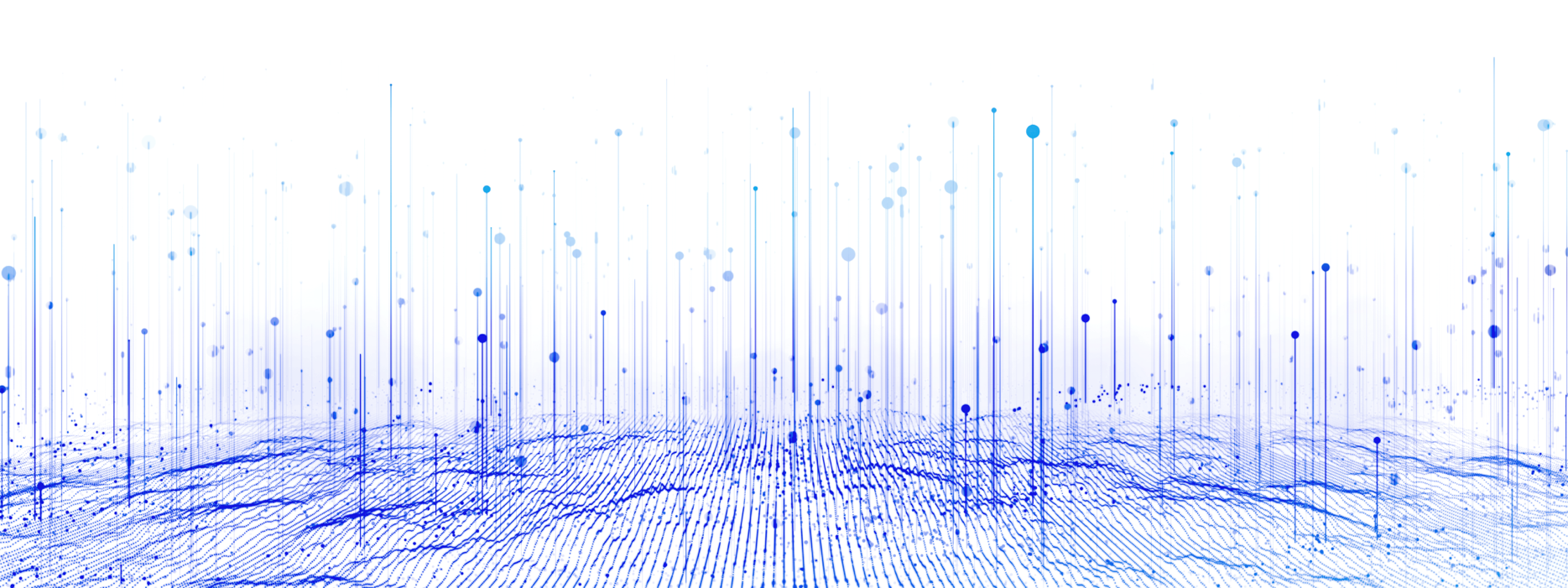
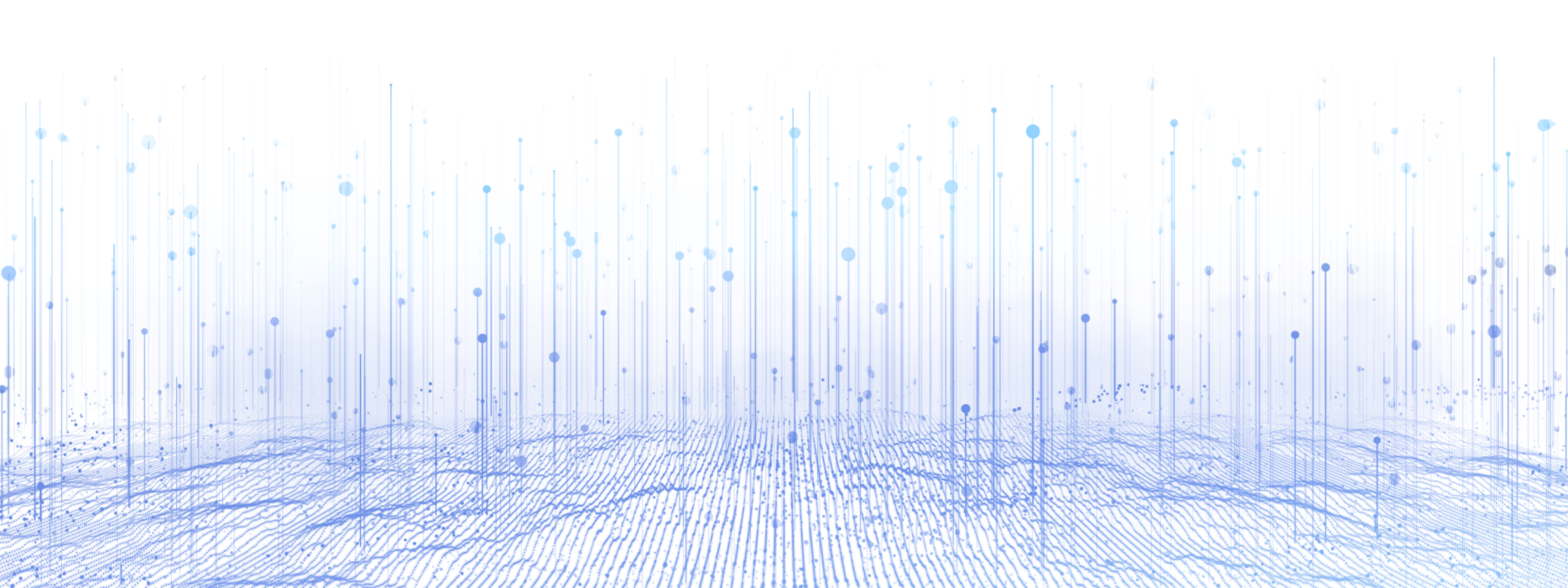


Whole Thing Project



JLK, Inc



**Outline**

[1. 통합환경 개발 및 계획 3](#_Toc43567562)

[1.1. 보고서의 목적 3](#_Toc43567563)

[1.2. 설계 현황 및 계획 3](#_Toc43567564)

[1.3. 통합 개발환경 요약 4](#_Toc43567565)

[1.4. 설치 과정 (개발자 참고자료) 5](#_Toc43567566)

[1.5. 프로그램 설계 요약 (개발자 참고자료) 9](#_Toc43567567)

[1.6. 프로그램 설계 상세 (개발자 참고자료) 10](#_Toc43567568)

[2. 서비스 프로세스 12](#_Toc43567569)

[2.1. 주민등록 등/초본 발급 13](#_Toc43567570)

[프로세스 개요 13](#_Toc43567571)

[프로세스 상세 13](#_Toc43567572)

[2.2. 홈텍스 사이트 스크래핑 15](#_Toc43567573)

[프로세스 개요 16](#_Toc43567574)

[프로세스 상세 16](#_Toc43567575)

[국세청 로그인 이슈 18](#_Toc43567576)

[3. 공인인증을 통한 보안 로그인 프로세스 개요 18](#_Toc43567577)

1. 통합환경 개발 및 계획
   1. 보고서의 목적

본 보고서는 현재 개발 중인 Whole Thing API에 개발에 대한 설계 현황 및 일정을 공유하기 위해 작성되었으며 추후 통합환경을 구성하는데 참고한다.

사전 데모버전을Restful API 방식으로 설계하였으며 해당 설계를 기준으로 이외 서비스를 제공할 수 있도록 확장하며, 개발 시 생긴 이슈를 공유한다.

* 1. 설계 현황 및 계획

|  |  |  |
| --- | --- | --- |
| 작업 구분 | 작업 기간 | 작업 현황 |
| 환경 구상 및 설계 | 2020-06-09 | 1. 환경구성에 대한 자료조사  2. 환경구성 드로잉 작업 |
| 로컬 환경구성 | 2020-06-10 ~ 2020-06-10 | 1. 설치 파일 다운로드  2. 폴더 스트럭쳐 |
| 로컬 데모파일 설계 및 제작 | 2020-06-10 ~ 2020-06-12 | 1. 기본신청 완료.  2. 각 조건에 따른 신청 완료  3. 예외처리 작업 완료  4. 로그 처리 작업 완료  5. 추후 개선 |
| 로컬 테스트 및 코드 최적화 | 2020-06-12 | 1. GIT 개설  2. 오류 수정 및 코드 최적화 |
| 공인인증 프로세스 기술조사, 데이터 송수신 포맷 개발 | 2020-06-15 ~ 2020-06-19 | 1. 공인인증 프로세스 기술조사  2. 데이터 송수신 포맷 개발  3. 데이터 로그 저장 포맷 및 테스팅 완료- |
| 공인인증 로그인 프로세스  개발인력 구인 | 2020-06-22 ~ 2020-06-25 | - |
| 스크래핑 엔진 개발 | 2020-06-29 ~ |  |
| 웹사이트 개발 | 2022-07-13 ~ |  |
|  |  |  |

* 1. 통합 개발환경 요약

|  |
| --- |
| Language |
| Python 3.8.2  Mysql ( Commuty Version, Engine : Inno DB , Character Encoding : UTF-8 )  Django |

|  |
| --- |
| Package |
| * 스크래핑   requests  beatifulsoup4  selenium   * 기타   추후 패키지 설치 참고 |

|  |
| --- |
| Issue |

\* 방안 우선순위

1) 오픈 API 방식 (오픈뱅킹, 정부 openAPI 등)

2) 스크린 스크래핑 (Beautiful Soup 4)

3) 동적 스크래핑 (Selenium)

|  |  |  |
| --- | --- | --- |
| 방식 | 장점 | 단점 |
| Open API | 빠른 속도  간편한 개발 | 일부 서비스에서는 협약 및 신청이 필요하며 비용이 발생 할 수 있음 |
| 스크린 스크래핑 | 빠른 속도 | 동적인 동작이 필요한 경우  개발 불가 |
| 동적 스크래핑 | 데이터 추출에 제약이 없음  (법적 이슈 제외) | 느린 속도 |

* 1. 설치 과정 (개발자 참고자료)

1. Python 설치 (3.8.2)

<https://www.python.org/downloads/>

1. 아나콘다 설치 (Python 3.7 64bit)

<https://www.anaconda.com/products/individual>

1. MYSQL 설치

https://dev.mysql.com/downloads/file/?id=495321

1. 패키지 및 프로젝트 설치

pip -m pip install --upgrade pip

pip install requests

pip install beatifulsoup4

pip install selenium

pip install django

pip install djangorestframework

pip install django-rest-swagger

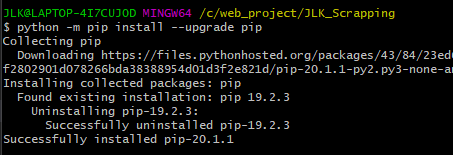
pip install pandas

pip install pyperclip

pip install pywin32

pip install -U pylint

pip install mysqlclient



1. Django & Mysql 설치 및 연동

참고자료 :

Django 설치 : <https://tothefullest08.github.io/django/2019/02/11/django01/>

MYSQL 연동 <https://catchsky.tistory.com/37>

1. Setting.py 설정

현재는 로컬을 바라보고 있으나 서버 개설 및 데이터 덤핑 후 운영 서버 DB 연동되도록 수정 예정.



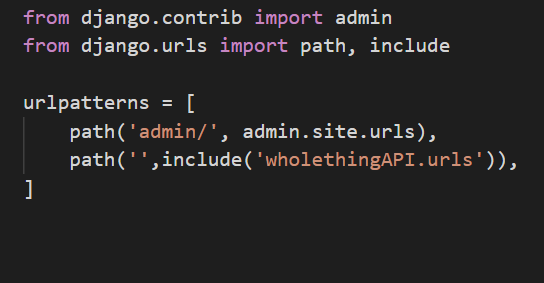
1. Rest API 연동 설정

API 연동 시 이슈 발생하면 아래 설정 확인 필요함.



1. Urls.py 설정

사이트 종류 추가 시 wholethingAPI/urls.py에 path 추가

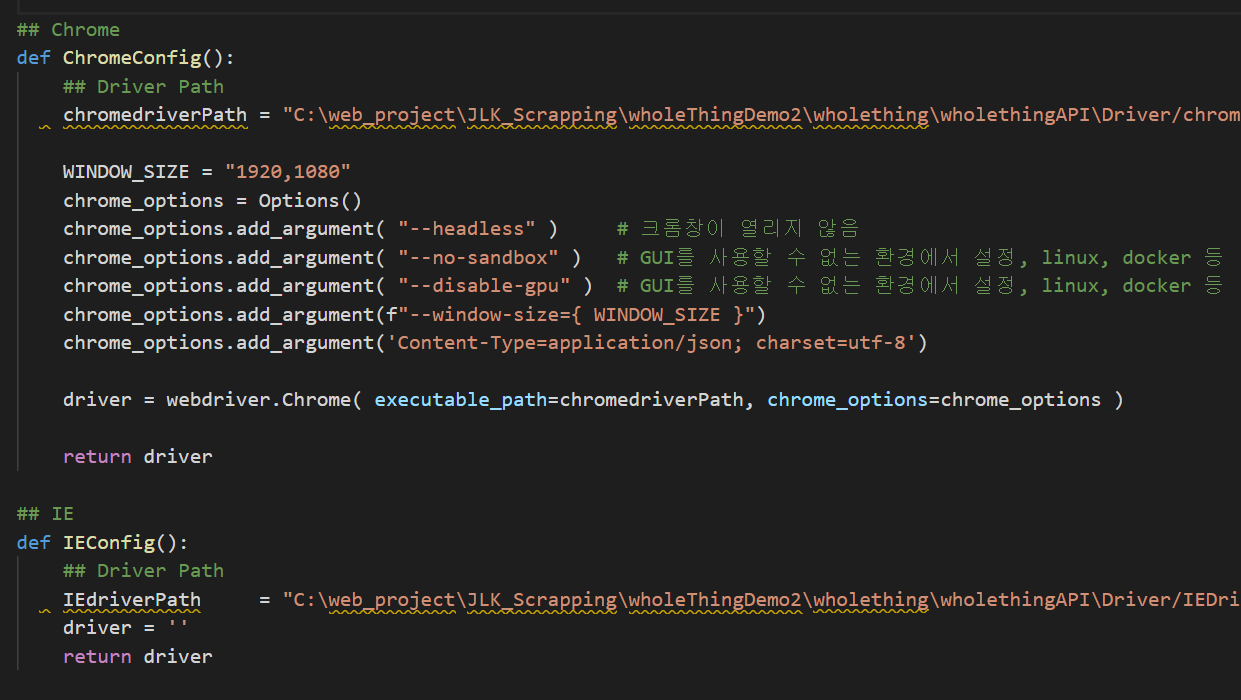


Wholething/urls.py wholethingAPI/urls.py

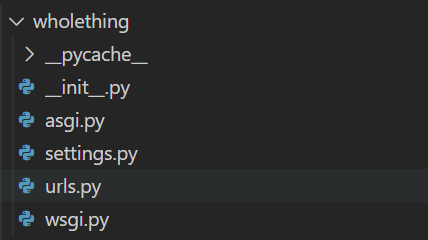
1. BrowserConfig.py 설정

현재 로컬을 기준으로 설정되었으며 각 환경에 맞게 경로 수정

각 브라우저 설정은 작업에 따라 확장

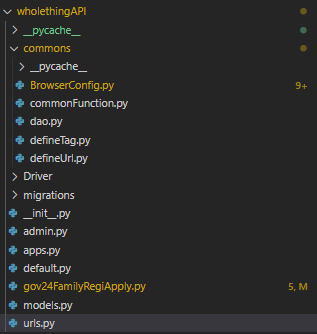


1. 폴더 구조



Setting.py :: DB 등 각 세팅파일

Urls.py :: wholethingAPI/urls.py로 랜딩



Urls.py :: URL 요청에 따라 각 파일 호출

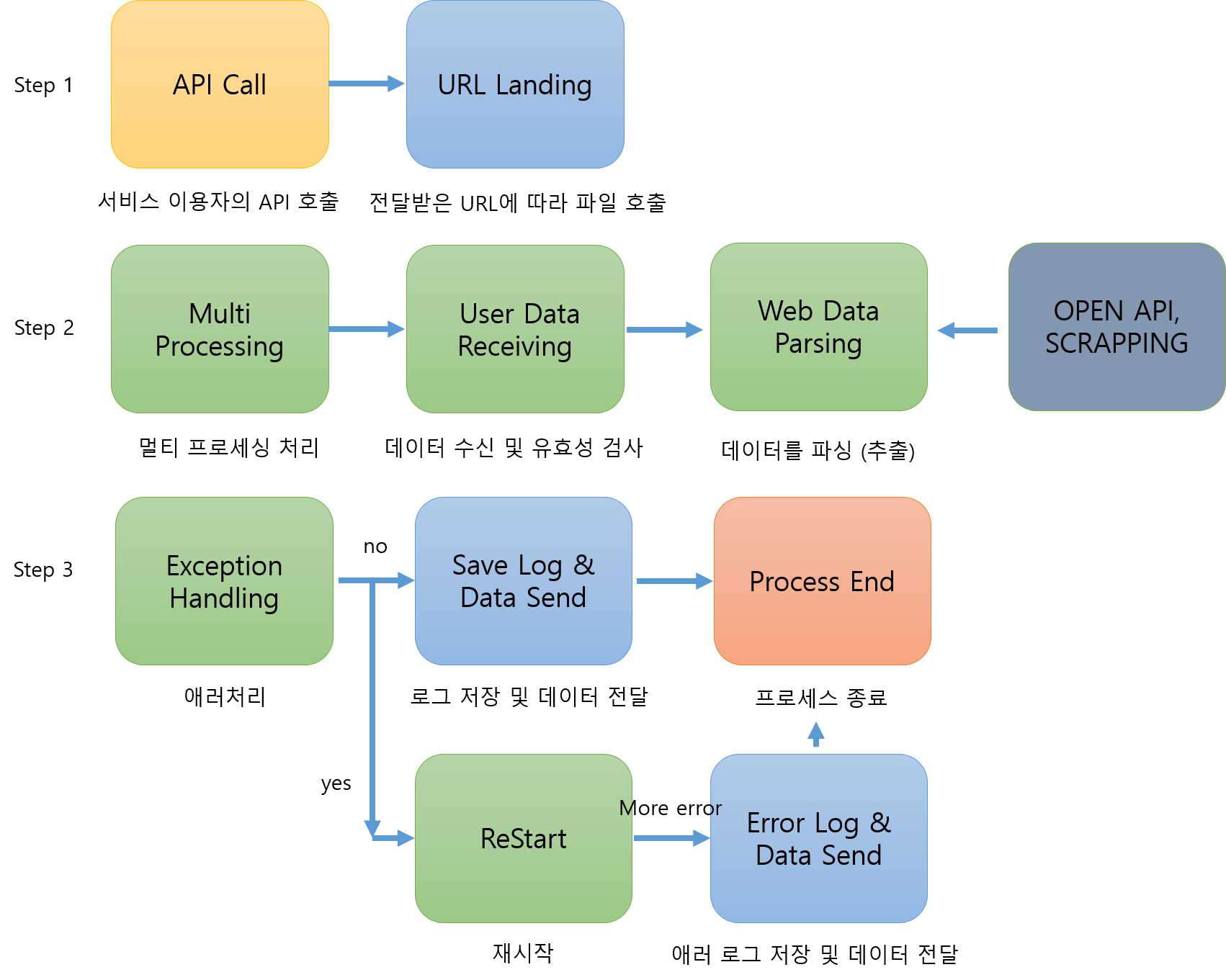
Commons / … :: 공통으로 사용되는 파일

dao.py :: Database 파일

BrowserConfig :: 브라우저별 환경 세팅

Driver :: 브라우저 접근을 위한 exe 파일

* 1. 프로그램 설계 요약 (개발자 참고자료)



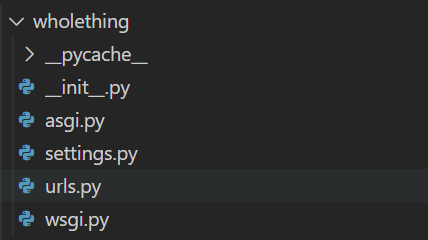
* 1. 프로그램 설계 상세 (개발자 참고자료)

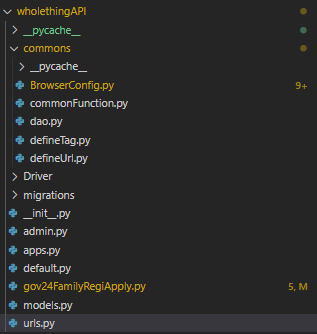
1. API 호출



1. URL 요청에 따라 서비스 파일 호출

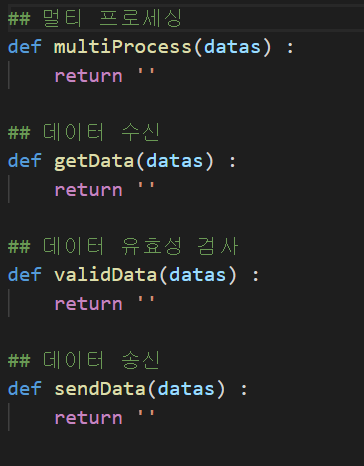
메인 프로세스 파일(gov24FamilRegiApply.py)은 각 urls.py에 의해 호출 됨





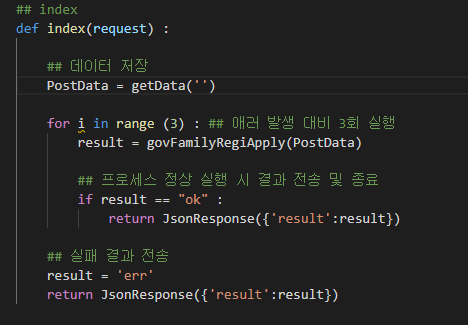
폴더 : wholething 폴더 : wholethingAPI

1. 멀티프로세싱 처리, 데이터 수신 처리, 데이터 유효성 검사 (추후 개발)



Commons/dataInout.py

1. 스크래핑, 데이터 파싱 등 프로세스 수행
2. 각 예외처리 실행 (함수 내부 Try Except)
3. 애러 발생 시 재시도 (현재 3회로 설정함)
4. 로그 저장
5. 결과 전달 및 종료



\*추후 변경 및 개선

1. 서비스 프로세스
   1. 주민등록 등/초본 발급

전달받은 유저의 정보를 토대로 주민등록등본을 자동으로 신청.



프로세스 개요

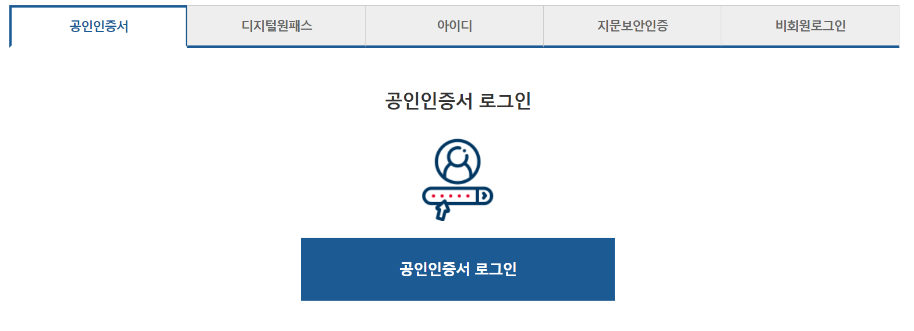
1. 유저의 아이디, 비밀번호, 공인인증서 비밀번호, 신청 세부사항을 전달 받는다
2. 해당 정보를 토대로 정부24 페이지에 로그인 한다.
3. 세부사항을 토대로 민원을 신청한다.
4. 신청 완료 후 인쇄 화면을 유저에게 제공한다.

프로세스 상세

실제 운영 시 브라우저창을 닫은 상태로 진행 됨.

1 . 로그인 페이지에 접근

* url : <https://www.gov.kr/nlogin/?regType=ctab>
* 해당페이지에 접근한 후 로딩이 완료되면 공인인증서 로그인 버튼을 클릭



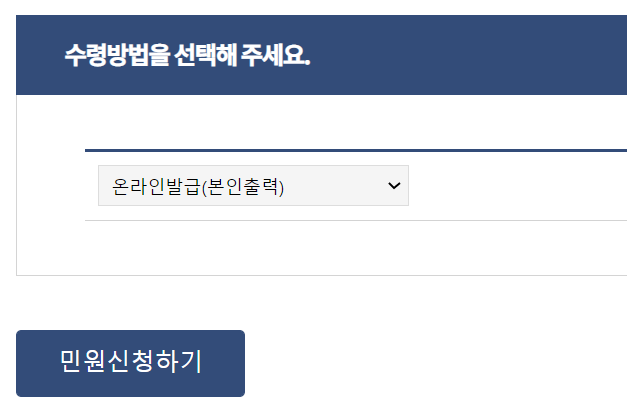
2 . 유저데이터 입력 및 로그인 실행

* 공인인증서 창에 유저 데이터를 입력하고, 전달받은 유저 정보로 로그인 실행



1. 세부사항 설정 및 신청

* 유저가 전달한 세부 사항에 따라 민원 신청
* 주소를 제외한 일반 발급이 다수일 것으로 예상되며 해당 기능 추가 예정



1. 결과 처리 및 종료

* 해당 페이지를 통해 정상 신청 여부를 판단하여 결과 전송



* 1. 홈텍스 사이트 스크래핑

홈텍스 사이트에 로그인하여 정보를 스크래핑하는 방안과 구현 중 발생한 이슈와 방향을 제시.



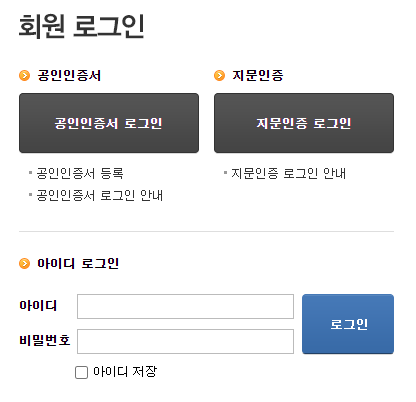
프로세스 개요

1. 유저의 아이디, 비밀번호, 공인인증서 비밀번호 및 유저 세부사항요청 정보를 전달받는다.
2. 해당 정보를 토대로 홈텍스 페이지에 로그인 한다.
3. 세부사항을 토대로 정보를 스크래핑 한다.
4. 스크래핑 완료 후 데이터를 취합, 정리하여 json 형식으로 전달한다.

프로세스 상세

1. 로그인 페이지에 접근

(<https://www.hometax.go.kr/websquare/websquare.wq?w2xPath=/ui/pp/index_pp.xml>)



1. 로그인 시도



1. 유저의 요청에 따라 필요 데이터 조회



국세청 로그인 이슈

국세청사이트 공인인증 로그인의 경우 정부24 페이지와 같은 동적 스크래핑 방식(매크로)으로 정상 로그인 되지 않았다. 원인을 파악 해본 결과 아래와 같이 확인되었다.

1. 페이지가 로딩된다.  
2. 공인인증서 로그인 클릭.  
3. 팝업창을 불러오는 함수를 호출.  
4. xpath나 태그를 검색하지만, 팝업 창이 페이지 로딩 된 이후 데이터를 불러오는 형식으로 되어 있기 때문에 프로그램 입장에서는 없는 요소(attribute)로 인식.

5. 애러발생.

이 문제점은 동적 스크래핑 방식이 아닌 정적 스크래핑 방식으로 공인인증 보안 로그인

이 구현되면 해결 될 것으로 확인된다. 정적 스크래핑 방식으로 로그인 하기 위해서는

아래와 같은 형식으로 로그인 처리가 완료되어야 한다.

**1. 네트워크** **패킷을 분석하여 공인인증서를 통한 보안 로그인 처리**

**2. 쿠키와 세션 데이터를 이용하여 로그인 유지** .

추가로 공인인증, 보안 관련 작업들은 상당한 R&D가 필요한 것으로 확인된다. 해당 부분

에 관한 기술을 보유하고 있는 있는 인력이나 업체를 구하는 동시에 해당 기술에 대한

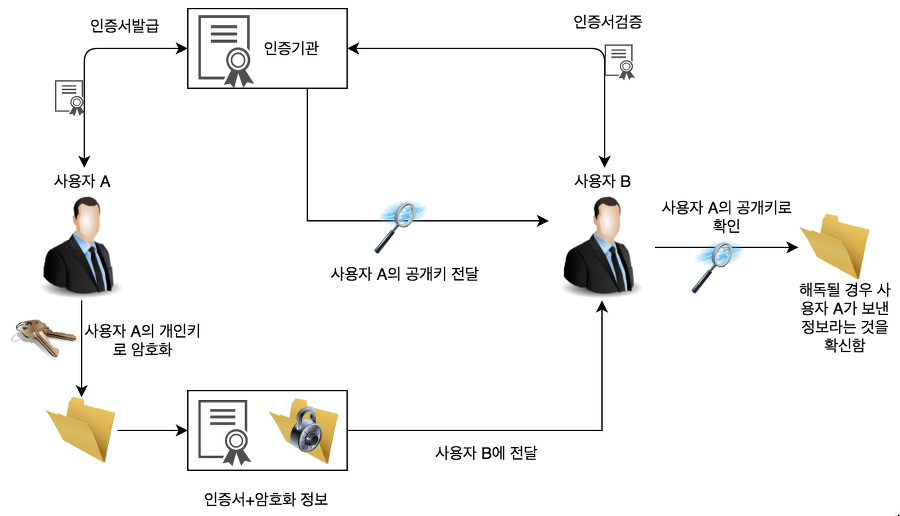
깊이 있는 기술자료를 조사하는 두 가지 방향으로 해결방안을 찾을 예정이다.

1. 공인인증서를 통한 보안 로그인

공인인증서를 통한 보안 로그인 프로세스 개요

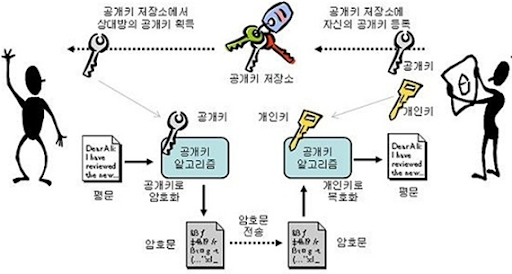
1. 전자서명 과정

해당 원리를 통해 공인인증 로그인 처리함



1. 공개키, 개인키 보안 처리

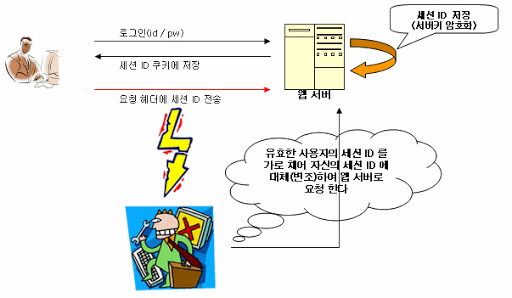
공개 된 키와 유저가 보유하고 있는 개인 키를 통해 유저를 확인한다. 공인인증의 경우 암, 복호화에 sha 256을 사용하는 것으로 확인됨



\*sha-256 : <https://ko.wikipedia.org/wiki/SHA>

1. 네트워크 패킷을 분석 및 세션 값 처리

분석된 네트워크 패킷에 맞게 세션 값을 전달, 유지하여 로그인 상태를 유지한다.



\*세션 : 서버에 임시로 저장되는 데이터로 브라우저가 종료되거나 일정 시간 이후 데이터가 소멸된다.

\*로그인 처리 : 임시데이터인 세션(혹은 쿠키)을 이용하여 로그인 처리한다. 공인인증 로그인의 경우 공개키와 개인키를 이용한 유저인증이 이뤄지며, 단순 세션을 가로채는 방법으로는 유저의 정보를 해킹할 수 없다.

TIP

wholething >> apiii

wholething >> apiiiAPI

요래바꿈 실행 될거임 패키지 메뉴얼 대로 다 깔고

아님 주요 파일만 복사하고 다시 깔던지