관하게 실제로 **실행에 바로 관여되는 코드**는 저 hello_world 함수 뿐입니다.
구조니 뭐니 신경쓰지 않고, 저렇게 라우팅 함수만을 올려서 **쓸데없는 걸 줄이고 서비스의 구현에만 집중할 수 있게 해주는 것이 바로 서버리스 아키텍처**인 것이죠!!

그래서 버셀에서도, 프론트엔드 개발과 서비스 구현에 집중해야 하는 프엔 개발자들이 머리 싸매고 백엔드를 다루고 있는 것이 보기 애
쳐로웠는지 이를 지원하고 있습니다. 물론 **AWS나 Azure 같은 상용 클라우드 플랫폼에서도 각각 Lambda, Functions**라는 이름으로 서버
리스 평션을 올릴 수 있게 해줍니다.

Creating Serverless Functions

To create a Serverless Function, create a file in an `/api` directory from your project root with the appropriate language extension.

NOTE: If you are using Next.js, use the `/pages/api` directory instead.
[Read more about API functionality with Next.js.](#)

For example, to create a TypeScript function, you can create a file called `get-user.ts` like below.

```
import { NextRequest, NextResponse } from '@vercel/node'

export default (req: NextRequest, res: NextResponse) => {
  res.json({ name: 'John', email: 'john@example.com' })
}
```

A TypeScript Serverless Function that returns a JSON response.

버셀의 다른 프로젝트처럼 역시 매우 간단합니다. 위에서 설명했듯이 **프로젝트의 /api 폴더에 라우팅 경로 이름으로 된 파일을 만들고, 배포하기만 하면 끝!!**

이상 개요를 마치고 다음 강의부터는 아래 것들을 해보겠습니다.

- 간단한 API 만들어서 Vercel로 배포하고 봇에서 끌어다가 써보기
- 타입스크립트 붙이기
- mLab로 새로운 MongoDB 데이터베이스를 만들고, 요청으로 들어온 값 저장하기
- axios랑 cheerio로 재미있는 거 파싱하기

감사합니다 바위~~~ ㄹ

#vercel #서버리스 #서버 #API #백엔



주노님의 게시글 더보기 >



❤️ 좋아요 5 💬 댓글 14

🔗 공유 신고

댓글 등록순 최신순

댓글알림



은댕이

3줄 요약중

2020.07.25. 18:14 답글쓰기



주노 작성자

1. API 서버 만들 때 처음부터 끝까지 구축할 필요 없음
2. Vercel이라는 곳에서 서버리스라는 걸 쉽게 만들도록 해줌
2. 요청 처리하는 함수만 몇개 자스로 짜서 올리면 끝

2020.07.25. 18:19 답글쓰기



은댕이

주노 7x

2020.07.25. 21:05 답글쓰기



주노 작성자

은댕이



2020.07.25. 21:26 답글쓰기

$e^0 + 1$
 $= 0$

Steve28

정보추 남깁니다 bb

2020.07.25. 18:23 답글쓰기



주노 작성자



2020.07.25. 18:28 답글쓰기



Kiri

좋은 정보 감사합니다!

2020.07.25. 18:50 답글쓰기



주노 작성자





성빈

오우 광고인줄



2020.07.25. 19:13 답글쓰기



주노

작성자

악마니팔고싶습니다



2020.07.25. 19:24 답글쓰기



성빈

주노

우왕 대대내



2020.07.25. 19:25 답글쓰기



주노

작성자

성빈



2020.07.25. 19:27 답글쓰기



윤

하트 누르고 갑니다

2020.07.26. 08:33 답글쓰기



주노

작성자

오예 고마어요 ㅁ

2020.07.26. 13:14 답글쓰기

Hibot

댓글을 남겨보세요



등록

'| 기타 강좌 |' 게시판 글

이 게시판 새글 구독하기

봇 개발자에서 앱 개발자로 넘어가보기 1단계, 스크립트 액티비티 사용 [7]


Dark Tornado 2020.07.30.

[스압] 원하는 기능 만드는 법 (날씨, 끝말잇기 등등) 🌤️🔗 [21]

Dark Tornado 2020.07.28.

Vercel로 빠르게 서버리스 API를 배포해 보자 (1) - 개요 [14]

주노 2020.07.25.

봇이 들어가있는 방 목록을 뽑아보자! 무료 사기 X 최신버전 1.5.2 드롭박스 404 바이러스 X  [35]

99TAN 90 2020.07.08.

루팅실패시 벽돌해결법 [18]

iphone 12 2020.07.07.

1 2

전체보기

이 카페 인기글

뭐여 카페가 갑자기 어케돌아가고있는겨냐

서진

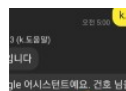
0 2



카톡을 한폰에서 두개쓰는데 답장도 두계정 전부 다 합니다 ππ

다행히도

16



어시스턴트

Ka
Mo ☹️
카페가 터지고 있다

AlphaDo

AlphaDO
♡2 💬12



카드들

정말 좋은거시에오

Lunar

0 2



[팁] 이전 에디터로 돌아가는 법



3333....333333 333333

1 2 3 4 5