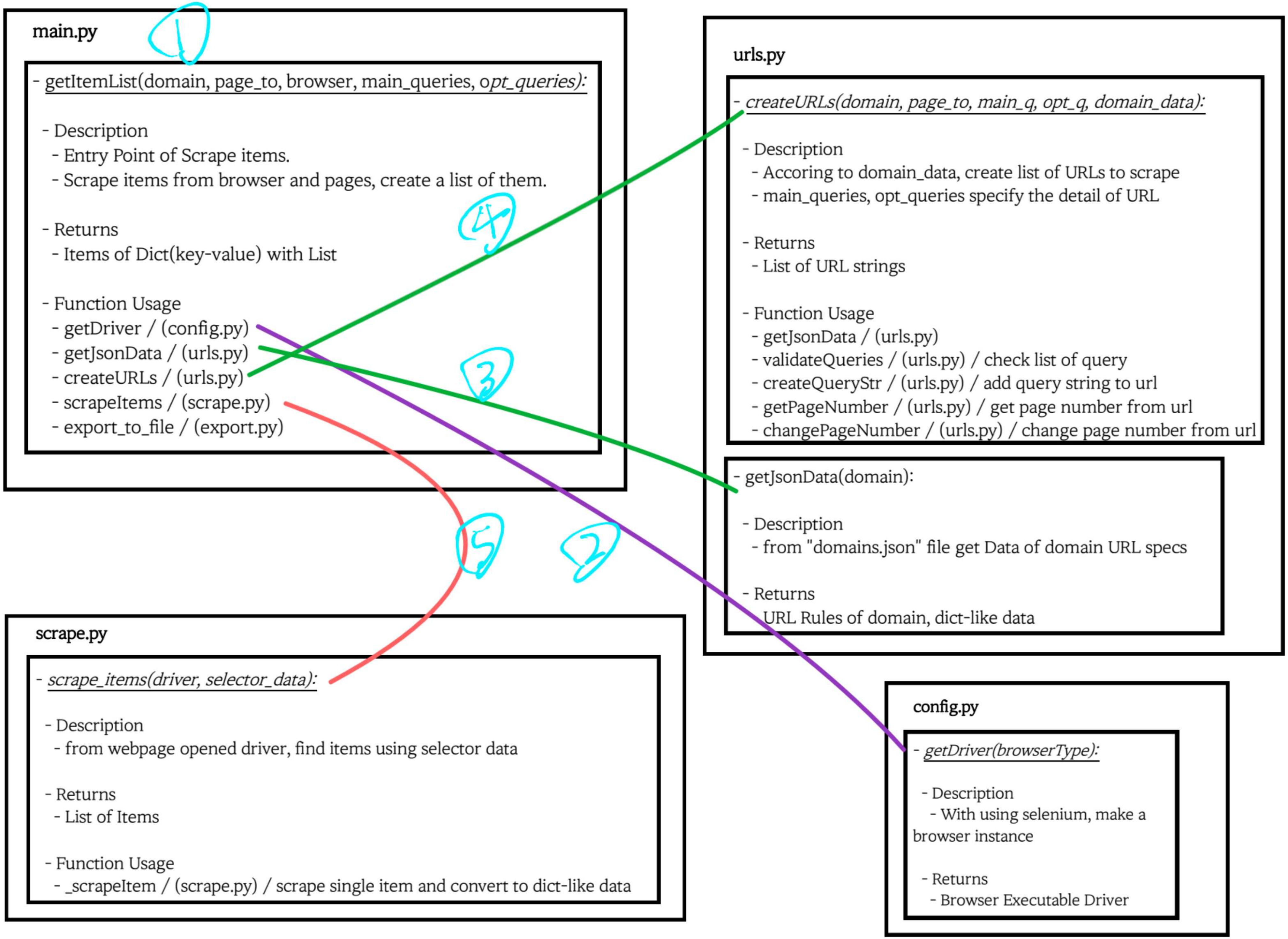

main.py

- getItemList(domain, page_to, browser, main_queries, o*pt_queries):*
  - Description
    - Entry Point of Scrape items.
    - Scrape items from browser and pages, create a list of them.
  - Returns
    - Items of Dict(key-value) with List
  - Function Usage
    - getDriver / (config.py)
    - getJsonData / (urls.py)
    - createURLs / (urls.py)
    - scrapeItems / (scrape.py)
    - export_to_file / (export.py)

urls.py

- *createURLs(domain, page_to, main_q, opt_q, domain_data):*
  - Description
    - Accoring to domain_data, create list of URLs to scrape
    - main_queries, opt_queries specify the detail of URL
  - Returns
    - List of URL strings
  - Function Usage
    - getJsonData / (urls.py)
    - validateQueries / (urls.py) / check list of query
    - createQueryStr / (urls.py) / add query string to url
    - getPageNumber / (urls.py) / get page number from url
    - changePageNumber / (urls.py) / change page number from url

- getJsonData(domain):
  - Description
    - from "domains.json" file get Data of domain URL specs
  - Returns
    - URL Rules of domain, dict-like data

scrape.py

- *scrape_items(driver, selector_data):*
  - Description
    - from webpage opened driver, find items using selector data
  - Returns
    - List of Items
  - Function Usage
    - _scrapeItem / (scrape.py) / scrape single item and convert to dict-like data

config.py

- *getDriver(browserType):*
  - Description
    - With using selenium, make a browser instance
  - Returns
    - Browser Executable Driver