<데이터 전달양식>

**플랫폼**

**웹사이트----4)**

<전체 개발순서>

1) 크롤링을 이용하여 데이터수집

2) 수집한 데이터를 가지고 통계적 분석 모델 또는 머신러닝

3) 분석모델을 다른곳에 적용할 수 있도록 **json파일**을 이용하여 데이터 저장

|  |
| --- |
| 1. **JSON(JavaScript Object Notation)** : Open API 대부분 JSON을 활용하여 데이터를 주고 받음.  1) JSON은 경량(Lightweight)의 data 교환 형식  2) Javascript에서 객체를 만들 때 사용하는 표현식을 의미  3) JSON 표현식은 사람과 기계 모두 이해하기 쉬우며 용량이 작아서, 최근에는 JSON이 XML을 대체해서 데이터 전송에 많이 사용됨  4) 특정 언어에 종속되지 않으며, 대부분의 프로그래밍 언어에서 JSON 포맷의 데이터를 다룰 수 있는 라이브러리를 제공  2. **JSON 형식**  1) name-value 형식의 쌍(pair)  - 여러 가지 언어들에서 object, hashtable, struct로 실현  - {String key : String Value}    2) 값들의 순서화된 리스트 형식  - 여러 가지 언어들에서 배열, 리스트로 실현  - [value1, value2, …]    3. **JSON Library - Jackson(Java) 적용하기**  - JSON은 쉽게 객체(Obejct)로 바꿔주거나 객체(Object)를 JSON으로 바꿔주는 Library가 대부분 언어에서 존재  - Jackson : https://github.com/FasterXML/jackson  4. **JSON 가져오기**  - JSON을 가져오기 위해서는, XMLHttpRequest로 불리는 API를 사용하면 된다.  1) 검색하려는 JOSN의 url을 변수에 저장 (JavaScript 코드 하단에 다음을 추가하면 됨)  var requestURL = 'https://mdn.github.io/learningarea/javascript/oojs/json/superheroes.json';  2) 요청을 만드려면 new 키워드를 사용하여 XMLHttpRequest 생성자에서 새 요청 객체 인스턴스를 생성 (그 다음 줄 아래에 다음을 추가하면 됨)  var request =newXMLHttpRequest();  3) 이제 open() 메소드를 사용하여 새 요청 열기 (그 다음 줄 아래에 다음을 추가하면 됨)  request.open('GET', requestURL);  4) reponseType을 JSON으로 설정하여 XHR이 서버가 JSON을 반환할것임을 알려주고, send() 메소드를 사용하여 요청  request.responseType ='json';  request.send();  5) 서버에서 응답이 반환되기를 기다렸다가 처리  request.onload=function(){  var superHeroes = request.response;populateHeader(superHeroes);  showHeroes(superHeroes);  } |

4) 수집한 결과를 json파일에 적용, **rest api**를 호출하여 이를 보여주는 페이지를 이용하여 시각화.

: 서버를 오픈해두고, 다른 곳에서 서버에 request를 보내면 데이터를 처리해서 그 결과를 보내줌.

Response 시 json형식으로 미리 정의해둔 것에 맞춰서 돌림.

|  |
| --- |
| - Rest API (Representational State Transfer) : HTTP URI (Uniform Resource Identifier)를 통해 자원(Recourse)를 명시, HTTP Method(POST,GET,PUT,DELETE)를 통해 해당 자원에 대한 CRUD Operation을 적용하는 통신방식  - 자원 : 해당 소프트웨어가 관리하는 모든 것 (문서, 그림, 데이터, 해당 소프트웨어 자체)  - 상태(정보) 전달 : 데이터가 요청되어지는 시점에서 자원의 상태(정보)를 전달 -> JSON or XML을 통해 주고받음    <Rest API의 장점>  - 웹의 기존 기술과 HTTP 프로토콜을 그대로 활용 -> 웹의 장점을 최대한 활용할 수 있는 아키텍쳐 스타일  - HTTP 표준 프로토콜에 따르는 모든 플랫폼에서 사용 가능  - 서버와 클라이언트의 역할을 명확히 분리 |

5) 수집한 결과를 json파일에 적용, **rest api**로 오픈하여 다른 곳에서 사용할 수 있도록 제공

