

FHIR 클라이언트의 서버 인터페이스 구현

진주은, 정성윤, 권혁주, 오채은, 이정훈
제주대학교 데이터사이언스학과

{jueunjin365, tjddbswkd1, gurwn917, codms8916}@naver.com, jhlee@jejunu.ac.kr

Sever interface implementation for FHIR clients

Jueun Jin, Sungyoon Jung, Hyukju Kwon, Chaeun Oh, Junghoon Lee
Dept. of Data Science
Jeju National University

요 약

본 논문에서는 Google Cloud Framework에서 저장되어 관련되는 FHIR 자원들을 조회, 검색, 생성하는 클라이언트를 개발한다. 구현된 응용은 RESTful API를 통해 클라우드 데이터에 접근하기 위해 파이썬의 request 라이브러리 함수를 호출하며, G-Cloud를 설치하여 세션 시작마다 구글 클라우드로부터 접근 키를 할당받는다. FHIR 기반의 의료 데이터는 JSON 형태를 가지며 클라우드 외부에서 각 필드의 중요도와 같은 추가적인 정보를 통해 목적에 맞는 응용을 개발할 수 있다.

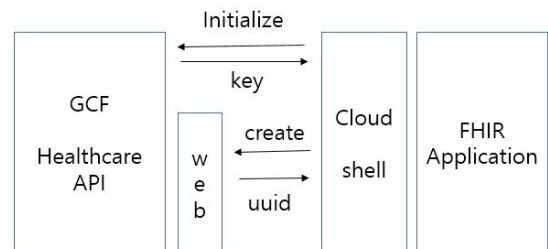
1. 서론

의료정보 표준인 HL7 FHIR (Fast Healthcare Interoperability Resource)가 의해 미국을 중심으로 표준 의료정보의 활용이 꾸준히 확산되고 있는 가운데[1] 구글의 Healthcare API와 아마존의 AWS FHIR Works과 같은 유명한 클라우드들도 FHIR 데이터 모델을 제공하고 있다[2]. 이 클라우드 서비스는 FHIR에서 정의된 각 자원의 속성에 따라 의료 데이터를 저장하며 클라우드 내에서 데이터를 저장하는 방식에 대해 외부에서는 인지할 필요 없이 표준화된 접근방식에 따라 응용을 작성하거나 데이터를 교환할 수 있다.

FHIR 자원은 환자, 검사, 내원, 진단 등 다양한 의료 행위를 기술하고 있으며 데이터 교환을 위해 JSON 포맷을 준용한다. 또 서버와 클라이언트 간 데이터 접근에 있어서는 Restful API를 지원하기 때문에 클라이언트는 get, post, put, delete 등의 요청을 서버에 전달하며 필요한 인자들을 설정하여야 한다. JSON 포맷은 파이썬의 딕셔너리와 자연스럽게 매핑될 뿐 아니라 파이썬은 requests 라이브러리를 통해 웹에 대한 다양한 접근을 허용한다. 더욱이 GUI나 데이터 분석 라이브러리를 결합할 수 있기 때문에 FHIR 클라이언트는 파이썬 언어로 개발될 수 있다.

2. FHIR 클라이언트의 수행 구조

<그림 1>은 GCF (Google Cloud Framework)와 이의 클라이언트 응용의 연동 과정을 보이고 있다. 일반적으로 클라우드 내에서의 기능을 수행하기 위해서는 클라우드에 로그인하여 내부에서 제공하는 기능들을 선택할 수도 있지만, 외부 응용 개발을 위해서는 G-cloud를 통해 cloud shell을 로컬 PC에 설치한다. 이 cloud shell이 수행을 시작할 때마다 GCF에게 초기화 요청을 보내며 GCF는 새로운 접근 키를 발급한다. 이후 이 키를 HTTP 요청 헤더에 명시하여야 한다. 또 주의할 사항은 GCF에서 새로운 자원을 생성하면 자동으로 id가 생성되어 클라이언트에 리턴되는 효율적인 응용 개발을 위해서는 클라이언트는 이를 저장하고 있어야 한다.



<그림 1> FHIR 클라이언트와 서버 연동

이 과정에서 cloud shell이 초기화될 때 새로 생성된 키를 확인하는 방법은 다음의 셸 명령을 수행하는 것이며 여기서 출력되는 키 값을 복사하여 파이썬

응용 프로그램에서 활용할 수 있어야 한다.

```
gcloud auth print-access-token
```

다음은 HTTP 메소드 호출을 위해 문자열 변수들을 미리 지정하여 두어야 하는데 <리스트 1>에서는 위의 셸 명령을 통해 얻은 키를 저장하고 구글 클라우드에서 생성한 데이터셋을 지정하는 URL을 변수에 저장하여 놓는다. 이때 접근키는 웹 요청의 헤더에 명시되어 있어야 하며 헤더는 Authorization 속성에 키를 포함하고 fhir+json 타입을 포함하도록 설정한다.

<리스트 1> 웹메소드 호출을 위한 기본 구조

```
key='ya29.a0AXeO80QmZ82kxp8xjnlAepWIPpMno1SgwybcinVnrqR0CgfAarndCcJ92iK-S9-KF5c_6l69BysHwAcP5hRlf7yWHntslSqhtojm_11HIZAGiYtYC_mJqwBA0z2E3LfToZF6mv2X4-XnBlds0c5RFicCLkcTnkgRDNjoQBnULrPhVITfaCgYKAa0SARASFQHGx2MiW4nftzyiq1-7M2CSi2C15Q0183'
```

```
hh = {'Authorization': 'Bearer '+key,
      'Content-Type': 'application/fhir+json'}
```

```
url = 'https://healthcare.googleapis.com/v1beta1/projects/fhirfhir/locations/asia-northeast3/datasets/FirstFHIR/fhirStores/FHIRStorage/fhir/'
```

이와 같이 설정되면 파이썬의 requests 라이브러리들을 통해 구글 클라우드에게 요청을 전송할 수 있는데 <리스트 2>는 이미 저장되어 있는 환자 자원에 대한 조회를 요청하는 코드이다. 여기서 id에는 대상 환자 자원의 아이디 값이 들어가 있다고 가정한다. 결국 대상 URL은 restful API의 형태를 따르게 된다. 호출이 완료되면 res.text에는 조회된 자원이 JSON 형태로 리턴되며 이를 딕셔너리를 변환하여 추가적인 기능을 구현할 수 있다.

<리스트 2> Get 메소드 호출에 의한 자원 조회

```
res = requests.get(url+resource+'/' +id, headers=hh)
print(res.text)
```

구글 클라우드에 저장된 자원을 검색하기 위해서는 검색 조건을 명시한 URL을 생성하면 되는데 <리스트 3>은 그 예를 보이고 있다. 성이 Oh인 환자들을 찾는 요청이며 이 결과들은 bundle의 형태로 리턴되어 복수의 자원들을 포함한다.

<리스트 3> FHIR 자원의 검색

```
res = requests.get(url+resource+'?family:exact=Oh',
                  headers=hh)
```

<리스트 4>는 post 메소드에 의해 자원들을 클라우드 내에 생성하는 코드인데 kkbundle.json에 복수의 자원들이 JSON 형태로 저장되어 있다. 물론 코드 내에서 딕셔너리로 저장된 자원들도 바로 HTTP 요청의 인자로 활용될 수 있다.

<리스트 4> Post에 의한 FHIR 객체 생성

```
with open('kkbundle.json', 'r') as f:      # Bundle
    jj = json.load(f)

url = url[:-1]
res = requests.post(url, data=json.dumps(jj), headers=hh)
rr = json.loads(res.text)
for e in rr['entry']:
    tt = e["response"]["location"].split('/')
    print(tt[14])
```

위의 코드를 수행하면 자원들이 생성되고 추후의 접근을 위해 id를 리턴한다. 추가적인 응용 개발을 위해서는 아래 표와 같이 id들을 클라이언트 내부에 저장하고 있어야 하며 특히 검사 값을 갖는 Observation 자원에 대해서는 타임스탬프와 같은 속성도 추가적으로 보유하는 것이 필요하다. 이에 의해 필요한 자원들을 직접적으로 조회, 삭제할 수 있다.

t1	observation	6cee3ed0-3cc9-4854-ad6c-ac8b65b28959
t2	observation	d720a3b1-381c-4228-9840-62eea224fca3
p1	patient	c181df1b-8fd6-407f-822d-5478170ccc56

3. 결론 및 추후 과제

구글 등 클라우드의 지원과 함께 FHIR 표준의 도입은 더욱 촉진될 것으로 예상되며 이를 기반으로 한 웹 클라이언트의 개발이 활성화될 수 있다. 더욱이 표준에는 사용자 정의 필드들을 포함하고 있는데 측정값의 긴급성 등을 추가하여 클라이언트 기능을 수행하는데 필요한 검색도 가능하다.

Acknowledgment

본 결과물은 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 3단계 산학협력 선도대학 육성사업(LINC3.0)의 연구결과입니다.

참고문헌

- [1] <https://www.hl7.org/fhir/>
- [2] Google Cloud Platform Healthcare API, available at <https://cloud.google.com/healthcare-api/docs/>