

[MASOCON 2017]

R로 만드는 API서버 삽질기

https://mrchypark.github.io/MSM_apiR

스타누르기는 컨텐츠 제작자를 춤추게 합니다.

[pdf버전] [문의하기] [의견 및 오류 신고]

**MICRO
SOFTWARE**

박찬업

발표자 소개

질문 / 상담 / 잡담 대화영!

박찬엽



- 서울도시가스 선행연구팀 연구원
 - 챗봇 엔진 개발 및 서버 구축
- 패스트 캠퍼스 데이터 분석 R 강의
 - 데이터 분석을 위한 중급 R 프로그래밍
- R 네이버 뉴스 크롤러 N2H4 관리자
 - ForkonLP 프로젝트
 - 뉴스 약 3천만건으로 wordvec 모델 공개 예정
- KAKAO@알코홀릭 R 질문방
- FACEBOOK@mrchypark
- GITHUB@mrchypark

R로 만드는 API서버 삽질기

자매품

R로 웹 데이터를 가져오는 4가지 방법
(은 크롤링)

의사 결정을 위한 데이터 분석 프로젝트의 단계

데이터 확보 - 전처리 - 모델링 - 시각화 - 문서화

머신러닝 프로젝트의 단계

데이터 확보 - 전처리 - 모델링 - **제품화**

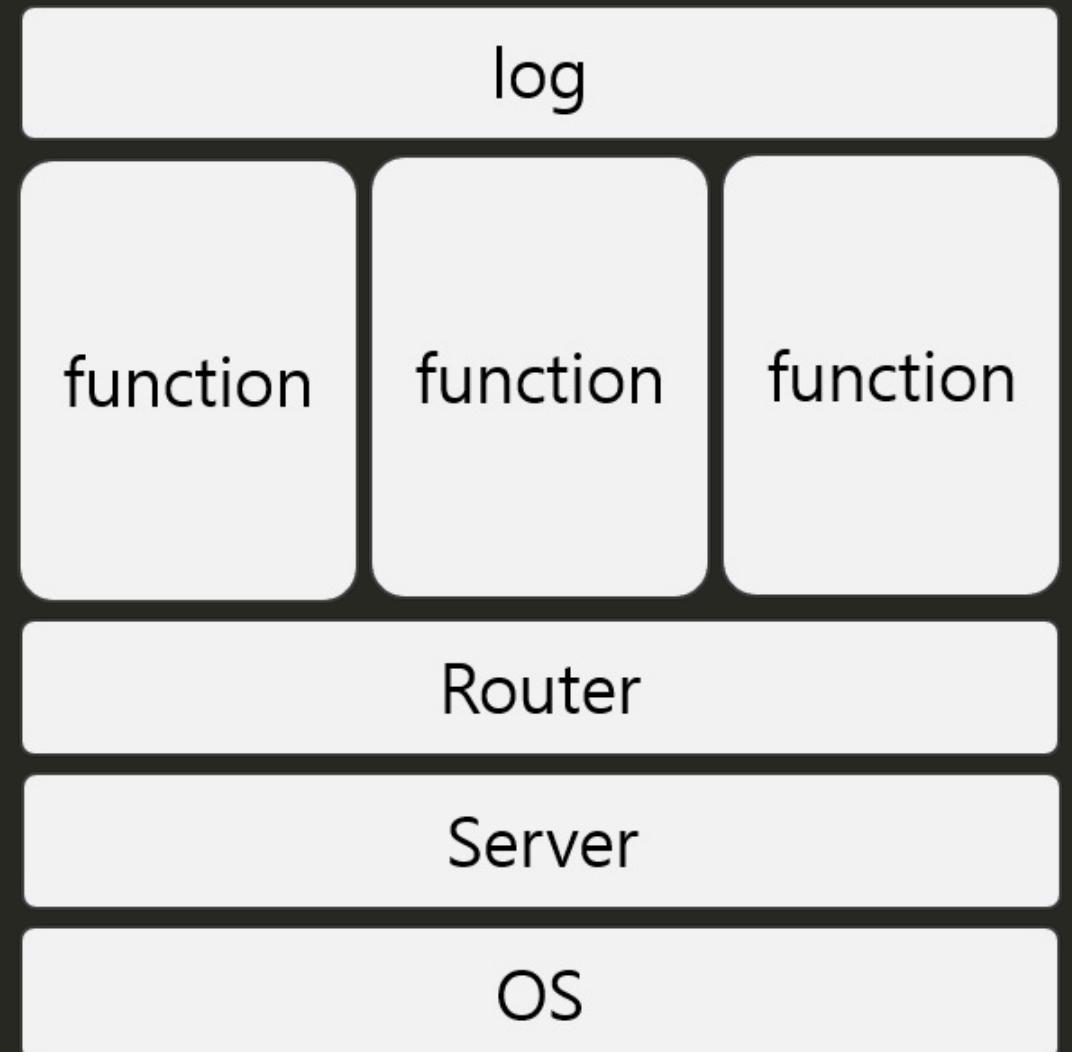
제품화

머신러닝 프로젝트의 결과물을 서비스에 활용하는 것 또한 많은 노력이 필요함



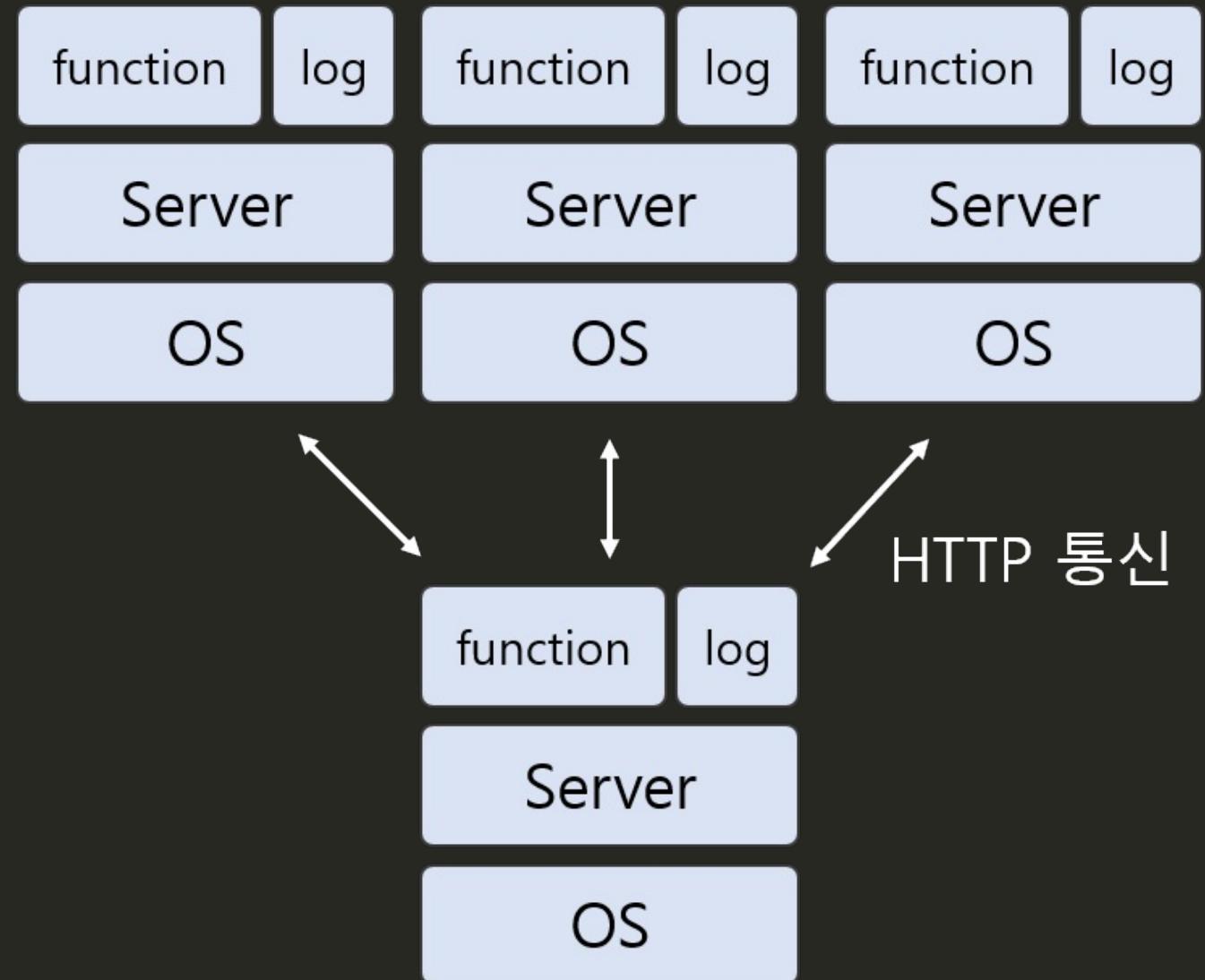
제품화하는데 노력이 덜 드는 방법은 없을까

Monolithic Architecture



외부 요청

Micro Service Architecture



외부 요청

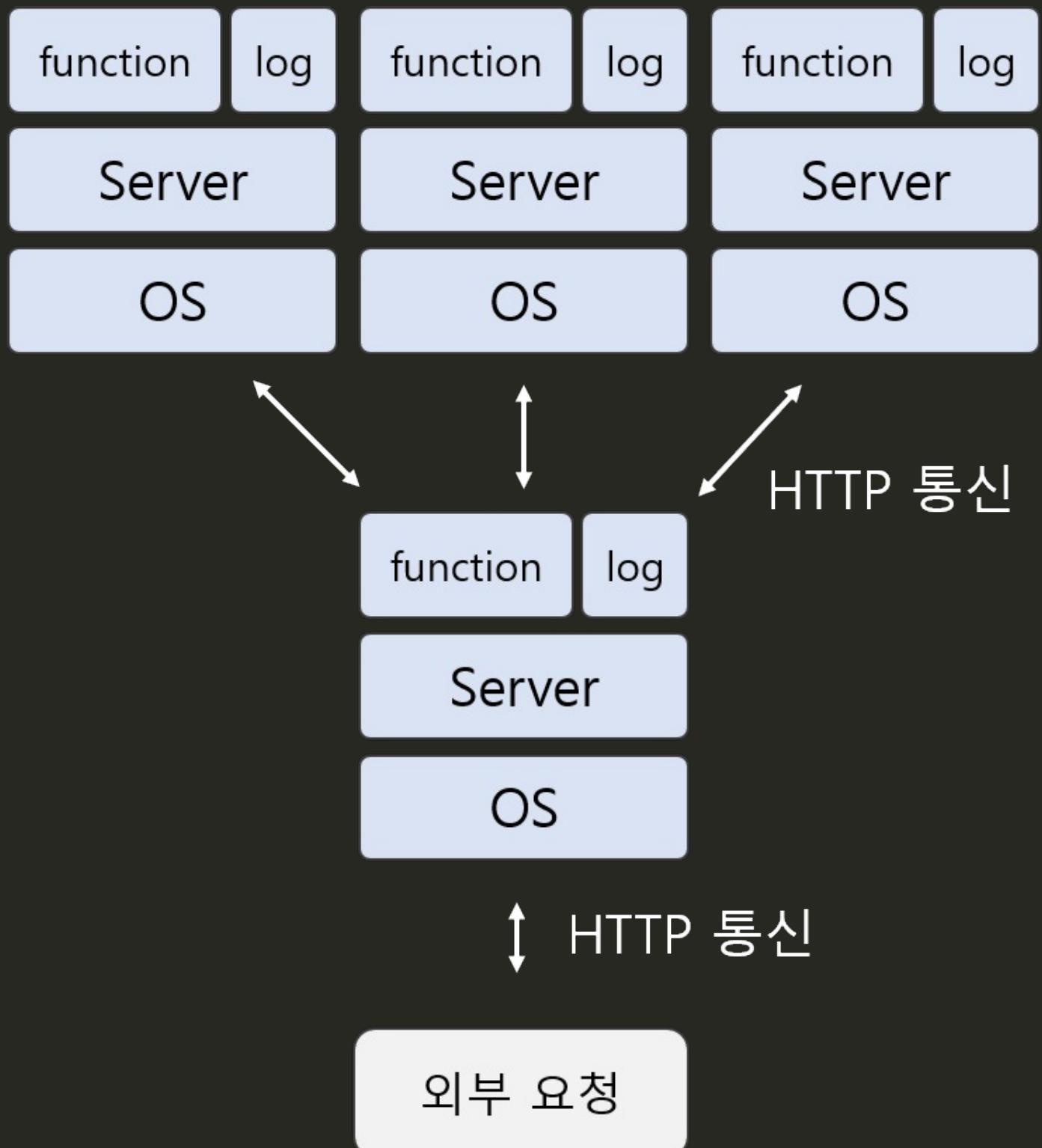
마이크로 서비스형 설계

장점

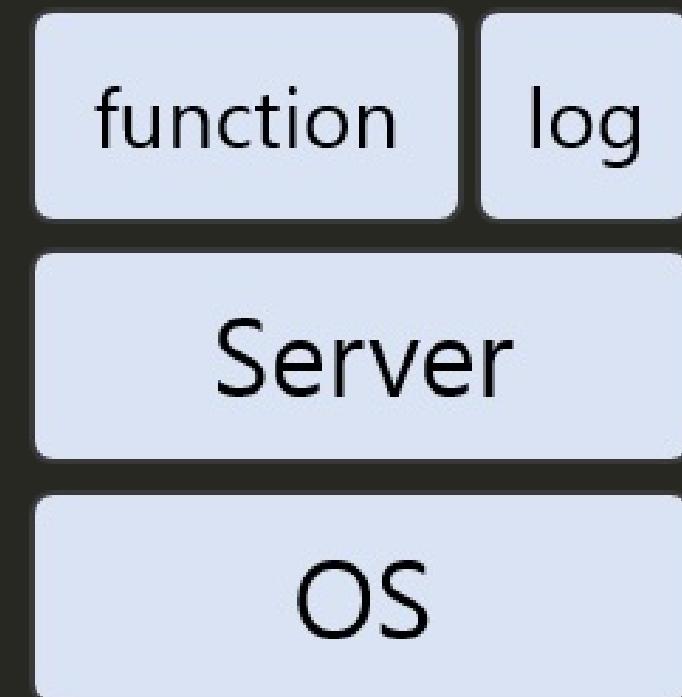
1. 기능 단위로 완결되어 개발 단위가 작음
2. 개별 서비스간의 의존성이 적어 유연함
3. 장애시 전체 서비스의 영향이 적음

단점

1. 개별 서비스 테스트 + 전체 테스트가 필요
2. 서비스간 interface 관리 이슈가 발생
3. 전체 서비스의 추가적인 로그 관리 필요



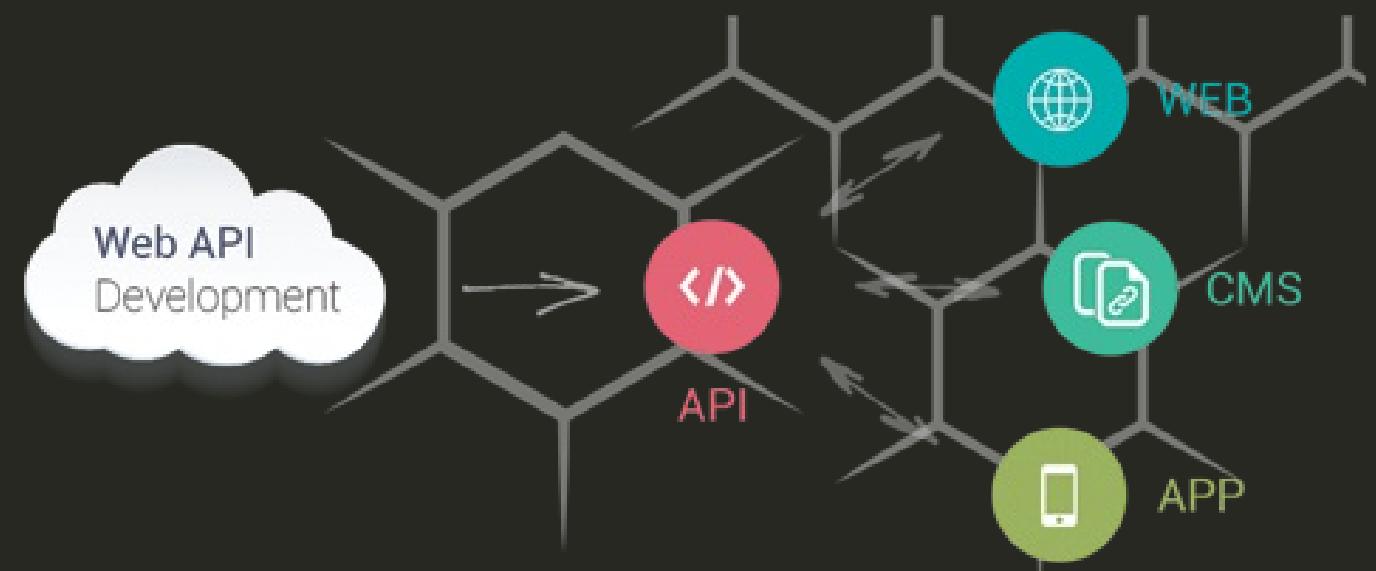
그럼 이거만 만들 줄 알면...!



그럼 작은 API 서버를 R로 만들어 보자

API 서버

http 표준으로 요청을 받아서 처리하여 응답하는 서버



http/1.1 명세

HTTP 메소드 ◆	RFC ◆	요청에 Body가 있음 ◆	응답에 Body가 있음 ◆	안전 ◆	멱등(Idempotent) ◆	캐시 가능 ◆
GET	RFC 7231 ↗	아니오	예	예	예	예
HEAD	RFC 7231 ↗	아니오	아니오	예	예	예
POST	RFC 7231 ↗	예	예	아니오	아니오	예
PUT	RFC 7231 ↗	예	예	아니오	예	아니오
DELETE	RFC 7231 ↗	아니오	예	아니오	예	아니오
CONNECT	RFC 7231 ↗	예	예	아니오	아니오	아니오
OPTIONS	RFC 7231 ↗	선택 사항	예	예	예	아니오
TRACE	RFC 7231 ↗	아니오	예	예	예	아니오
PATCH	RFC 5789 ↗	예	예	아니오	아니오	예

아 됐고!



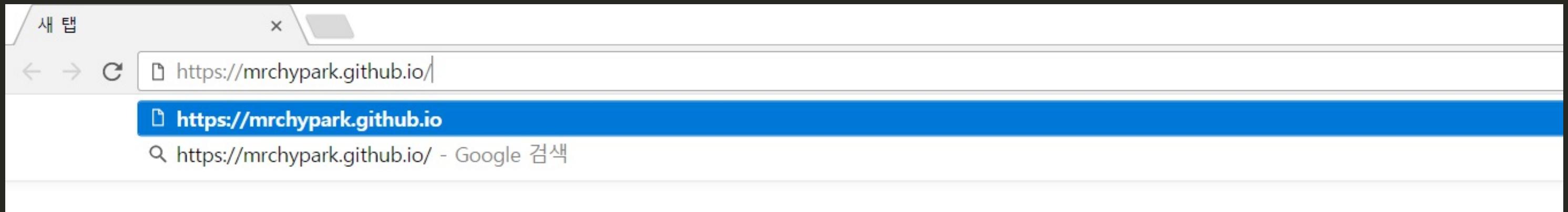
두 가지만 알면 됩니다.

GET vs POST

심지어 POST만 알아도 됨.

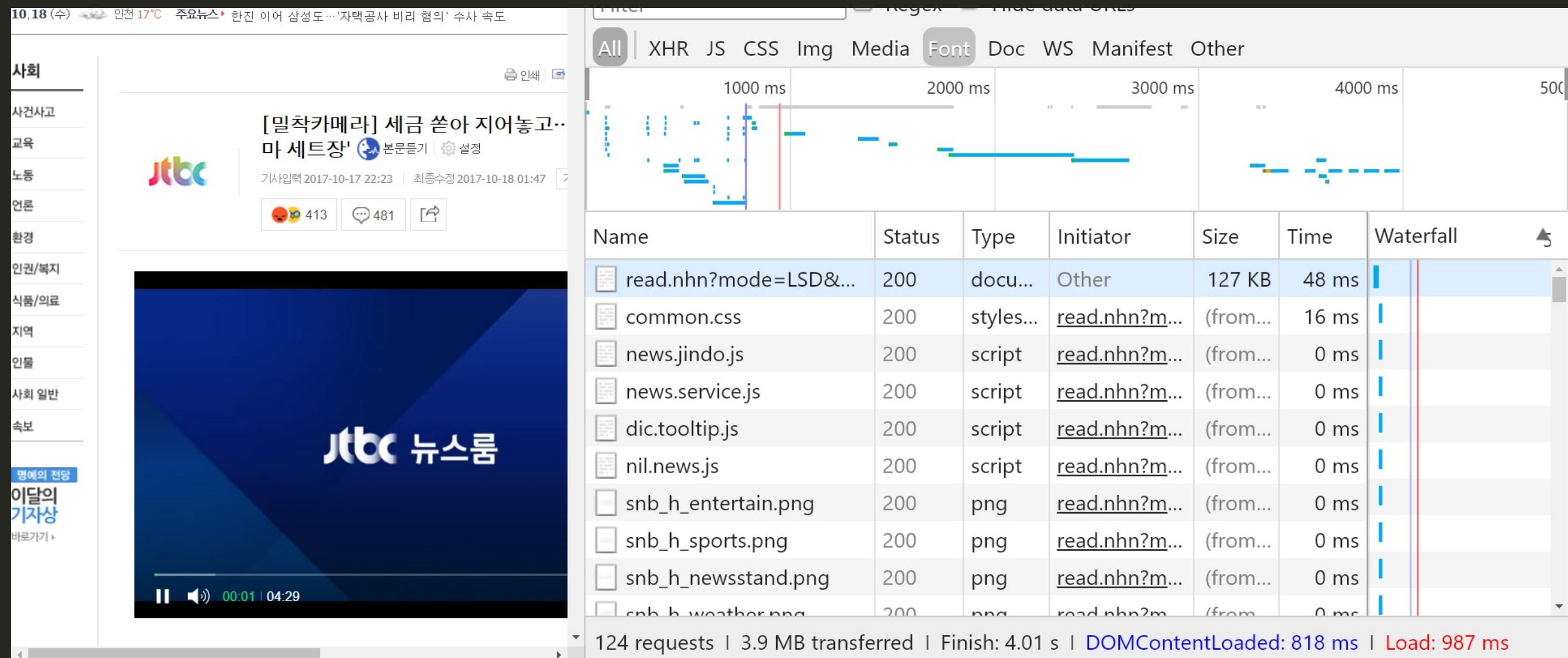
GET 이란

브라우저에서 주소를 입력하고 엔터를 치는 것!



그럼 서버는

요청하면 주기로 정해진 데이터(ex> 이미지 파일)를 제공



The screenshot shows a web browser window. On the left, there's a sidebar with various news categories like '사회', '사건사고', '교육', etc. The main content area displays a news article from JTBC. The headline reads '[밀착카메라] 세금 쓴아 지어놓고.. 마 세트장' with a timestamp of 2017-10-17 22:23 and 2017-10-18 01:47. Below the headline are interaction counts (413 likes, 481 comments) and a share button. The background of the main content area is dark blue with the JTBC logo and '뉴스룸' text.

On the right, the Network tab of the developer tools is open, showing a timeline and a list of requests. The timeline at the top shows several requests as blue bars, with time markers at 1000 ms, 2000 ms, 3000 ms, and 4000 ms. The list below shows the details of these requests:

Name	Status	Type	Initiator	Size	Time	Waterfall
read.nhn?mode=LSD&... (from)	200	document	Other	127 KB	48 ms	
common.css (from)	200	stylesheet	read.nhn?m...	16 ms		
news.jindo.js (from)	200	script	read.nhn?m...	0 ms		
news.service.js (from)	200	script	read.nhn?m...	0 ms		
dic.tooltip.js (from)	200	script	read.nhn?m...	0 ms		
nil.news.js (from)	200	script	read.nhn?m...	0 ms		
snb_h_entertain.png (from)	200	png	read.nhn?m...	0 ms		
snb_h_sports.png (from)	200	png	read.nhn?m...	0 ms		
snb_h_newsstand.png (from)	200	png	read.nhn?m...	0 ms		
snb_h_weather.png (from)	200	png	read.nhn?m...	0 ms		

At the bottom of the developer tools, there's a summary: 124 requests | 3.9 MB transferred | Finish: 4.01 s | DOMContentLoaded: 818 ms | Load: 987 ms.

우리 GET 서버가 할 일

서버에게 GET 요청이 오면 필요한 데이터를 전달해주기.

"안녕하세요!!"

jug를 소개합니다!

```
if (!requireNamespace("jug")) {  
  install.packages("jug")}  
library(jug)  
  
jug() %>%  
  get("/", function(req, res, err){  
    res$json(enc2utf8("안녕하세요!!"))  
    return(res)  
}) %>%  
  simple_error_handler_json() %>%  
  serve_it()
```

Serving the jug at <http://127.0.0.1:8080>

R로 GET 요청하기

R에서 http 표준의 요청을 처리해 주는 유용한 패키지는 `httr`입니다.

```
if (!requireNamespace("httr")) {  
  install.packages("httr")}  
  
## Loading required namespace: httr  
  
library(httr)  
  
target<- "http://127.0.0.1:8080/"  
content(GET(url=target))  
  
## [1] "안녕하세요!!"
```

GET 서버를 만들었다!

jug 패키지를 이용해 R로 api 서버를 만드는 만듬

이제 POST 서버를 만들어 보자

더 알아야 할 것

요청(request) vs 응답(response) + 헤더(header)¹

1. 헤더(header): http://www.w3ii.com/ko/http/http_header_fields.html

GET과 POST의 차이점

요청 헤더에 body라는 이름으로 값을 보내면 POST이고 없으면 GET

GET과 POST의 차이점

요청 헤더에 body라는 이름으로 값을 보내면 POST이고 없으면 GET

R 함수로 예를 들면

```
# 입력없이 결과를 주는 GET  
R.Version()
```

```
## $platform  
## [1] "x86_64-w64-mingw32"  
##  
## $arch  
## [1] "x86_64"  
##  
## $os  
## [1] "mingw32"  
##  
## $system  
## [1] "x86_64, mingw32"  
##  
## $status  
## [1] ""  
##  
## $major  
## [1] "3"
```

```
# a, b 입력으로 결과를 주는 POST  
a<-1;b<-2  
sum(a,b)
```

```
## [1] 3
```

우리 POST 서버가 할 일

서버에게 POST 요청이 한글 문장을 "sent"라는 이름으로 같이 보내면 띄어쓰기 개수를 세서 줌.

```
# 입력  
"{sent='R로 만드는 API서버 삽질기'}"  
  
# 예상 결과  
"3"
```

stringr with jug

stringr은 정규표현식을 활용한 글자 처리를 도와주는 패키지

```
if (!requireNamespace("jug")) {  
  install.packages("jug")}  
library(jug)  
  
if (!requireNamespace("stringr")) {  
  install.packages("stringr")}  
library(stringr)  
  
count_ws<-function(sent){  
  stringr::str_count(sent, "[[:space:]]")  
}  
  
jug() %>%  
  post("/", decorate(count_ws)) %>%  
  simple_error_handler_json() %>%  
  serve_it()
```

R로 POST 요청하기

httr로 POST 요청을 해보겠습니다.

```
library(httr)

url<-http://127.0.0.1:8080
body<-list(sent="R로 만드는 API서버 삽질기")
content(POST(url, body=body), "text")

## No encoding supplied: defaulting to UTF-8.

## [1] "3"
```

POST 서버를 만들었다!

R의 서버 개발 패키지들

`httpuv, Rserve, jug, plumber, fiery`

이렇게 많은 패키지가 있는데...

오픈소스 사용시 중요한 점.

사용자층이 넓을 것, 설명서가 충실할 것, 버전이 1.X.X 이상일 것 등등

reticulate 를 소개합니다!

python 패키지를 R에서 사용할 수 있게 해주는 패키지

Flask

python의 가볍고 유명한 웹 서버 개발 프레임워크

github 스타 수

plumber: 453

Rserve: 161

jug: 119

fiery: 114

httpuv: 98

Flask: 30,605

나에게 힘을 조금씩만 나눠줘



lumiamitie님께서 초기 삽질을 대신해 주셨습니다!

<http://lumiamitie.github.io/r/flask-on-r/>

COMMENTS

댓글 한 건 lumiamitie

4 mrchypark ▾

추천 공유

인기순 ▾



토론 참여하기



mrchypark • 5달 전

하아 미리 해주셔서 정말 감사합니다ㅠ_ㅠ 덕분에 빨리 해볼 수 있게 됬어요!

^ | v . 수정 . 답글 . 공유 ,

GET 서버 예시

```
library('reticulate')

flask = import('flask')
app = flask$Flask('__main__')

app$route('/'){
  index = function() {return('Hello R user Conference!')}
}

app$run()
```

flask on r

장점

- flask의 풍부한 자료를 활용



단점

- python 도 알아야 함



어쨌든 서버만드는 법을 알았으니...

k톡 챗봇에 도전해 봤습니다...(!?!)

G사 클라우드 + k사 p친구 + jug(R) + mariaDB
서버 + 채팅 플랫폼 + 웹서버 + log용 DB

날씨봇을 만들어 보자

https://github.com/mrchypark/wthr_chatr

체레스터 친구하고 날씨 물어보기



log는 남기고 싶고, 일반DB는 너무 불편하고 (돈내야 하고...)

G사 클라우드 + k사 p친구 + jug(R) + G사 Form(?!?)

서버 + 채팅 플랫폼 + 웹서버 + log용 DB

googleformr

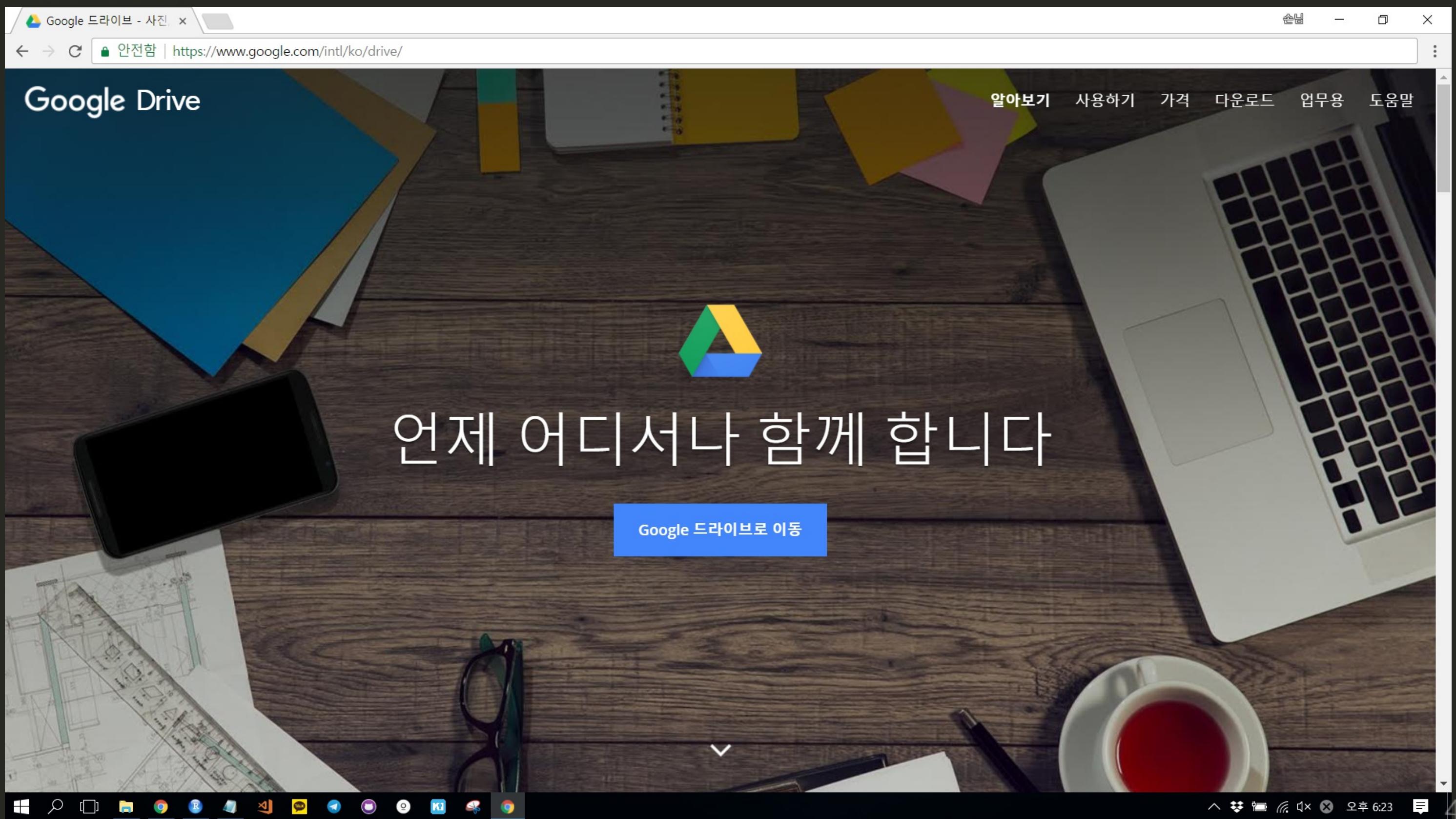
googleformr는 구글 form 기능을 이용해서 api 호출 없이 구글 sheet에 데이터를 입력하는 방법을 제공합니다.

```
if (!requireNamespace("googleformr")){
  devtools::install_github("data-steve/googleformr", dependencies = TRUE)
}
```

```
## Loading required namespace: googleformr
```

```
library(googleformr)
```

구글 폼 만들기



The image shows a screenshot of a Google Drive landing page in Korean. The page features a wooden desk background with various office items like a laptop, smartphone, coffee cup, and papers. The Google Drive logo is centered. The main text reads "언제 어디서나 함께 합니다" (Work together anywhere, anytime). A blue button below the text says "Google 드라이브로 이동" (Move to Google Drive). At the top, there's a navigation bar with links like "알아보기", "사용하기", "가격", "다운로드", "업무용", and "도움말". The browser address bar shows "안전함 | https://www.google.com/intl/ko/drive/". The taskbar at the bottom includes icons for File Explorer, Task View, Start, Search, Taskbar settings, and Google Chrome.

새로만들기

A screenshot of the Google Drive web interface. The top navigation bar shows the title "내 드라이브 - Google 드" and the URL "https://drive.google.com/drive/my-drive". The main search bar contains the text "드라이브 검색". On the left, a sidebar lists navigation options: "내 드라이브" (selected), "컴퓨터", "공유 문서함", "최근 문서함", "Google 포토", "중요", "휴지통", and "백업". The main content area shows a single file named "newsforv.tar" with a download icon. The file is listed under the "파일" section. The top right corner features a user profile icon with a 'c', and icons for "새로 만들기" (New), "내 드라이브" (My Drive), and settings. The bottom of the screen shows the Windows taskbar with various pinned icons.

[더보기](#)

A screenshot of the Google Drive web interface. The top navigation bar shows the title "내 드라이브 - Google 드" and the URL "https://drive.google.com/drive/my-drive". The main menu on the left includes options like "새로 만들기" (New), "내 드라이브" (My Drive), "파일 업로드" (Upload file), "폴더 업로드" (Upload folder), "Google 문서" (Google Document), "Google 스프레드시트" (Google Sheets), and "Google 프레젠테이션" (Google Slides). A search bar at the top right says "드라이브 검색". On the far right, there are icons for notifications and account settings. The central area displays a file named "newsforv.tar" with a download icon. The bottom of the screen shows the Windows taskbar with various pinned icons.

Google 설문지 선택

The screenshot shows the Google Drive web interface. At the top, there's a navigation bar with a back arrow, forward arrow, refresh button, and a link to 'https://drive.google.com/drive/my-drive'. Below the bar is a search field labeled '드라이브 검색' and a user profile icon with a 'c'.

The main area is titled '내 드라이브' (My Drive). On the left, there's a sidebar with options like '새로 만들기' (Create New), '폴더' (Folder), '파일 업로드' (Upload File), '폴더 업로드' (Upload Folder), 'Google 문서' (Google Document), 'Google 스프레드시트' (Google Sheets), 'Google 프레젠테이션' (Google Slides), '더보기' (More), and '백업' (Backup).

A dropdown menu is open under '새로 만들기', listing several Google app options:

- Google 문서
- Google 드로잉
- Google 내 지도
- Google 사이트 도구

At the bottom of the dropdown, there's a link to '연결할 앱 더보기' (More connected apps).

The taskbar at the bottom of the screen shows various pinned application icons, including File Explorer, Google Chrome, and other Microsoft Office applications. The system tray on the right displays network, battery, and system status icons.

설문지 작성 화면

The screenshot shows a Google Form editor window titled "제목 없는 설문지" (Untitled Survey). The interface includes a toolbar at the top with icons for file operations, search, and settings. The main area displays a single question labeled "제목없는 질문" (Untitled Question) with one option: "옵션 1" (Option 1), represented by a radio button. A sidebar on the right contains icons for adding new questions or sections, and a status bar at the bottom indicates "apis.google.com의 응답을 기다리는 중..." (Waiting for responses from apis.google.com).

내 드라이브 - Google 드라이브 - Google 설문지

안전함 | https://docs.google.com/forms/u/0/d/18yxN8iWwqgDQ3vc1MYTQWqDOClx0PoP_3tADD29raCs/edit

← 제목 없는 설문지

질문 응답

제목 없는 설문지

설문지 설명

제목없는 질문

옵션 1

apis.google.com의 응답을 기다리는 중...

53 / 73

질문 세팅 - 질문 제목으로 값을 받음

A screenshot of a Google Forms survey titled "제목 없는 설문지". The interface shows a "질문" (Question) tab selected. A context menu is open over a question labeled "제목없는 질문" (Untitled Question). The menu lists various response types: 단답형 (Single Answer), 장문형 (Long Text), 객관식 질문 (Multiple Choice), 체크박스 (Checkboxes), 드롭다운 (Dropdown), 파일 업로드 (File Upload), 직선 단계 (Linear Scale), 객관식 그리드 (Multiple Choice Grid), 체크박스 그리드 (Checklist Grid), 날짜 (Date), and 시간 (Time). The "객관식 질문" option is currently selected. The Google Drive navigation bar at the top indicates the user has 18 items in their drive.

응답탭 선택

내 드라이브 - Google 드 x 제목 없는 설문지 - Google 드 x

안전함 | https://docs.google.com/forms/d/18yxN8iWwqgDQ3vc1MYTQWqDOClx0PoP_3tADD29raCs/edit#responses

← 제목 없는 설문지 파일 ☆ 드라이브에서 모든 변경사항이 저장되었습니다.

보내기 ⋮ c

질문 응답

응답 0개 + ⋮

응답받기

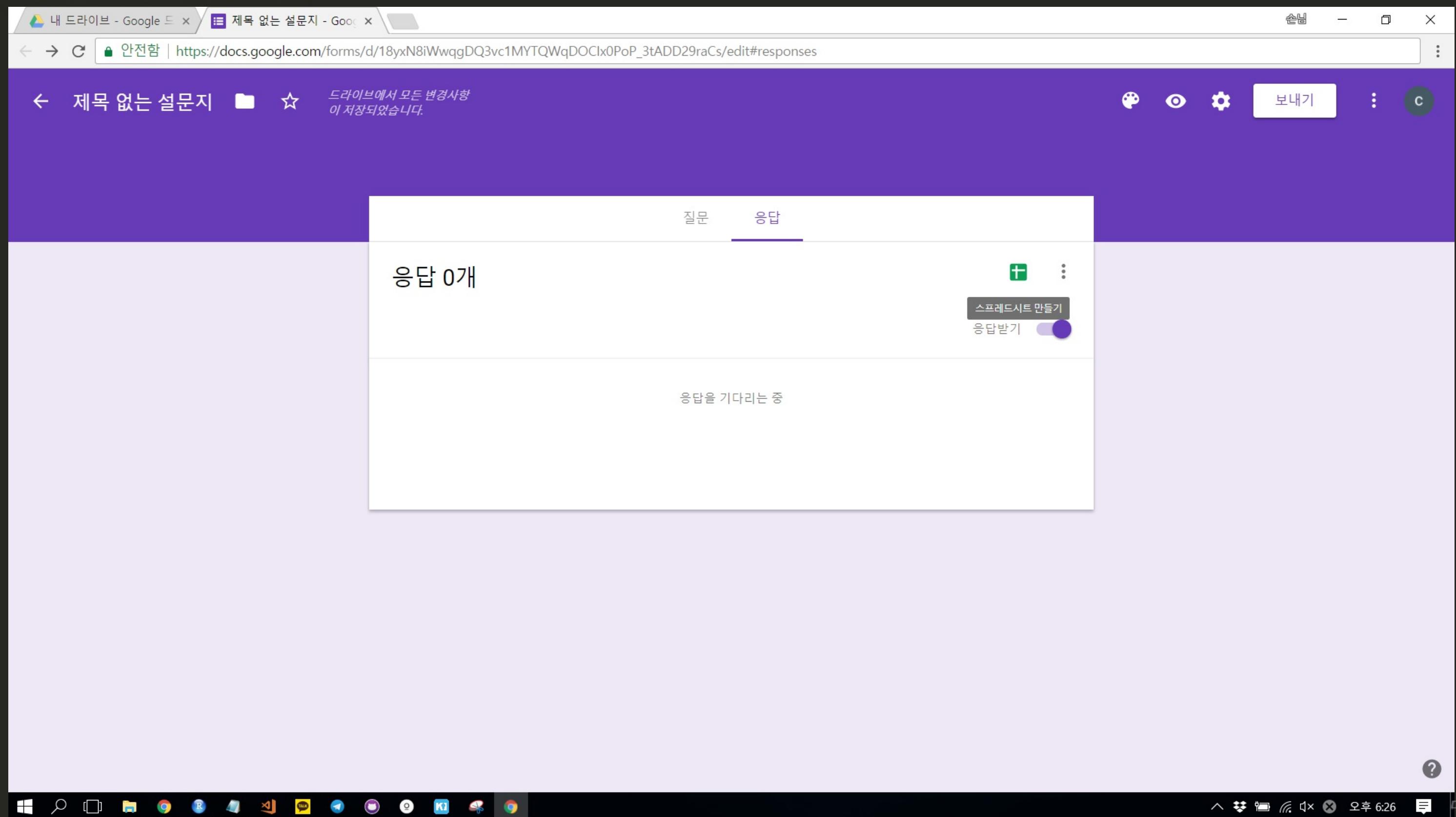
응답을 기다리는 중

?

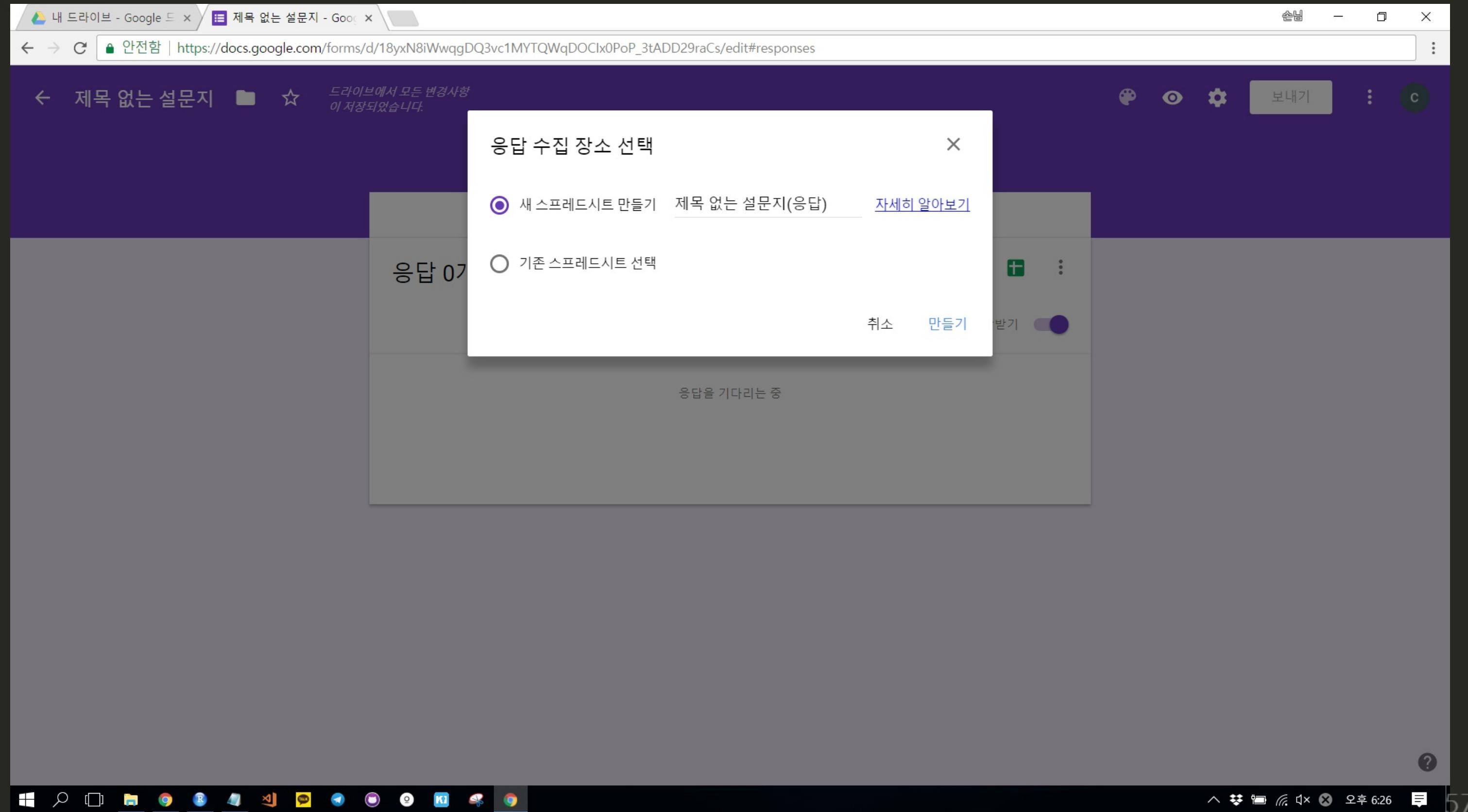
55 / 73

The screenshot shows a Google Forms survey page. At the top, there are two tabs: '질문' (Question) and '응답' (Response), with '응답' being the active tab. Below the tabs, it says '응답 0개'. There is a green '+' button and a three-dot menu icon. A toggle switch labeled '응답받기' (Accept responses) is turned on. Below this, it says '응답을 기다리는 중' (Waiting for responses). The browser's address bar shows the URL: https://docs.google.com/forms/d/18yxN8iWwqgDQ3vc1MYTQWqDOClx0PoP_3tADD29raCs/edit#responses. The title of the page is '제목 없는 설문지' (Untitled Survey). The operating system taskbar at the bottom shows various pinned icons and the date/time as '오늘 6:26'.

스프레드시트 만들기 클릭



새스프레드시트 만들기로 선택후 만들기



스프레드시트로 연결 완료

내 드라이브 - Google 드 x 제목 없는 설문지 - Google Sheets x

안전함 | https://docs.google.com/forms/d/18yxN8iWwqgDQ3vc1MYTQWqDOClx0PoP_3tADD29raCs/edit#responses

← 제목 없는 설문지 ⌂ ☆ 드라이브에서 모든 변경사항이 저장되었습니다.

보내기 ⋮ c

질문 응답

응답 0개

+ ⋮

응답받기

응답을 기다리는 중

스프레드시트에 연결

?

This screenshot shows a Google Forms survey page titled '제목 없는 설문지' (Untitled Survey). At the top, there are two tabs: '내 드라이브 - Google 드' and '제목 없는 설문지 - Google Sheets'. The URL in the address bar is https://docs.google.com/forms/d/18yxN8iWwqgDQ3vc1MYTQWqDOClx0PoP_3tADD29raCs/edit#responses. A message at the top right indicates that all changes have been saved to Drive. The main content area shows a single response slot with the text '응답 0개' (0 responses) and a green plus sign icon. Below it, a message says '응답을 기다리는 중' (Waiting for responses). On the right side of the response slot, there are three icons: a green plus sign, a three-dot menu, and a toggle switch labeled '응답받기' (Accept responses) which is turned on. At the bottom left, there is a button labeled '스프레드시트에 연결' (Connect to spreadsheet). The taskbar at the bottom of the screen shows various pinned icons, and the system tray on the right displays network, battery, and system status information.

설문지 파일과 스프레드시트 파일 확인

내 드라이브 - Google 드라이브 제목 없는 설문지 - Google 드라이브

안전함 | https://drive.google.com/drive/my-drive

Google 드라이브 드라이브 검색

새로 만들기 내 드라이브 ▾

파일 이름 ↑

내 드라이브

컴퓨터

공유 문서함

최근 문서함

Google 포토

중요

휴지통

백업

제목 없는 설문지...

제목 없는 설문지...

newsforv.tar

notifications.google.com의 응답을 기다리는 중...

59 / 73

폼 입력 url 확인 - /edit을 제외한 전체 url

The screenshot shows a Google Form in edit mode. The URL in the browser's address bar is https://docs.google.com/forms/u/0/d/18yxN8iWwqgDQ3vc1MYTQWqDOClx0PoP_3tADD29raCs/edit. The form title is "제목 없는 설문지". There is one question titled "제목없는 질문" with the option "옵션 1". The interface includes tabs for "질문" and "응답", and a sidebar with various editing icons.

내 드라이브 - Google 드라이브 - Google 설문지

https://docs.google.com/forms/u/0/d/18yxN8iWwqgDQ3vc1MYTQWqDOClx0PoP_3tADD29raCs/edit

← 제목 없는 설문지

질문 응답

제목 없는 설문지

설문지 설명

제목없는 질문

옵션 1

+

T

■

▶

=

?

폼 입력 url로 입력 함수 만들기

위에서 복사한 url(/edit은 제외)을 이용해서 입력하는 함수를 작성합니다. body의 list 내 데이터 이름은 설문 항목명이어야 합니다.

```
form <- "https://docs.google.com/forms/d/yyyyyyyyyyyyyyyyyyyyyyyyyyyy"  
ping <- googleformr::gformr(form)  
body<-list(question1=data1,  
            question2=data2,  
            question3=data3)  
ping(body)
```

폼 입력 url로 입력 함수 만들기

위에서 복사한 url(/edit은 제외)을 이용해서 입력하는 함수를 작성합니다. body의 list 내 데이터 이름은 설문 항목명이어야 합니다.

```
form <- "https://docs.google.com/forms/d/yyyyyyyyyyyyyyyyyyyyyyyy"  
ping <- googleformr::gformr(form)  
body<-list(question1=data1,  
            question2=data2,  
            question3=data3)  
ping(body)
```

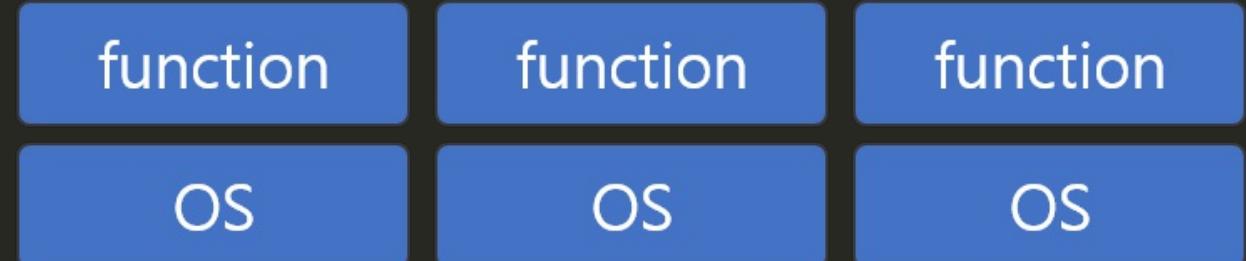
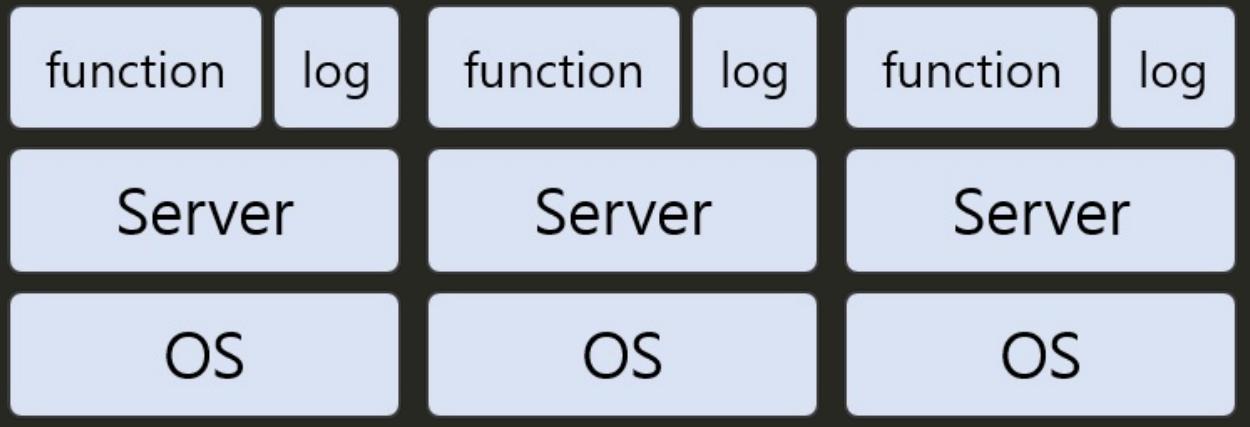
설문지 질문 이름을 파악하고 싶을 때

```
form <- "https://docs.google.com/forms/d/yyyyyyyyyyyyyyyyyyyyyyyy"  
form %>% get_form() %>% get_form_questions()
```

더 좋은 방법은 없을까

Micro Service Architecture

Function as a Service



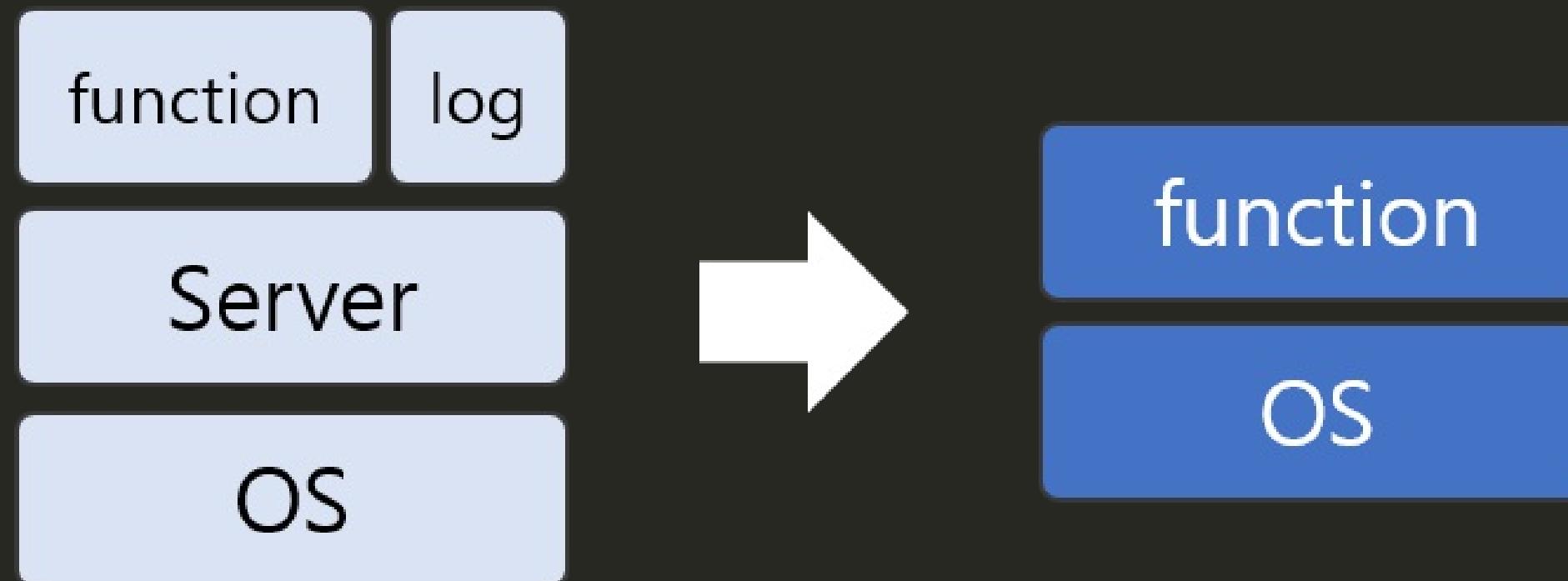
↔ HTTP 통신

↔ HTTP 통신

외부 요청

외부 요청

서버도 몰라도 됨



serverless로 R 함수를 api 서버로 만들자



OPENFAAS

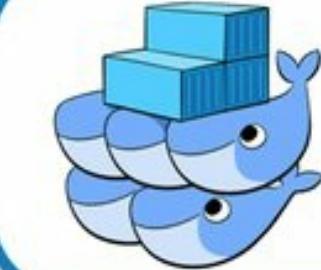
Functions as a Service

API Gateway

Function Watchdog



Prometheus



Swarm



Kubernetes



표준 입출력으로 함수 작성

```
## 표준 입력을 받아 line 객체로 저장
f <- file("stdin")
open(f)
line<-readLines(f, n=1, warn = FALSE)

## 데이터를 처리하여 result 객체로 저장
result<-paste0("Hi ", line)

## 표준 출력으로 결과 전달
write(result, stderr())
```

친절한 예시

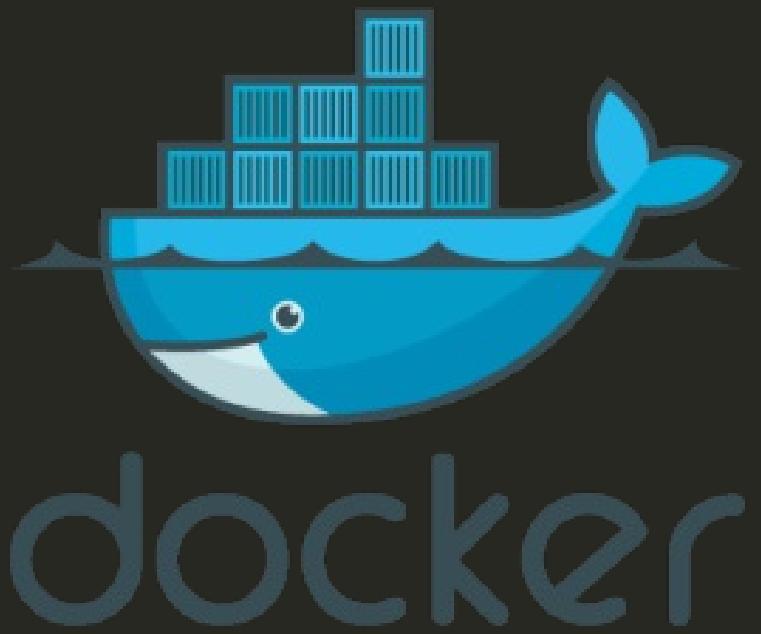
The screenshot shows a GitHub repository page for 'openfaas / faas'. The repository has 238 stars and 373 forks. The 'Code' tab is selected, showing the file structure: 'faas / sample-functions / BaseFunctions / R / handler.R'. The current branch is 'master'. A commit by 'mrchypark' was made on 1 Sep, adding an R sample function and README. The commit hash is 'fdb67e3'. The file 'handler.R' contains 8 lines of code:

```
1 #!/usr/bin/env Rscript
2
3 f <- file("stdin")
4 open(f)
5 line<-readLines(f, n=1, warn = FALSE)
6
7 write(paste0("Hi ", line), stderr())
```

서버는 몰라도

docker를 좀 알면 편합니다.

```
FROM artemklevtsov/r-alpine:latest  
  
ADD https://github.com/openfaas/faas/releases/download/0.6.1/fwatchdog /usr/bin  
RUN chmod +x /usr/bin/fwatchdog  
  
WORKDIR /root/  
  
COPY handler.R .  
  
ENV fprocess="Rscript handler.R"  
  
HEALTHCHECK --interval=1s CMD [ -e /tmp/.lock ] || exit 1  
  
CMD ["fwatchdog"]
```



는 시장입니다.

R로 api 서버를 만드는데 혼자 벅차다!

저한테 연락하시면 됩니다.



끝!

<https://mrchypark.github.io/MSCKR>

[pdf버전] [문의하기] [의견 및 오류 신고]