

# 백엔드 시스템 입문

## : Rest API 설계 및 배포

KUSITMS 25기 교육기획팀 | 손재정



# 목차



백엔드 시스템 기본 이론



REST API 설계



API 테스트 및 문서 작성



API 서버 배포



*01*

## 백엔드 시스템 기본 이론

백엔드 시스템이란 ? 서버 ? API ?

서버 터졌다 !! ☠

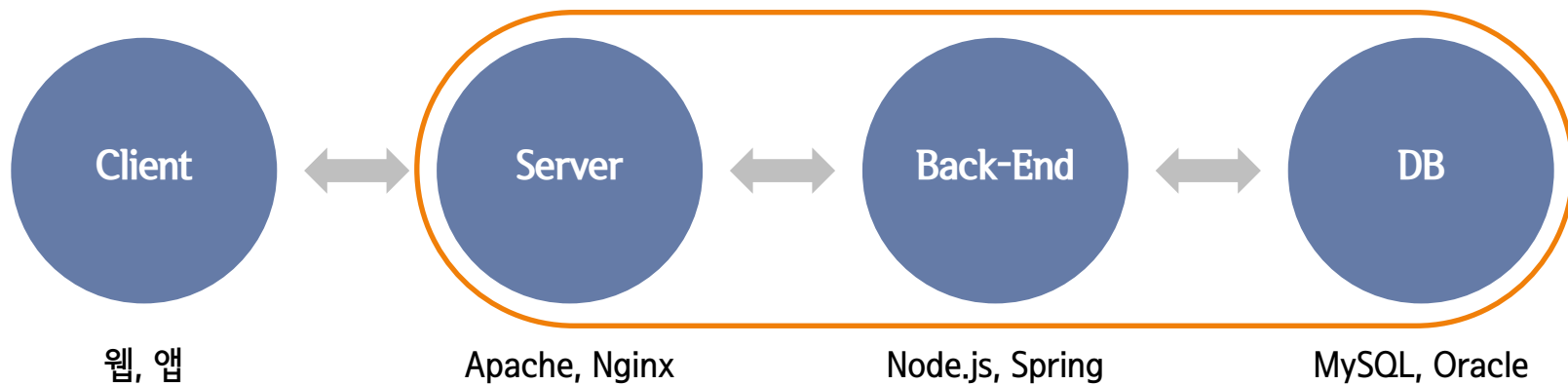


## 서버 (server)

; 클라이언트에게 네트워크를 통해 정보나 서비스를 제공하는  
컴퓨터 시스템으로, 컴퓨터 프로그램 혹은 장치를 의미

= 클라이언트의 요청에 응답 하는 것

## Back-End 시스템



# API (Application Programming Interface)

= 서버 - 클라이언트의 **대화 규칙**



**프론트엔드**  
(Client-side)

GET (가져와줘), POST (전송해줘), DELETE (지워줘)



**백엔드**  
(Server-side)

2xx (ok!), 4xx(요청오류), 5xx(서버오류)

## 실습 진행



Node.js

Javascript 실행환경,  
서버 실행에 주로 사용됨.



Replit. DB

Replit 자체 DB  
(\*Replit : 온라인 IDE 사이트)



Heroku

PaaS 클라우드 서비스,  
무료 호스팅

# 02

## Rest API 설계

HTTP 요청 (GET/POST/DELETE), HTTP Status Code



```
const express = require('express')
const app = express()
```

```
var port = 8080
app.listen(port, () => {
  console.log(`${port} 번 포트에 연결 중 ...`);
})
```

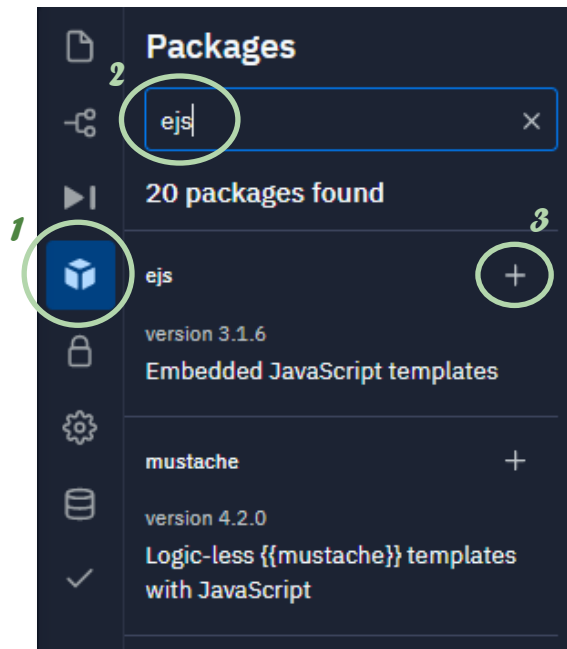
### Express

; Node.js를 사용하여 쉽게 서버를 구성할 수 있도록 해주는 라이브러리

### 포트

; 컴퓨터가 접속하는 장소

; 포트 번호 = 컴퓨터가 어느 서버에 접속해야하는지 알려주는 번호



## EJS

; 서버 데이터를 HTML에 쉽게 넣을 수 있도록 도와주는 HTML 렌더링 엔진

### ■ ejjs 설치하기

- ① 패키지 아이콘 클릭
- ② 패키지명 'ejs' 검색
- ③ + 버튼 눌러 패키지 설치

🔗 npm install [ 패키지명 ] 과 동일한 작업.

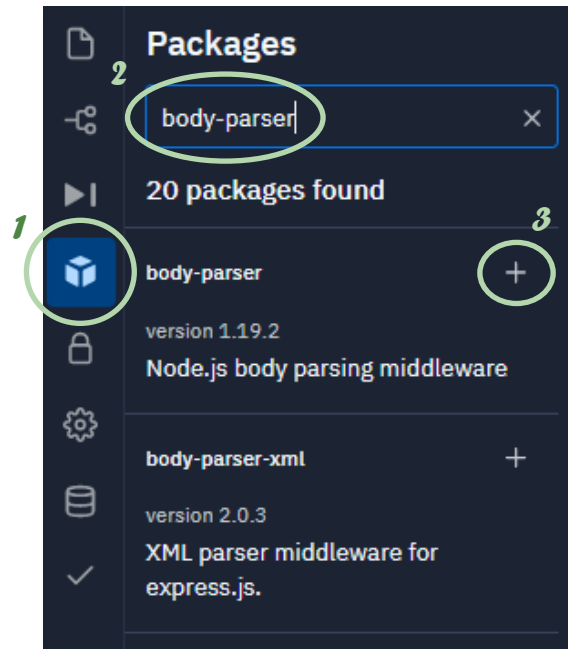
다른 패키지들도 모두 이런 방식으로 설치하면 됩니다 !

## body-parser

; req.body 데이터를 사용자가 원하는 형태로 parsing 하여 사용할 수 있게 해주는 역할

### ■ body-parser 설치하기

- ① 패키지 아이콘 클릭
- ② 패키지명 'body-parser' 검색
- ③ + 버튼 눌러 패키지 설치



```
const bodyParser = require('body-parser')
app.use(express.urlencoded({extended : true}))
app.use(express.static(__dirname + '/views'))
app.set('view engine', 'ejs')
```

- ejs, body-parser

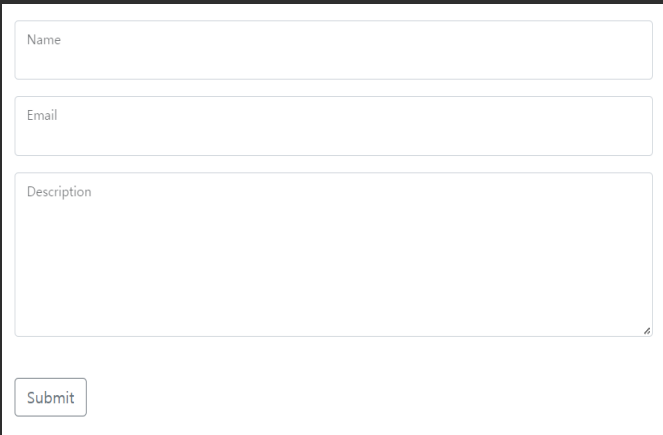
```
app.get('/', function (req, res) {
  res.render('index.ejs');
})
app.get('/write', function (req, res) {
  res.render('write.ejs');
})
app.get('/list', function (req, res) {
  res.render('list.ejs');
})
```

- res.render

; 서버가 클라이언트에게 특정 view 템플릿을 이용해  
html을 빌드하여 보내겠다 !

- req : 클라이언트의 request 에 대한 정보
- res : 서버가 클라이언트에게 response 할 정보

```
<form action="/add" method="POST">
```



Name

Email

Description

Submit

```
</form>
```

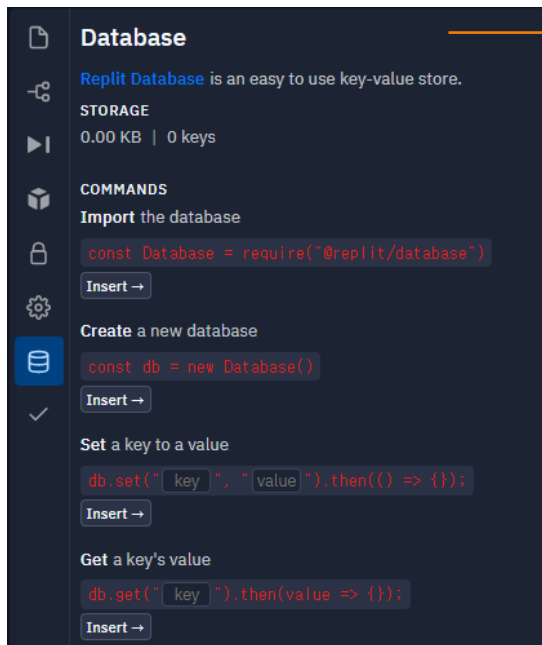
## action

폼을 서버 쪽의 어디로 전송할지

## method

GET, POST 중 하나를 선택

(cf. POST 방식이 GET 방식보다 보안성이 높다 !)



```
Database
Replit Database is an easy to use key-value store.
STORAGE
0.00 KB | 0 keys
COMMANDS
Import the database
const Database = require("@replit/database")
Insert →
Create a new database
const db = new Database()
Insert →
Set a key to a value
db.set("key", "value").then(() => {});
Insert →
Get a key's value
db.get("key").then(value => {});
Insert →
```

## Replit 기본 제공 데이터베이스

; key - value 구조 (Set, Get, Delete, List)

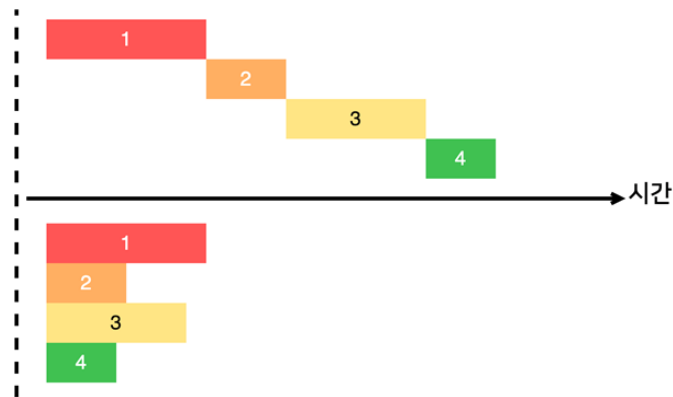
; NoSQL, 비관계형 데이터베이스

; @replit/database 패키지 설치

key	value
김노드	"node@mmm.mmm", "안녕하세요 !"
아마존	"aws@mmm.mmm", "저는 아마존"
강큐밀	"kumily@mmm.mmm", "큐시즘 안녕"

## Javascript의 비동기(Asynchronous)

- ; 병렬적으로 작업을 실행하여, 작업의 순서가 확실하지 X
- ; 서버에 무언가 요청했을 때, 서버가 요청에 대한 응답을 언제 줄 지 모르는데 마냥 기다릴 수 없기 때문에 필요한 특성





새로고침 없이 서버와 데이터를 주고받을 수 있도록 해주는 코드

[ 옵션 ]

method : 어떤 HTTP 요청을 할지

url : 서버의 어디로 요청을 보낼지

data : 어떤 데이터를 보낼지

[ 메소드 ]

.done( ) : 성공

.fail( ) : 실패

.always( ) : 성공, 실패에 관계없이 언제나





```
res.status(400).send({ message : "name 오류"})
```

- 200 OK : 요청이 성공적으로 처리
- 400 Bad Request : 잘못된 문법으로 서버가 이해할 수 없는 요청
- 404 Not Found : 알려지지 않은 URL
- 500 Internal Server Error : 서버에서 처리 시에 문제가 생김



### Querystring

? 이후 = 으로 연결된 데이터를 전달하는 방식

```
/getValue?name=김노드
```

### Path

경로를 변수로서 사용하여 데이터를 전달하는 방식

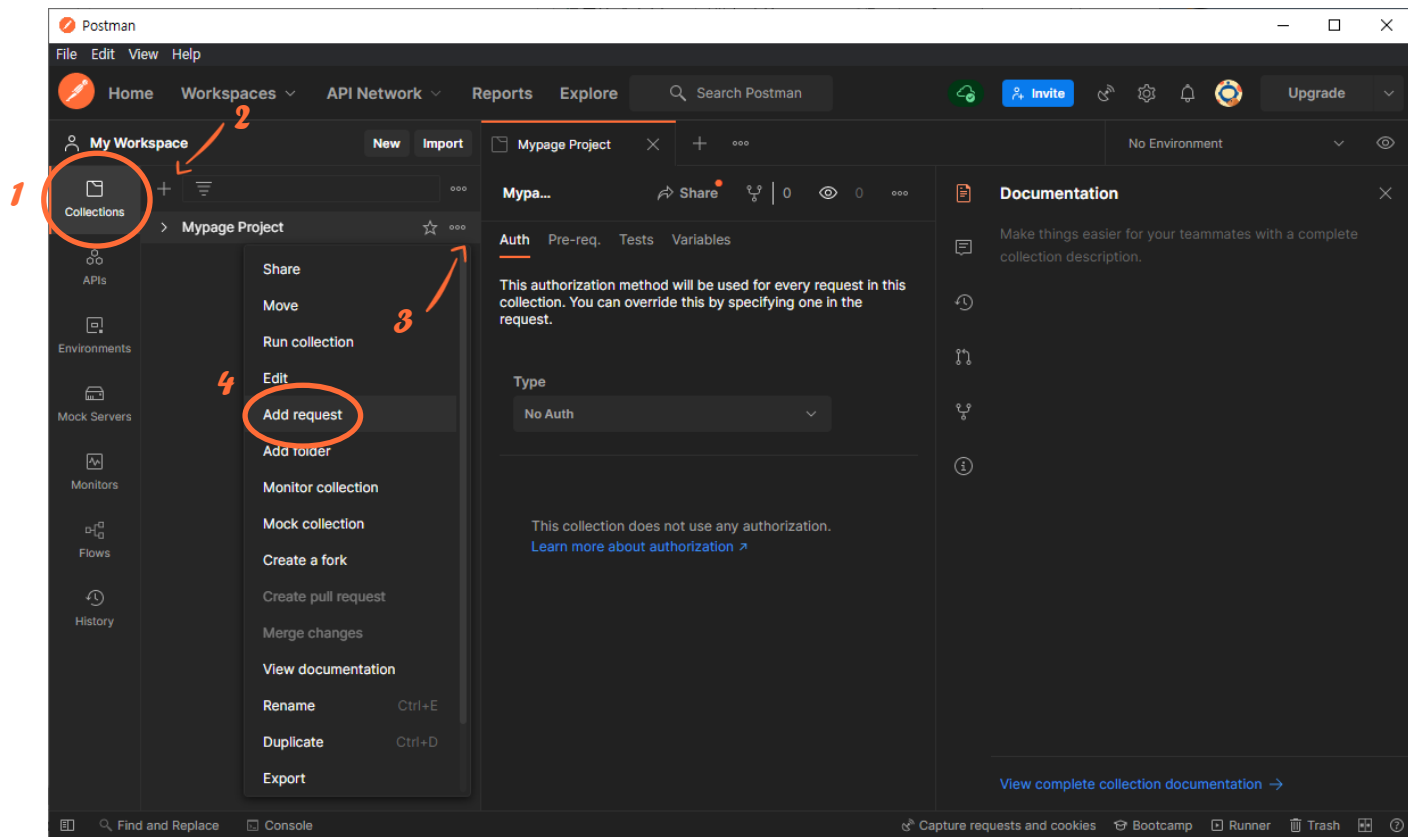
```
/getValue/김노드
```

*03*

## API 테스트 및 문서 작성

Postman 사용해보기

# Collection, Request 생성



# GET 요청 테스트

The screenshot shows the Postman interface with a GET request configured. The annotations are as follows:

- 1**: Points to the request name `/getValue/:key` in the top bar, with the text "이름 설정" (Set name).
- 2**: Points to the `GET` method dropdown in the top bar.
- 3**: Points to the URL `https://mypage-projectfin.jaejif.repl.co/getValue/:key` in the top bar.
- 4**: Points to the value `김노드` in the Path Variables table.
- 5**: Points to the `Send` button in the top right.

**Query Params Table:**

KEY	VALUE	DESCRIPTION
Key	Value	Description

**Path Variables Table:**

KEY	VALUE	DESCRIPTION
key	김노드	Description

**Body Tab (JSON):**

```
1 {
2   "name": "김노드",
3   "email": "nodejs@mmm.mmm",
4   "description": "안녕하세요 ~! Node.js 공부 중입니다"
5 }
```

# API 테스트 결과 저장

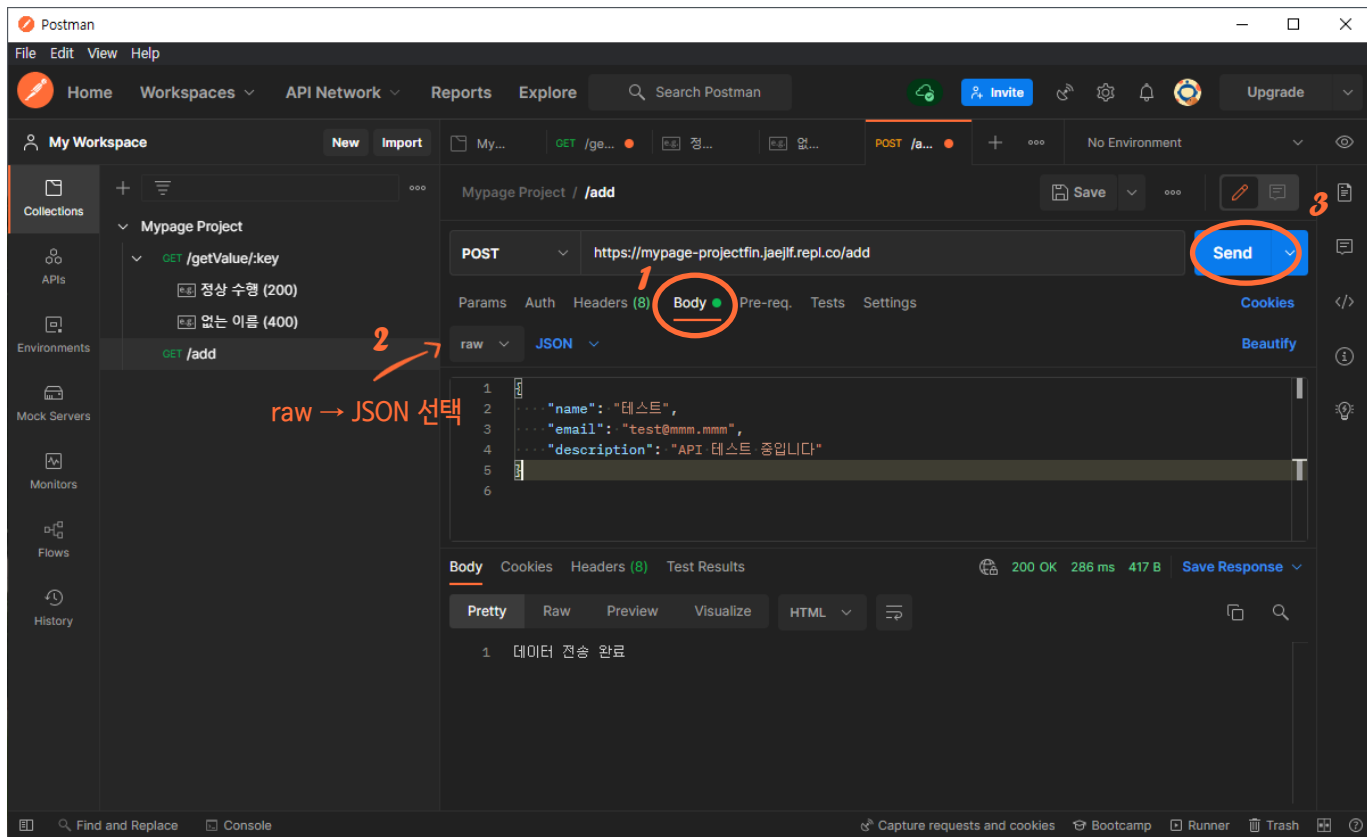
Postman interface showing an API test result. The request is a GET to `https://mypage-projectfin.jaejif.repl.co/getValue/:key`. The response status is 200 OK. The response body is JSON:

```
1 {
2   "name": "김노드",
3   "email": "nodejs@mmm.mmm",
4   "description": "안녕하세요 ~! Node.js 공부 중입니다"
5 }
```

Annotations:

- Status Code**: Points to the 200 OK status.
- output**: Points to the response body.
- 테스트 결과 저장**: Points to the **Save as example** button.

# POST 요청 테스트



# API 문서 작성

The screenshot displays the Postman application window. The left sidebar shows the 'My Workspace' section with a 'Collections' list containing 'Mypage Project'. Under 'Mypage Project', there are two items: 'GET /getValue/key' and 'GET /add'. The 'GET /add' item is selected, and its details are shown in the main panel. The URL is 'https://mypage-projectin.jaejif.repl.co/getValue'. The method is 'GET'. The 'Send' button is visible. The 'Documentation' tab is active, showing a list of items: '입력된 key에 해당하는 데이터 리턴', '성공 (status code : 200)', and '실패 (status code : 400)'. A red circle highlights the 'Documentation' tab icon, and a red arrow points to it with the text 'Markdown 문법'.

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import Mypage ... GET /getValue... 정상 수월... 없는 이월... POST /add No Environment

Collections + Mypage Project

- GET /getValue/key
  - 정상 수행 (200)
  - 없는 이월 (400)
- GET /add
  - 정상 수행(200)

APIs Environments Mock Servers Monitors Flows History

GET https://mypage-projectin.jaejif.repl.co/getValue Send

Params Auth Headers (6) Body Pre-req. Tests Settings

Query Params

KEY	VALUE	DESCRIF	Bulk Edit
Key	Value	Description	

Path Variables

KEY	VALUE	DESCRIF	Bulk Edit
key	이노드	Description	

Body 400 Bad Request 293 ms 433 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [2]
2 "message": "NOT FOUND"
3 [2]
```

Documentation

입력된 key에 해당하는 데이터 리턴

- 성공 (status code : 200)
- 실패 (status code : 400)

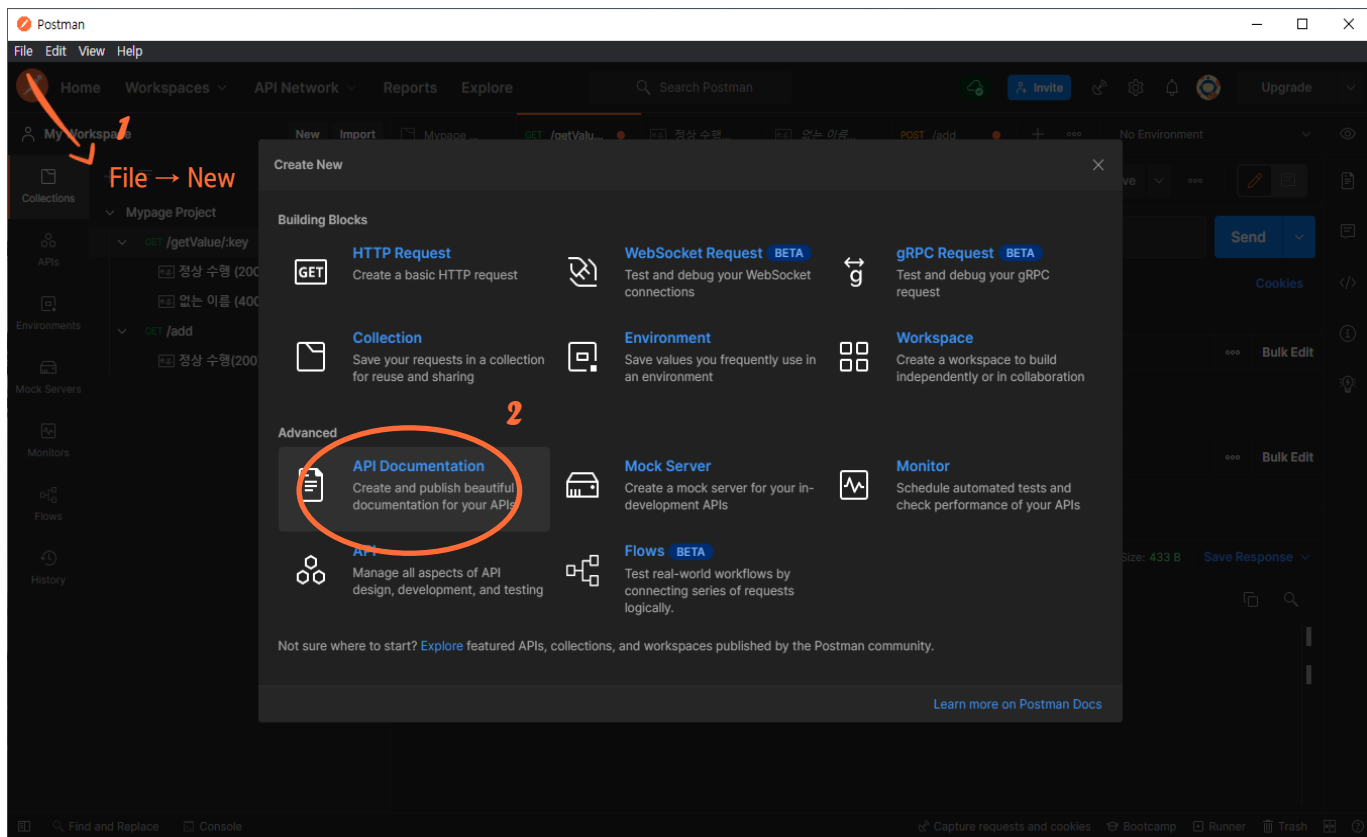
Markdown 문법

View complete collection documentation →

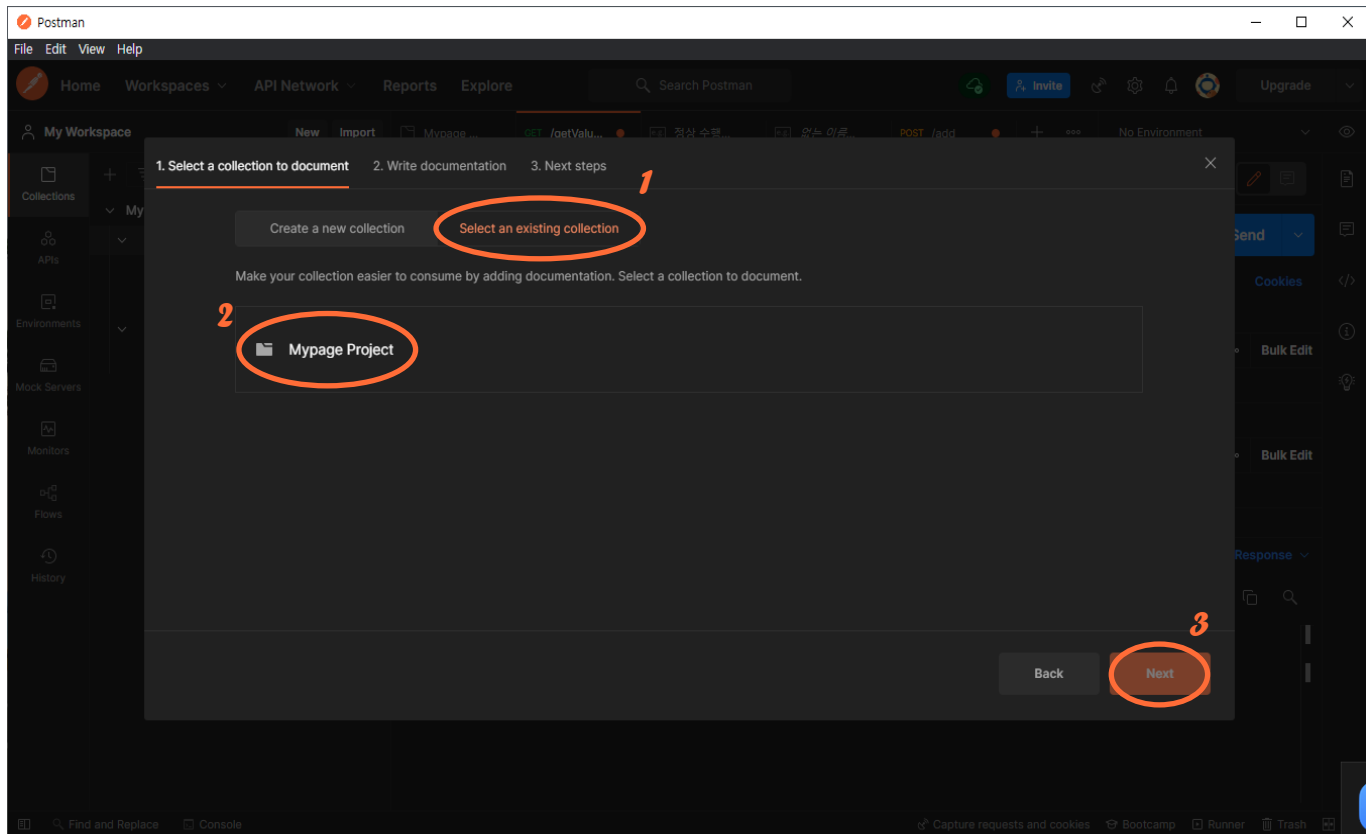
Find and Replace Console Capture requests and cookies Bootcamp Runner Trash



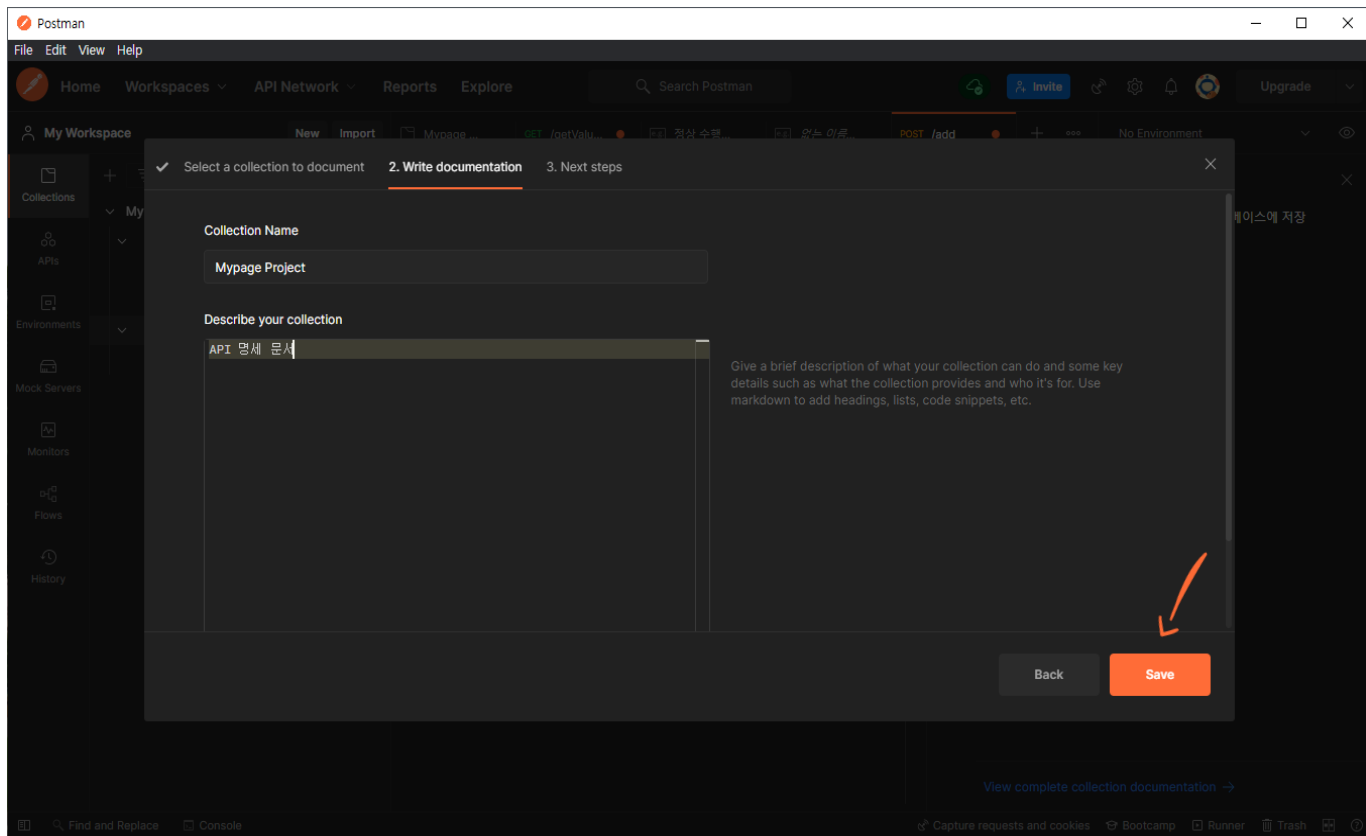
# API 문서 내보내기



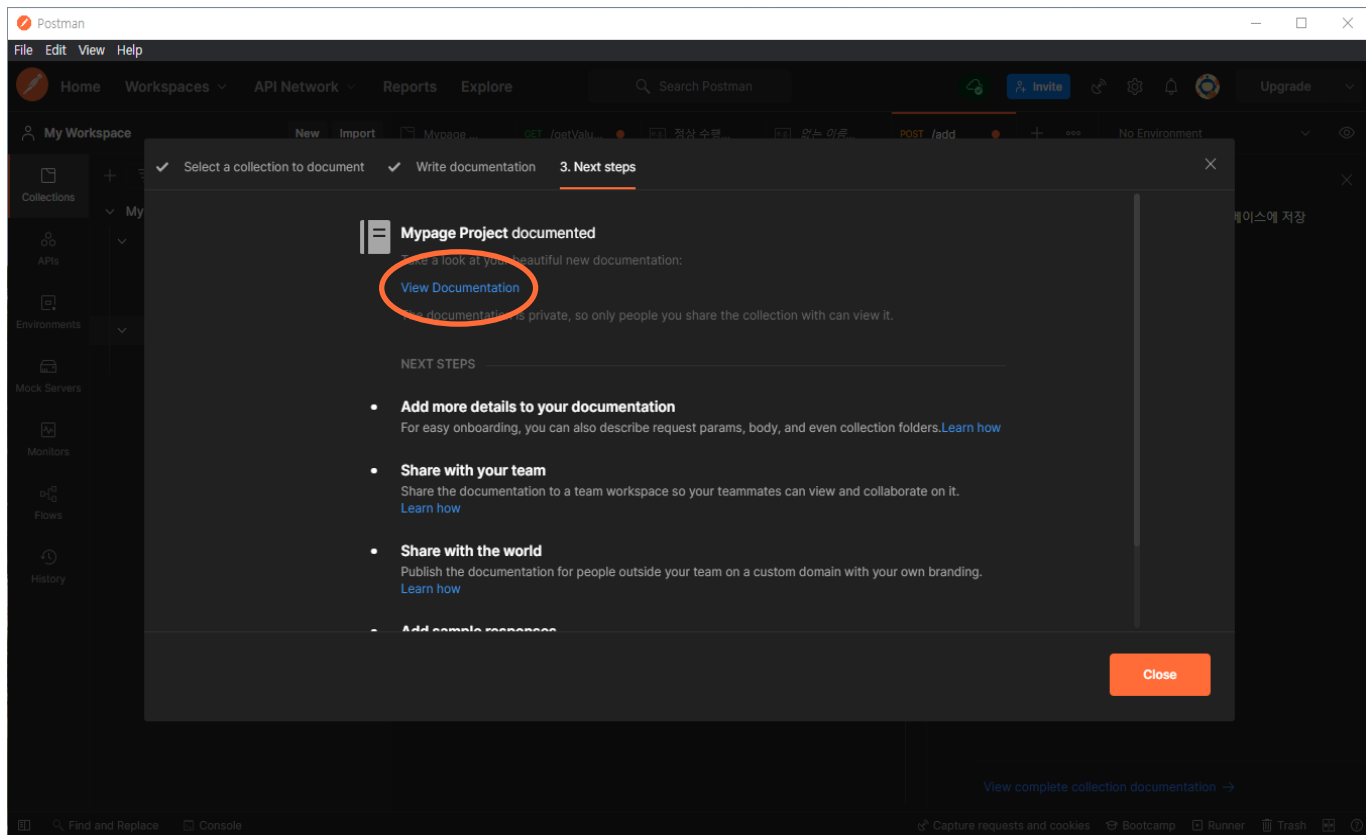
# API 문서 내보내기



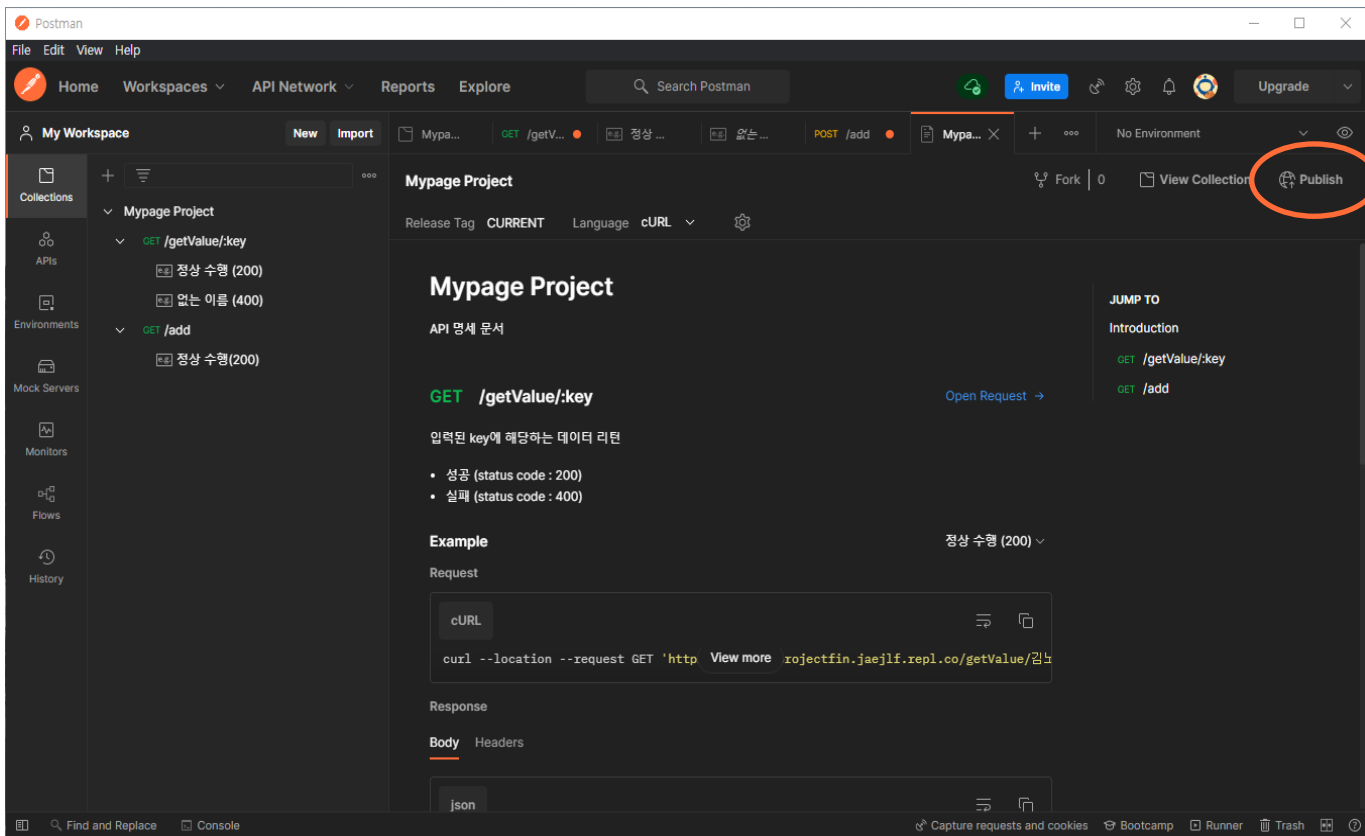
# API 문서 내보내기



# API 문서 내보내기



# API 문서 내보내기



# 04

## API 서버 배포

Heroku & Github을 사용한 API 서버 배포

# Heroku 배포를 위한 코드 수정

## ① Port 설정

헤로쿠에서 사용하는 포트는 헤로쿠 서버의 환경변수가 결정하므로, `process.env.PORT`로 포트 설정 변경

## ② “start” 명령어 설정

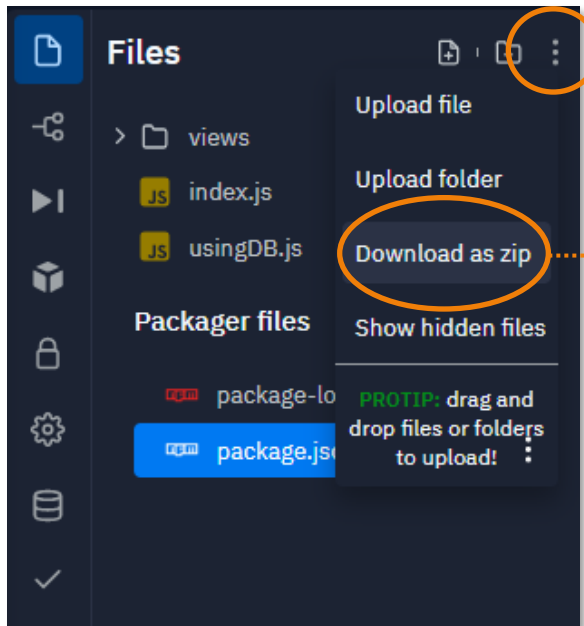
헤로쿠는 기본적으로 `npm start` 명령어를 사용해 프로젝트를 실행

## ③ devDependencies → dependencies 로 이동

헤로쿠는 `devDependencies`를 읽지 않음. (\*devDependencies : -D 옵션으로 설치한 패키지)

# Replit 코드 다운로드

(\* 실습을 위해 온라인 IDE를 사용했기 때문에, VS Code 에디터 등을 사용해서 로컬에서 개발하는 보통의 프로젝트와 같은 환경을 만들기 위한 과정)



압축 파일로 다운로드 받은 후, 로컬의 원하는 위치에서 압축 해제



# Github 레포지토리 생성

The screenshot shows the GitHub profile of 'Jaejeong Sohn' (jaejlf). The interface is in dark mode. At the top, there's a navigation bar with 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the user's profile is displayed on the left, including their avatar (a cartoon character), name, and location. The main content area shows the user's repositories. The 'Repositories' tab is selected and circled with a green line and a green '1'. To the right of the 'Repositories' tab, there's a 'New' button, also circled with a green line and a green '2'. The 'New' button is a green square with a white computer icon and the word 'New' in white. Below the 'New' button, there are three repositories listed: 'Mypage-Project' (Private), 'TIL' (Private), and 'Algorithm' (Public). Each repository has a 'Star' button and a dropdown arrow. The 'Mypage-Project' repository has a description '[큐시즘 세션] 백엔드 시스템 입문 : REST API 설계 및 배포' and was updated yesterday. The 'TIL' repository has a description 'Today I Learn' and was updated 4 days ago. The 'Algorithm' repository has tags 'cpp' and 'algorithm'.

Search or jump to... Pull requests Issues Marketplace Explore

Overview **Repositories 14** Projects Packages Stars 11

Find a repository... Type Language Sort **New**

**Mypage-Project** Private Star  
[큐시즘 세션] 백엔드 시스템 입문 : REST API 설계 및 배포  
EJS Updated yesterday

**TIL** Private Star  
Today I Learn  
algorithm database react-native node-js server  
Updated 4 days ago

**Algorithm** Public Star  
cpp algorithm

**Jaejeong Sohn**  
jaejlf  
Edit profile  
14 followers · 13 following  
Sookmyung Women's Univ.  
Seoul, Korea

# Heroku app 생성

The screenshot shows the Heroku dashboard interface. At the top, there's a navigation bar with the Heroku logo, a search bar, and a user profile dropdown. The main content area has a purple header with a 'Welcome to Heroku' message. Below this, there are two main cards: 'Create a new app' and 'Create a team'. The 'Create a new app' card is highlighted with a purple arrow. A dropdown menu is open next to the 'New' button in the top right, showing options for 'Create new app' and 'Create new pipeline'. The footer contains links for help and getting started.

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Personal

(App을 만든 적이 있다면 여기 ▼)

New

Create new app

Create new pipeline

Dismiss

Welcome to Heroku

Now that your account has been set up, here's how to get started.

Create a new app

Create your first app and deploy your code to a running dyno.

Create new app

Create a team

Create teams to collaborate on your apps and pipelines.


Create a team

Looking for help getting started with your language?



Get started by reading one of our language guides in the Dev Center

<https://dashboard.heroku.com/new-app?org=personal-apps>

# Heroku app 생성

 **HEROKU**

Jump to Favorites, Apps, Pipelines, Spaces...



Create New App

1


App name

mypage-project

✓

mypage-project is available

Choose a region

 United States

⌵

Add to pipeline...

2

Create app

heroku.com

Blogs

Careers

Documentation

Support

Terms of Service

Privacy

Cookies

© 2022 Salesforce.com

# Heroku → Github 접근 권한 부여

The screenshot shows the Heroku dashboard interface. At the top, there's a navigation bar with the Heroku logo and a search bar. Below the navigation bar, there's a dropdown menu labeled 'Choose a pipeline'. The main content area is titled 'Deployment method' and contains three options: 'Heroku Git' (Use Heroku CLI), 'GitHub' (Connect to GitHub), and 'Container Registry' (Use Heroku CLI). The 'GitHub' option is circled in purple and labeled with a '1'. Below this, there's a section titled 'Connect to GitHub' with instructions on how to connect the app to a GitHub repository. At the bottom of this section, there's a purple button labeled 'Connect to GitHub', which is pointed to by a purple arrow labeled with a '2'.

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Choose a pipeline

Deployment method

Heroku Git  
Use Heroku CLI

GitHub  
Connect to GitHub

Container Registry  
Use Heroku CLI

Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

View your code diffs on GitHub

Connect your app to a GitHub repository to see commit diffs in the activity log.

Deploy changes with GitHub

Connecting to a repository will allow you to deploy a branch to your app.

Automatic deploys from GitHub

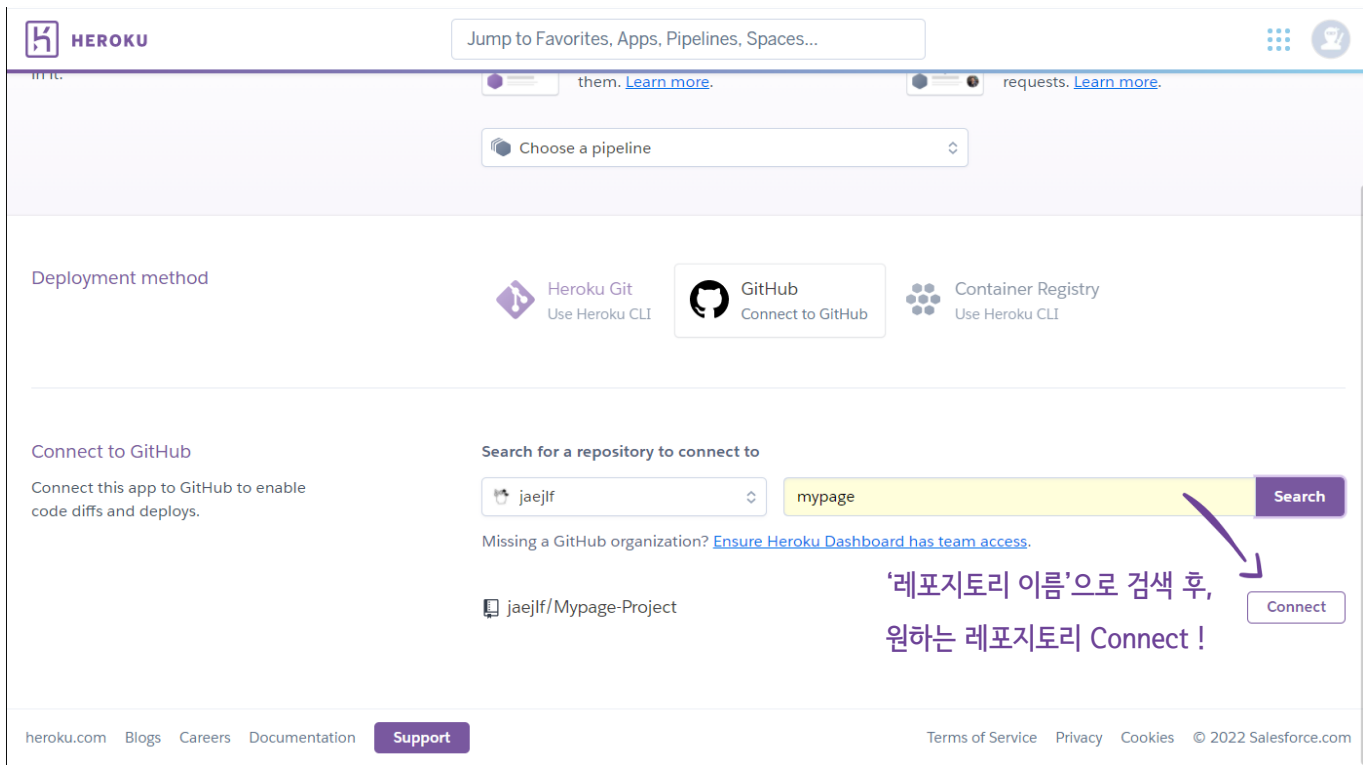
Select a branch to deploy automatically whenever it is pushed to.

Create review apps in pipelines

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more](#).

Connect to GitHub

# Heroku - Github 레포지토리 연결



**HEROKU** Jump to Favorites, Apps, Pipelines, Spaces...

them. [Learn more.](#) requests. [Learn more.](#)

Choose a pipeline

**Deployment method**

- Heroku Git Use Heroku CLI
- GitHub Connect to GitHub
- Container Registry Use Heroku CLI

**Connect to GitHub**

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

jaejlf mypage **Search**


Missing a GitHub organization? [Ensure Heroku Dashboard has team access.](#)

jaejlf/Mypage-Project



**‘레포지토리 이름’으로 검색 후, 원하는 레포지토리 Connect !**

heroku.com Blogs Careers Documentation **Support** Terms of Service Privacy Cookies © 2022 Salesforce.com

# 배포 !

 **HEROKU**

Jump to Favorites, Apps, Pipelines, Spaces...



☐ Wait for CI to pass before deploy  
Only enable this option if you have a Continuous Integration service configured on your repo.

**Enable Automatic Deploys**



### Manual deploy

Deploy the current state of a branch to this app.

### Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)


**Choose a branch to deploy**

 main 


**Deploy Branch**

*클릭 !*


Receive code from GitHub




Build **main** 084d1145




Release phase



Deploy to Heroku



Your app was successfully deployed.

 View

**Thank You -!**

