



我的 简历 PDF 可以在 这里 找到。

<https://github.com/01Joseph-Hwang10/01joseph-hwang10.github.io/raw/gh-pages/files/resume/resume-backend-agnostic.pdf>

姓名

邮箱: **joseph95501@gmail.com**

电话: **010-9049-9550**

网站: **<https://01joseph-hwang10.github.io>**

Github: **<https://github.com/01Joseph-Hwang10/>**

我是一个 后端 工程师。 我 擅长 后端 开发。

我 拥有 5 年 后端 开发 经验， 我 擅长 使用 各种 后端 技术 来 构建 可扩展 的 系统， 我 擅长 使用 各种 数据库 来 存储 数据， 我 擅长 使用 各种 中间件 来 提高 系统的 性能。

"我是一个 后端 工程师" 这个 说法 有点 笼统， 我 需要 更 具体 的 描述 来 说明 我的 技能 和 经验。 我 需要 更 具体 的 描述 来 说明 我的 技能 和 经验。



### Start-ups

我是一个 后端 工程师。 我 擅长 使用 各种 后端 技术 来 构建 可扩展 的 系统。 我 擅长 使用 各种 数据库 来 存储 数据。 我 擅长 使用 各种 中间件 来 提高 系统的 性能。



### Lean Programming

我是一个 后端 工程师。 我 擅长 使用 各种 后端 技术 来 构建 可扩展 的 系统。 我 擅长 使用 各种 数据库 来 存储 数据。 我 擅长 使用 各种 中间件 来 提高 系统的 性能。



### Code Aesthetics

我是一个 后端 工程师。 我 擅长 使用 各种 后端 技术 来 构建 可扩展 的 系统。 我 擅长 使用 各种 数据库 来 存储 数据。 我 擅长 使用 各种 中间件 来 提高 系统的 性能。

工作经历

#### 工作经历 (2021.03 - 至今)

- 后端 工程师 (全职)
- 后端 工程师 (兼职)

教育经历

## Joseph (@tripbuilder)

- 2021.07 - 2021.12
- 前端 / Front-end Developer
- AI 项目 前端 开发 项目 `TripBuilder` 项目

## Joseph (@labis)

- 2021.12 - 2022.11
- 前端 / Development Lead & Co-Founder
- 前端 开发 项目 开发 项目 `LearnQue` 项目

## Joseph23 (@shepherd23)

- 2022.07 - 现在
- 前端 / Product Manager & Software Engineer
- E-Commerce 项目 开发 项目 开发 项目 `PickHound` 项目

项目 项目



项目 项目 项目 项目 项目 项目.

<https://01joseph-hwang10.github.io>

## => 项目 项目 项目 Caching 项目 项目

项目 项目 项目

- 项目 项目 项目 `Firestore` 项目 项目 项目 项目 项目 项目 项目 项目 项目 项目.
  - 项目 项目 项目 项目 项目 项目 项目 `NestJS` 项目 `ActionProviderModule` 项目.
- 项目 项目 项目 `300 / 1000` -> `1500 / 1000` 项目
- => 项目 项目 项目 项目

## => 项目 项目 项目 项目 项目 项目 项目 项目 项目 项目

项目 项目 项目 项目

- 项目 项目 项目 项目 `API` 项目 项目 项目 项目 项目.
- `GCL` 项目 项目 `numpy`, `pandas`, `matplotlib` 项目 项目 项目 项目 项目, 项目 项目 项目 项目 项目 项目 项目.
- 项目 项目 项目 `API` 项目 项目 项目 项目, `Date.now()` 项目 项目 项目 项目 项目 项目.
- => `Firestore` 项目 `count` 项目 项目 项目 项目 项目, `count` 项目 项目 项目 项目 项目.

## => Leaky Bucket Policy 项目 API 项目 项目 项目 项目

项目 项目 项目

- 项目 项目 项目 `API` `Leaky Bucket Policy` 项目 100 20 项目 `API` 项目 项目.
- 项目 项目 项目 项目 项目 项目 项目 `API` 项目 `HTTP 429` 项目 项目
- => `LeakyBucketClient` 项目 项目 项目 项目 项目, `429` 项目 项目 项目 项目 项目 项目 项目.

## => 项目 项目 项目 Worker 项目 项目

👉 🏠 📄

- 容器化 する ことで Google Cloud Run に 容器化 可能。
  - 容器 実行 時に API Leaky Bucket Policy を 用い、 実行 時に DB に 接続 する ことで Cloud Run に 容器化 する こと 100% 可能 する こと。
- => GCE 容器化 する こと Cloud Load Balancer に 接続 する ことで Worker 容器化 可能 する こと。
  - Cloud Build Substitution Variable に Docker Build Args を 指定、 Cloud Run Worker 容器 実行 可能 する こと。

## => GCE instance group autoscaler と instance termination について 容器化 する こと

👉 🏠 📄

- GCE instance group autoscaler に scale down する こと instance に 接続 可能、 容器化 instance に 接続 可能 する こと。
- => NestJS に `OnApplicationShutdown` を 実装 する ことで 実行 時に 接続 可能 する こと。
  - Fire and forget 方式 request を Shutdown process 実行 可能 する こと。

## => 容器化 する こと 容器化 する こと

👉 🏠 📄

- Bandit Model に 接続 する こと HTTP 413 接続 可能 する こと。
- 容器 `np.float32`, `gzip` を 実行 可能 する こと、 容器化 可能 する こと。
- => 容器 実行 可能 10% 程度
- => 容器 実行 可能 2% 程度

## => 容器化 する こと 容器化 する こと

👉 🏠 📄

- Bandit 容器 実行 可能 する こと engines を 実行 可能 する こと。
- 容器 実行 可能 する こと blueprint を 実行 可能 する こと。
- 容器 実行 可能 する こと Redirect 実行 可能 する こと。
- => 容器 実行 可能 する こと 容器化 可能 する こと。

## => 容器化 する こと AI 容器化 する こと

👉 🏠 📄

- 容器化 する こと 容器化 可能 する こと engines を 実行 可能 する こと。
- => 容器 実行 可能 する こと AI 容器化 可能 する こと。
- => 容器 AI 容器 実行 可能 する こと 容器化 可能 する こと。

## => 容器化 する こと, 容器化 する こと 容器化 する こと

👉 🏠 📄

- Lerna, Python Module に 接続 Monorepo 方式 (Recommendation API, Bandit Engine, Banner Manager)
- Lerna, `standard-version` を 実行 可能 する こと (容器 実行 可能)
- 容器 実行 可能 する こと engines を 実行 可能 する こと (Recommendation API, Banner Manager)
  - 容器 実行 可能 する こと Github Actions, Google Artifact Registry に 接続 可能 する こと
  - => 容器 実行 可能 する こと 容器化 可能 する こと。

## => 容器化 する こと 容器化 する こと

👉 🏠 📄

- 容器化 CI/CD 容器化 Github Actions に `push` を 実行 可能 する こと。
- 容器化 実行 可能 する こと engines を 実行 可能 する こと (Recommendation API, Banner Manager)
- Lerna に 接続 可能 する こと engines を 実行 可能 する こと (Recommendation API, Banner Manager)
  - => 容器 実行 可能 する こと CI/CD 容器化 可能 する こと。

## => Jest, Pytest に 接続 TDD 容器化 する こと

👉 🏠 📄

- Jest, Pytest 등 단위 테스트 프레임워크를 Unit/E2E 테스트에 활용.
- Insomnia 등 API 테스트 도구를 사용하여 API 테스트를 자동화하고, 테스트 커버리지를 90% 이상으로 높임. (테스트 20% → TDD 테스트 20%)
- => 테스트 코드도 테스트를 위한 테스트 코드를 작성하여 테스트 커버리지를 80% 이상으로 높임. (테스트 4 ~ 5% → TDD 테스트 0 ~ 1%)

## => **API 테스트를 위한 Private File Server**



- API 테스트를 위한 Private File Server를 구축.
- [static-webpage-with-auth](#)를 사용하여 API 테스트를 위한 인증을 구현.

## => **Google Sheets를 활용한 데이터 관리**



- NestJS의 Django와 Admin Dashboard를 활용하여, Third Party 라이브러리 (e.g. AdminJS)와 Firestore의 DB를 사용하여 데이터 관리.
- Google Sheets, Sheets API, Cloud Functions를 사용하여 데이터 관리와 API를 연결.
- => UI를 사용하여 데이터 관리와 API를 연결.
- => 데이터 관리와 API를 연결하여 데이터 관리와 API를 연결.

## => **Cloud Logging & Slack을 활용한 모니터링**



- API 테스트를 위한 GCP 프로젝트를 생성하고, Cloud Logging을 설정.
- GCP의 Cloud Functions를 사용하여 API 테스트를 위한 Slack 알림을 전송.
- => API 테스트를 위한 GCP 프로젝트를 생성하고, Cloud Logging을 설정.
- 4XX, 5XX 등 API 테스트를 위한 에러 메시지를 Slack에 전송.
- GCP의 Logging Query Language를 사용하여 API 테스트를 위한 데이터를 조회.
- => API 테스트를 위한 데이터를 조회.



Typescript
HTML5
CSS3
ES6
CommonJS
Webpack
Babel
Rollup
Vite
Lerna

Python
Numpy
Pandas
Scikit Learn
Django
Flask
PyTest
Selenium
Sphinx
React

Redux
RTK Query
TailwindCSS
MUI
Emotion
Cypress
Storybook
StencilJS
D3

NextJS
Figma
NestJS
Express
Jest
Swagger
TypeORM
Typedoc
SQL
Firestore

Cloud Computing
Firebase
Cloud Run
Cloud Build
Cloud Storage
Cloud Logging

Cloud Compute Engine
Docker
Linux
Github
Git
Github
Slack
Notion

Korean (Native)
English (TOEFL iBT 98 @ 2023.02)

작성 일자: 2023년 8월 29일