



我的 简历 PDF 可以在 这里 查看。

<https://github.com/01Joseph-Hwang10/01joseph-hwang10.github.io/raw/gh-pages/static/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。 我的 技能 包括 后端 开发。

我的 技能 包括 后端 开发 的 各种 技术, 如 数据库, 网络, 安全, 等等。 我 也 对 前端 开发 有 一定 的 了解。 我 的 工作 经验 包括 在 大型 公司 和 初创 公司 工作。

"我是一个 后端 工程师" 这个 说法 可能 会 让 人 感到 困惑, 因为 后端 开发 的 范围 很 广。 我 的 技能 包括 数据库, 网络, 安全, 等等。 我 也 对 前端 开发 有 一定 的 了解。 我 的 工作 经验 包括 在 大型 公司 和 初创 公司 工作。



### Start-ups

我是一个 后端 工程师。 我的 技能 包括 数据库, 网络, 安全, 等等。 我 也 对 前端 开发 有 一定 的 了解。 我 的 工作 经验 包括 在 大型 公司 和 初创 公司 工作。



### Lean Programming

我是一个 后端 工程师。 我的 技能 包括 数据库, 网络, 安全, 等等。 我 也 对 前端 开发 有 一定 的 了解。 我 的 工作 经验 包括 在 大型 公司 和 初创 公司 工作。



### Code Aesthetics

我是一个 后端 工程师。 我的 技能 包括 数据库, 网络, 安全, 等等。 我 也 对 前端 开发 有 一定 的 了解。 我 的 工作 经验 包括 在 大型 公司 和 初创 公司 工作。

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 在 Google Cloud Run 部署 部署。
  - 容器 在 API Leaky Bucket Policy 下，在 容器 在 DB 容器 在 Cloud Run 容器 在 容器 1 容器 在 容器 在 容器。
- => GCE 容器 在 Cloud Load Balancer 容器 在 容器 在 Worker 容器 在 容器 在 容器。
  - Cloud Build Substitution Variable 在 Docker Build Args 容器，Cloud Run Worker 容器 在 容器 在 容器。

## => GCE instance group autoscaler → instance termination → → → → →



- GCE instance group autoscaler scale down instance group, delete instance group, delete instance group.
- => NestJS `OnApplicationShutdown` lifecycle hook.
  - Fire and forget request, Shutdown process, delete instance group.

=> □□ □□□□ □□□ □□ □□ □□



- Bandit Model은 1000000 번 HTTP 413 에러 발생하면 종료.
- `np.float32`, `gzip` 이 데이터셋을 불러오는 데, 데이터셋을 불러오는 데.
- => 이 데이터셋은 100 MB
- => 이 데이터셋은 20 MB

=> 1000 10000 100 10 1000 10 10



- **Bandit** ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ໃຊ້ ແອັບພລິເຄຊັນ **engines** ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແອັບພລິເຄຊັນ.
- ແມ່ນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ **blueprint** ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ.
- ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ **Redirect** ແມ່ນ ແອັບພລິເຄຊັນ.
- => ແມ່ນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແມ່ນ ແອັບພລິເຄຊັນ ແມ່ນ ແມ່ນ.

=> 如何 利用 AI 來 學習



- 000000 000 000 0 000000 000 000 000000 00 00 000, 00 00 000 000 000.
- => 00 000 000 000000 00 00000 00 0000 00 AI 00 00 000.
- => 00 AI 000 000 0 00 00 00 000 00 000 00.

=> 0000 00, 000 000 0 0000 0 00 00 00



- Lerna, Python Module `lerna` Monorepo `lerna` (Recommendation API, Bandit Engine, Banner Manager)
- Lerna, `standard-version` `standard-version` `standard-version` `standard-version` (CI Pipeline)
- `lerna` `lerna` `lerna` `lerna` `lerna` `lerna` (Recommendation API, Banner Manager)
  - `lerna` `lerna` `lerna` `lerna` `lerna` `lerna` Github Actions, Google Artifact Registry `lerna` `lerna` `lerna`
  - => `lerna` `lerna` `lerna` `lerna` `lerna` `lerna`.

=>                          



- 00000 CI/CD 00 00000 Github Actions push 0000 00 000.
- 00000 0000 000 000000 000000 00 0000 00 0000 000 0000 000 000.
- Lerna 000 000 000 00 000 0000000 0000 00 0000 000 00 0000000 000 000 00000 00 00000 000 00000 000.
- => 0000 00 000 0 CI/CD 00 00 000

## => Jest, Pytest TDD



- Jest, Pytest 등 단위 테스트 프레임워크를 Unit/E2E 테스트에 사용.
- Insomnia 등 API 테스트 도구를 사용하여 API 테스트를 자동화하고, 테스트 커버리지를 90% 이상으로 높인다. (API 20개 → TDD 테스트 2개)
- => 테스트 코드도 테스트를 받아 테스트 커버리지를 80% 이상으로 높인다. (API 4 ~ 5개 → TDD 테스트 0 ~ 1개)

## => API 테스트 프레임워크를 사용하여 Private File Server 테스트



- API 테스트를 위해 Private File Server 테스트를 작성.
- [static-webpage-with-auth](#) 등 API 테스트 도구를 사용하여 API 테스트를 자동화.

## => Google Sheets를 사용하여 API 테스트 자동화



- NestJS의 Django와 Admin Dashboard를 사용하여, Thrid Party 라이브러리 (e.g. AdminJS)와 Firestore의 DB를 사용하여 API 테스트를 자동화.
- Google Sheets, Sheets API, Cloud Functions를 사용하여 API 테스트를 자동화하고, 테스트 커버리지를 90% 이상으로 높인다.
- => UI 테스트를 위해 API 테스트를 자동화하고, 테스트 커버리지를 90% 이상으로 높인다.
- => API 테스트를 자동화하고, 테스트 커버리지를 90% 이상으로 높인다.

## => Cloud Logging & Slack을 사용하여 API 테스트 자동화



- API 테스트를 위해 GCP의 Cloud Logging을 사용하여 API 테스트를 자동화.
- GCL을 사용하여 Cloud Functions를 사용하여 API 테스트를 자동화하고, Slack을 사용하여 API 테스트를 자동화.
- => API 테스트를 위해 GCP의 Cloud Logging을 사용하여 API 테스트를 자동화.
- 4XX, 5XX 등 API 테스트를 자동화하고, 테스트 커버리지를 90% 이상으로 높인다.
- GCL의 Logging Query Language를 사용하여 API 테스트를 자동화하고, 테스트 커버리지를 90% 이상으로 높인다.
- => API 테스트를 자동화.



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)

API 테스트: 2023년 8월 29일