

HTTP 기초

REST API, RESTful Architecture

REST API?

The screenshot shows the Kakao Developers website with the 'REST API 도구' (REST API Tools) section selected. On the left, there's a sidebar with a navigation menu:

- iOS 개발가이드
- iOS 레퍼런스
- Android 개발가이드
- Android 레퍼런스
- JavaScript 개발가이드
- JavaScript 레퍼런스
- REST API 개발가이드
- 커스텀링크 생성가이드
- 소셜 플러그인
- REST API 도구
- Cache 삭제 도구
- SDK 다운로드

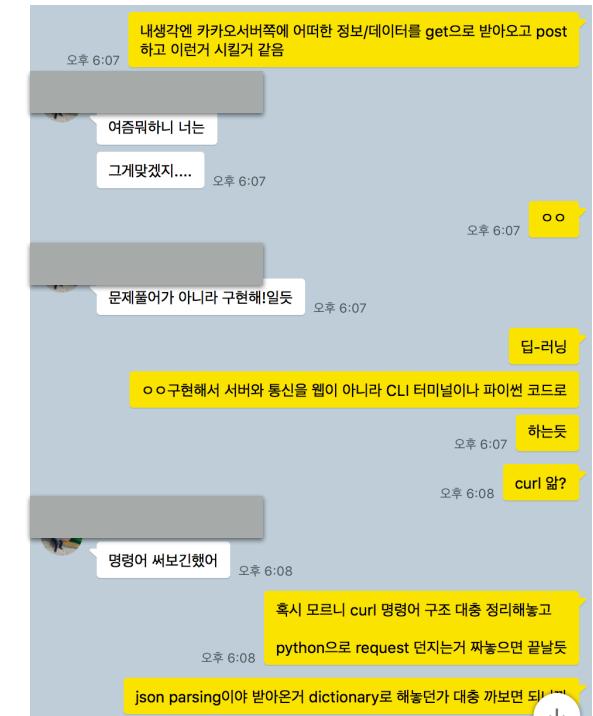
The main content area is titled 'REST API 도구'. It includes a note about logging in with Kakao Account for testing. There are input fields for 'developers-sample' and 'Access token'. A prominent yellow button labeled '카카오계정으로 로그인' (Log in with Kakao Account) is highlighted. Below these, there's a 'REST API Test' section with tabs for '사용자 관리', '카카오스토리', '카카오톡', '푸시 알림', '검색', '로컬', and '비전'. A series of API endpoints are listed:

- POST /v1/user/signup (앱 연결)
- POST /v1/user/unlink (앱 연결 해제)
- POST /v1/user/logout (로그아웃)
- GET POST /v2/user/me (사용자 정보 요청)
- POST /v1/user/update_profile (사용자 정보 저장)
- GET /v1/user/ids (사용자 리스트 요청)

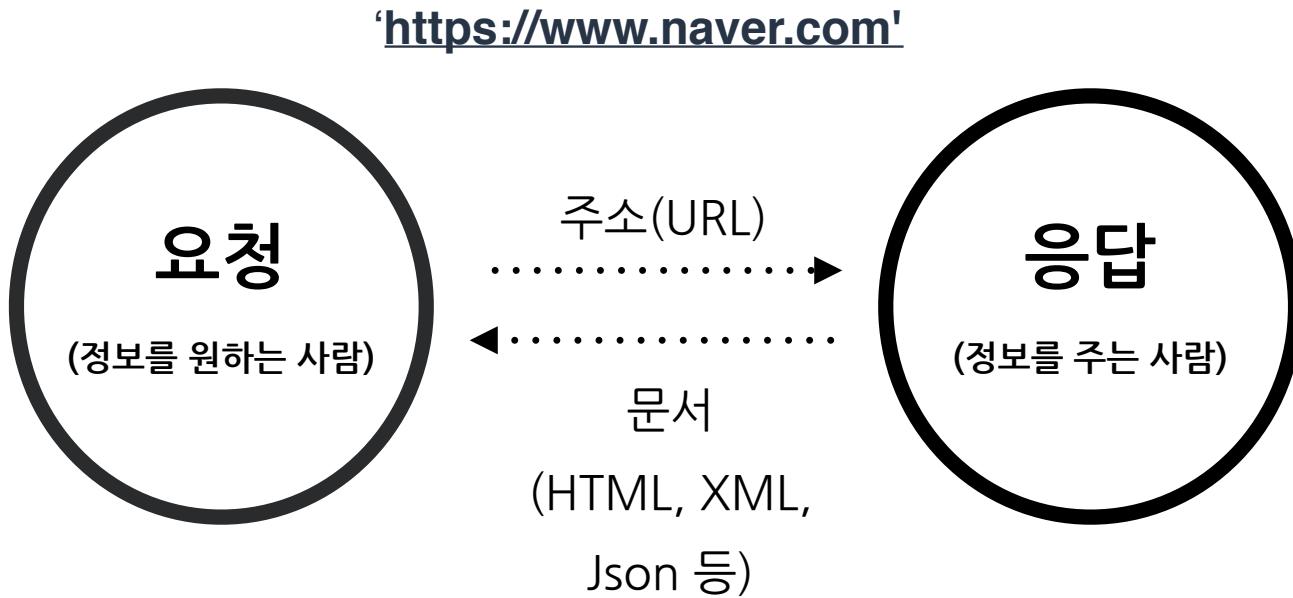
REST API?

[10/11] 2차 코딩테스트 사전 안내사항

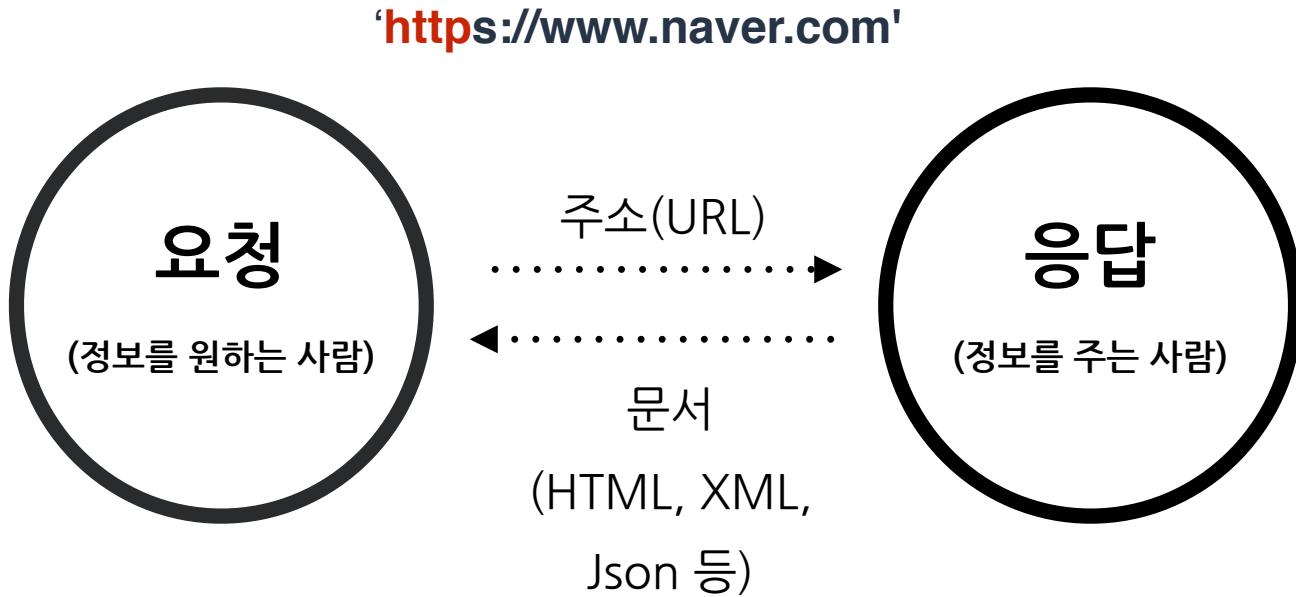
- 본 테스트는 언어 및 개발 환경의 제약이 없습니다.
본인에게 가장 편한 환경과 자신 있는 언어를 준비해주세요.
- 본 테스트 시 REST API와 JSON Parser가 필요합니다. 자신이 사용하는 언어에서 동작하는 코드를 작성해주세요.
 - 오픈소스 및 외부 라이브러리를 사용 하셔도 됩니다.
 - <https://httpbin.org>를 참고해서 REST API를 요청 후 반환되는 JSON 응답을 파싱해보세요.



웹 서비스의 기초



웹 서비스의 기초



HTTP

HTTP

(Hypertext Transfer Protocol)

문서를 전송하기 위한 프로토콜(규약)

HTTP

HTTP 기본 속성

비고

Stateless

상태정보가 저장되지 않음. 즉, 요청 사이에는 연결고리가 없음.
클라이언트가 서버와 상호작용하기 위해서 HTTP 쿠키*를 만들고,
상태가 있는 세션을 활용할 수 있도록 보완

Connectless

서버에 요청을 하고 응답을 한 이후에 연결은 끊어짐.

HTTP URL

URI

(Uniform Resource Identifier)

통합자 자원 식별자

HTTP URL

URL

(Uniform Resource Locator)

통합자 자원 지시자

HTTP URL

Scheme

Host

Port

Path

http://localhost:3000/posts/3

http://google.com/search?q=http Query

**http://getbootstrap.com/docs/4.1/layout/
overview#containers** fragment

HTTP URL

HTTP 요청 URL on Django

```
def index(request):  
    print(request)
```

HTTP URL

HTTP 요청 URL on Django

```
<class 'django.core.handlers.wsgi.WSGIRequest'>
    <WSGIRequest: GET '/boards/'>
```

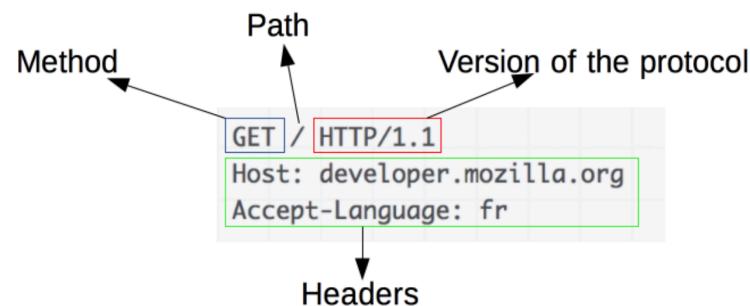
HTTP URL

HTTP 요청 URL on Django

```
def index(request):
    print(request.scheme)
    print(request.method)
    print(request.get_full_path())
    print(request.get_host())
print(request.build_absolute_uri())
```

HTTP 메시지

HTTP 요청 메시지



HTTP 메시지

HTTP 요청 메시지

구분

비고

시작줄(요청메시지)

GET /boards/3

헤더
(요청헤더, 일반헤더,
응답헤더 등)

User-Agent: Mozilla/5.0 (Macintosh;...); Firefox/51.0
Accept: text/html, ...
Accept-Language: en-US, en;

본문

POST 등 요청의 경우 본문의 내용이 포함될 수 있으며,
일반적으로 HTML 폼과 연계되어 사용됨

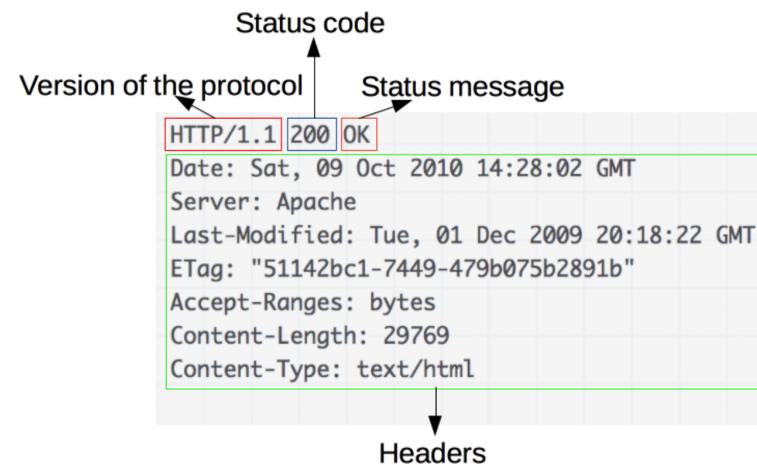
HTTP 메시지

HTTP 요청 메시지 **on Django**

request.META

HTTP 메시지

HTTP 응답 메시지



HTTP 메시지

HTTP 응답 메시지

구분

비고

상태줄

프로토콜 버전, 상태 코드
HTTP/1.1 404 Not Found.

헤더
(응답헤더)

Date: Web, 10 Aug 2016 13:17:18 GMT
Server: Apache
Set-Cookie: csrftoken=...

본문

모든 응답이 본문을 갖지는 않으나
단일 파일/단일 리소스 본문 혹은 다중 리소스 본문

HTTP 응답코드

HTTP 응답 코드

구분	비고
200 OK	GET: 리소스를 불러와서 메시지 바디에 전송되었습니다. HEAD: 개체 헤더가 메시지 바디에 있습니다. PUT 또는 POST: 수행 결과에 대한 리소스가 메시지 바디에 전송되었습니다. TRACE: 메시지 바디는 서버에서 수신한 요청 메시지를 포함하고 있습니다.
400 Bad Request	서버가 요청을 받지 못하였음
401 Unauthorized	미승인(비인증)
403 Forbidden	접근할 권리가 없음(클라이언트가 누군지 알고 있음)
404 Not Found	요청받은 리소스를 찾을 수 없음
500 Internal Server Error	서버 에러

HTTP 메서드

HTTP 메서드

구분

비고

GET

지정 리소스의 표시를 요청하며, 오직 데이터를 받기만 함.

POST

클라이언트 데이터를 서버로 보냄.

PUT/PATCH

서버로 보낸 데이터를 저장/지정 리소스의 부분만을 수정

DELETE

지정 리소스를 삭제

HTTP methods

HTTP 메서드 **on Django**

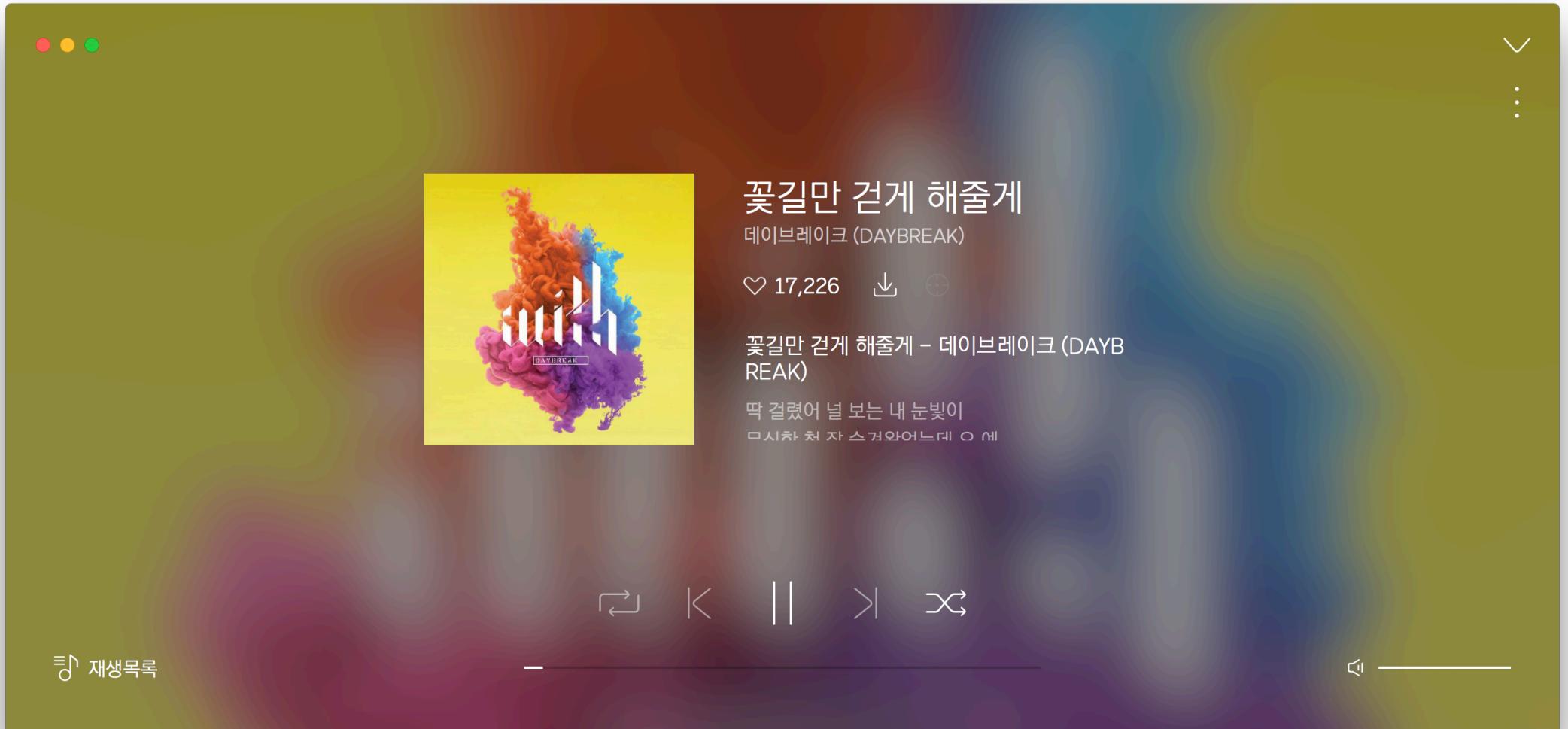
request.method

RESTful

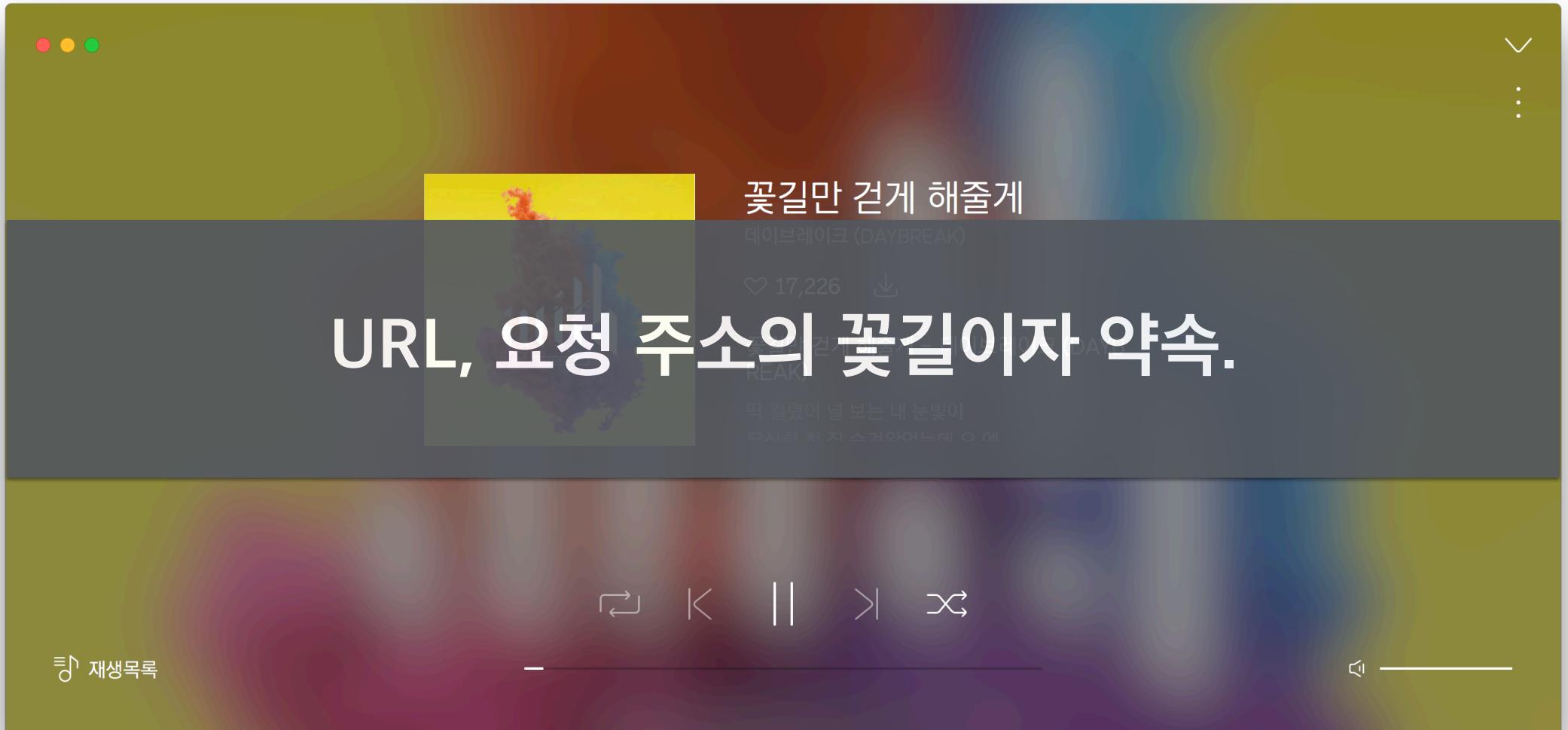
RESTful

(Representational State Transfer)

RESTful

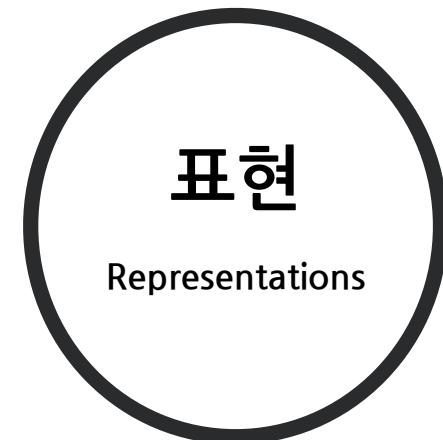


RESTful



RESTful

REST 구성



RESTful

REST 특징

구분	비고
Uniform	Uniform Interface 지정한 리소스에 대한 조작을 통일되고 한정적인 인터페이스로 활용
Stateless (무상태성)	무상태성 성격을 가지고 있으며, 세션 정보나 쿠키 정보를 별도로 저장하지 않고 관리하여 단순하게 요청만을 처리
Cacheable (캐시가능)	HTTP의 캐시 기능이 적용 가능함
Self-descriptiveness (자체표현구조)	무상태성 성격을 가지고 있으며, 세션 정보나 쿠키 정보를 별도로 저장하지 않고 관리하여 단순하게 요청만을 처리
Cilent - Server 구조	클라이언트 서버에서 개발할 내용이 명확하고 의존성이 줄어듬
계층형 구조	REST 서버는 다중 계층으로 구성될 수 있음.

RESTful

REST 중심 규칙

1. **URI는 정보의 자원을 표현해야 한다.**
2. **자원에 대한 행위는 HTTP Method로 표현한다.**

RESTful

GET /users/read/1

RESTful

GET /users/read/1

X, URI에 불필요한 정보가 포함.

GET /users/1

RESTful

GET /users/delete/1

RESTful

GET /users/delete/1

X, URI는 자원을 표현하는데에만 중점을 둬야함.

DELETE /users/1

RESTful

GET /users/create/1

RESTful

GET /users/create/1

X, GET 메소드는 적절하지 않음.

POST /users/1

RESTful

RESTful Web Service HTTP methods					
Resource	GET	PUT	POST	DELETE	
Collection URI, such as <code>http://example.com/resources/</code>	컬렉션에 속한 자원들의 URI나 그 상세사항의 목록을 보여준다.	전체 컬렉션은 다른 컬렉션으로 교체한다.	해당 컬렉션에 속하는 새로운 자원을 생성한다. 자원의 URI는 시스템에 의해 할당된다.	해당 컬렉션에 속하는 새로운 자원을 생성한다. 자원의 URI는 시스템에 의해 할당된다.	해당 컬렉션에 속하는 새로운 자원을 생성한다. 자원의 URI는 시스템에 의해 할당된다.
Element URI, such as <code>http://example.com/resources/item17</code>	요청한 컬렉션 내 자원을 반환한다.	해당 자원을 수정한다.	해당 자원에 귀속되는 새로운 자원을 생성한다.	해당 컬렉션내 자원을 삭제한다.	해당 컬렉션내 자원을 삭제한다.

URL 작성 TIP

1. 기본

- /는 계층 관계를 나타내는데 사용
- URI에는 소문자를 사용
- 파일 확장자는 포함시키지 않음
- 밑줄 대신에 하이픈을 활용

URL 작성 TIP

2. 리소스 관계를 표현하는 방법

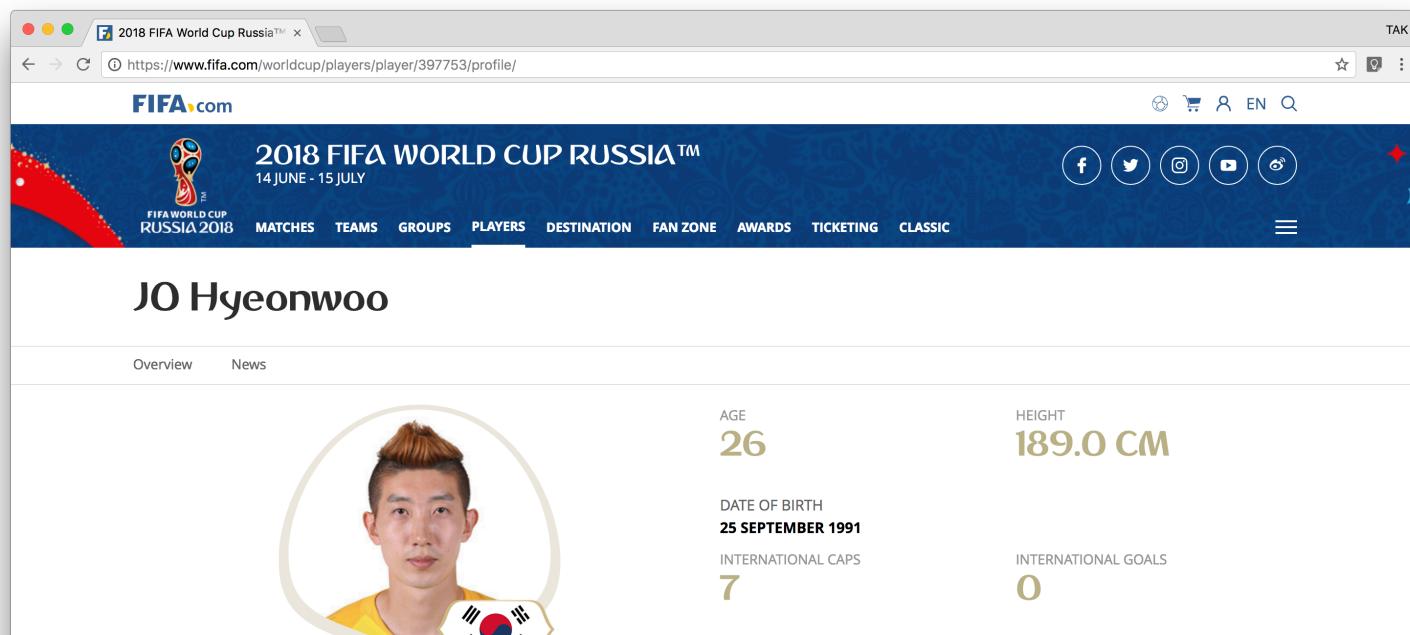
GET **fifa.com/worldcup/players/player/397753/profile**

월드컵 참가한 선수들 중 **397753** 선수의 **프로필**

URL 작성 TIP

2. 리소스 관계를 표현하는 방법

GET **fifa.com/worlcup/players/player/397753/profile**



URL 작성 TIP

3. 리소스를 표현하는 Collection과 Document

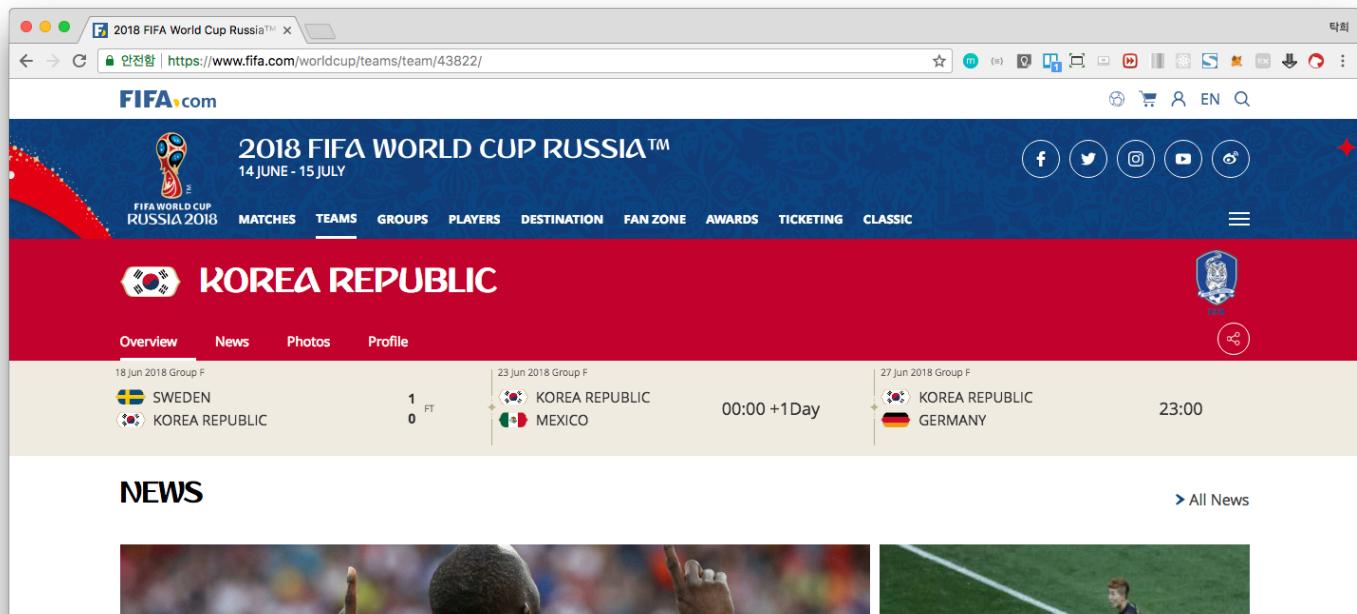
<https://www.fifa.com/worldcup/teams/team/43822>

월드컵 참가한 팀들 중 팀 43822

URL 작성 TIP

3. 리소스를 표현하는 Collection과 Document

<https://www.fifa.com/worldcup/teams/team/43822/>



RESTful

RESTful on Django?

\$ python manage.py show_urls

RESTful

HTML에서 공식적인 지원은 GET/POST

일부 프레임워크를 제외하고는 대부분 위 두가지로 수행해야함.