

# 우당탕탕 법인카드 자동화

기존에 올렸던 문서를 수정한 글입니다.

## 아무튼 완성한 Output

## ※ 주의사항

- 한번에 하나의 프로세스만 처리 가능합니다. 공짜 VM에 올려놔서, 여러 개 동시에 돌렸더니 메모리 폭발... (그래서 차례로 돌아가게 해놨습니다)
- 안드로이드만 가능합니다.
- 고용량의 이미지를 올리면 서버가 터질 수 있습니다. 이미지를 그대로 찍어서 올리면 되는데, 스크롤 캡쳐 기능으로 길게 찍어서 올리면 터짐...
- 이미지는 따로 자르지 않아도 됩니다. 그냥 그대로 올리면 됨 아니, 자르면 오히려 데이터 누락 생길수도
- 당연히 비고나 특이사항은 따로 작성합니다 그야... 이건 법카 앱 인식만 하니까...

https://github.com/BeaverHouse/payment-back

https://github.com/BeaverHouse/payment-ui

BE endpoint : <a href="http://130.162.149.32:8000/docs">http://130.162.149.32:8000/docs</a>

UI는 진짜 기능만 돌아가게 만들었으니 무시해 주세요

## 개발한 이유

법인카드 내역 앱 봐가면서 일일히 옮겨적기 귀찮아서(...)

## 사실 예전에 몇 번 실패했습니다

- 엑셀 다운로드가 안 되기 때문에 스크린샷에서 글씨를 인식해야 하는 상황
- Python으로 tesseract 깔아서 인식시켜 보려고 했으나, 한국어 인식률이 안 좋아서 포기

https://github.com/tesseract-ocr/tesseract

- 인식은 포기하고, 텍스트라도 포맷에 맞추면 자동으로 액셀을 생성하려고 openpyxl + Fastapi 이용한 간단한 앱을 AWS Lambda에 띄웠었습니다 하지만 문제가 여럿 있었는데.
  - AWS Lambda로 프록시 api 서버를 띄워 버리면
    파일 Response는 base64 인코딩이 되어 온다는 것

## AWS API Gateway: why post request body is encoded base64?

Am using API gateway to trigger a Lambda function, using Lambda Proxy integration, however the request body is encoded base64, why is that? and how can I change it to be JSON to stringify JSON?





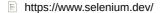
- 엑셀 파일 처리를 위해서 원본 엑셀 파일이 있어야 하는데,S3 쓸 게 아니라면 tmp 파일에 들어가야 하고, 가끔씩 못 읽을 때가 있음
- 그 외... 뭔진 모르겠는데 에러가... 아주 많음...
- 그래서 그냥 컴퓨터에서 로컬로 엑셀 파싱하는 코드만 돌리고 있었습니다

## 다시 하게 된 계기

• 2023년 4월 기준 프로젝트에서 열심히 잡무을 자동화하고 있는데, 이 과정에서 selenium을 제대로 쓸 수 있게 됨

#### Selenium

Selenium automates browsers. That's it!





• 한국어 인식률 좋은 OCR = 네이버 Clova AI 아래 페이지가 체험 페이지인데. 브라우저당 10회 제한이라고는 되어 있지만 사실상 localStorage 지우면 무한으로 사용 가능한 듯?했음

CLOVA OCR - 네이버 클로바

정확하고 빠른 텍스트 추출, CLOVA OCR을 체험해보세요.



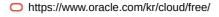
tttps://clova.ai/ocr/?lang=ko

- AWS Lambda는 아무리 생각해도 파일을 처리하는 앱을 띄우기에는 부적합했음

  - 그렇다면 VM에 띄우면 되잖아? 해서 Oracle VM에 띄우기로 함
    - Oracle 프리티어 계정 만들면 작은 VM 2개 무료로 띄울 수 있습니다. 이렇게 간단한 거 띄우기에는 좋음...

## 클라우드 서비스 무료 이용

Oracle Cloud Free Tier는 기업에게 무제한으로 사용할 수 있는 상 시 무료 클라우드 서비스를 제공합니다.





## Workflow

- 1. FastAPI로 File을 받는 api를 구축합니다.
- 2. 파일을 받으면, 이미지를 카드내역만 남도록 자르는 전처리 과정을 거치고. 따로 폴더에 저장을 합니다. (Pillow 사용)
- 3. 백그라운드 프로세스로 selenium 을 사용해 크롬을 띄우고, Clova 사이트에서 OCR을 수행합니다.
- 4. 처리된 텍스트를 파싱해서 엑셀을 작성합니다.

그 외에 state 확인하는 API 간단하게 만들어 주고, Oracle VM에 앱을 띄우고, (나중에 기회가 되면 작성...) 쉽게 쓸 수 있게 Flutter로 UI를 만들면 저렇게 됩니다. 이 쪽은 거의 기본 기능을 사용한 거라서 특별히 할 말은 없네요.

## 주요 수정사항

selenium 은 돌릴 때 메모리를 많이 먹습니다. 이건 어쩔 수 없는데...

driver를 돌릴 때마다 이를 제대로 종료해 주지 않으면 selenium 이 메모리 사용량을 계속 먹으면서 점점 느려지다가 결국에는 터집니다.

## Python 기준, selenium driver를 닫을 때

driver.close() 는 완전히 driver를 종료시키지 않습니다.

driver.quit() 쓰세요

## 개선 방안

사실 이 앱의 가장 큰 문제는 주의사항에서도 보다시피 프로세스를 한 번에 하나만 돌릴 수 있고, 고용량 이미지를리면 터지는 등 서버가 불안정한 것입니다.

## 이건 결국 작은 VM에서 메모리를 많이 먹는 크롬을 돌려서 생기는 일인데

- VM의 성능을 올린다= 프리 티어를 벗어난다 =
- Clova API를 사용하면 selenium 안 써도 됩니다.
  - = 월에 일정 건 넘어가면 비용 청구 = 💰

음... 안될 것 같습니다...?