[데이터 트랙 - 향상] 웹서비스 데이터에 연결하여 조회하기

- 여기에 나오는 실습(따라하기)은 모바일 도서 2편-[실습 2] 웹서비스를 이용한 음반 정보 앱을 이용하겠습니다. 교육 전에
- 이 도서를 다운로드 해 두시기 바랍니다.

  https://welcome.devgear.co.kr/guide-embarcadero/book/

3-Tier 란

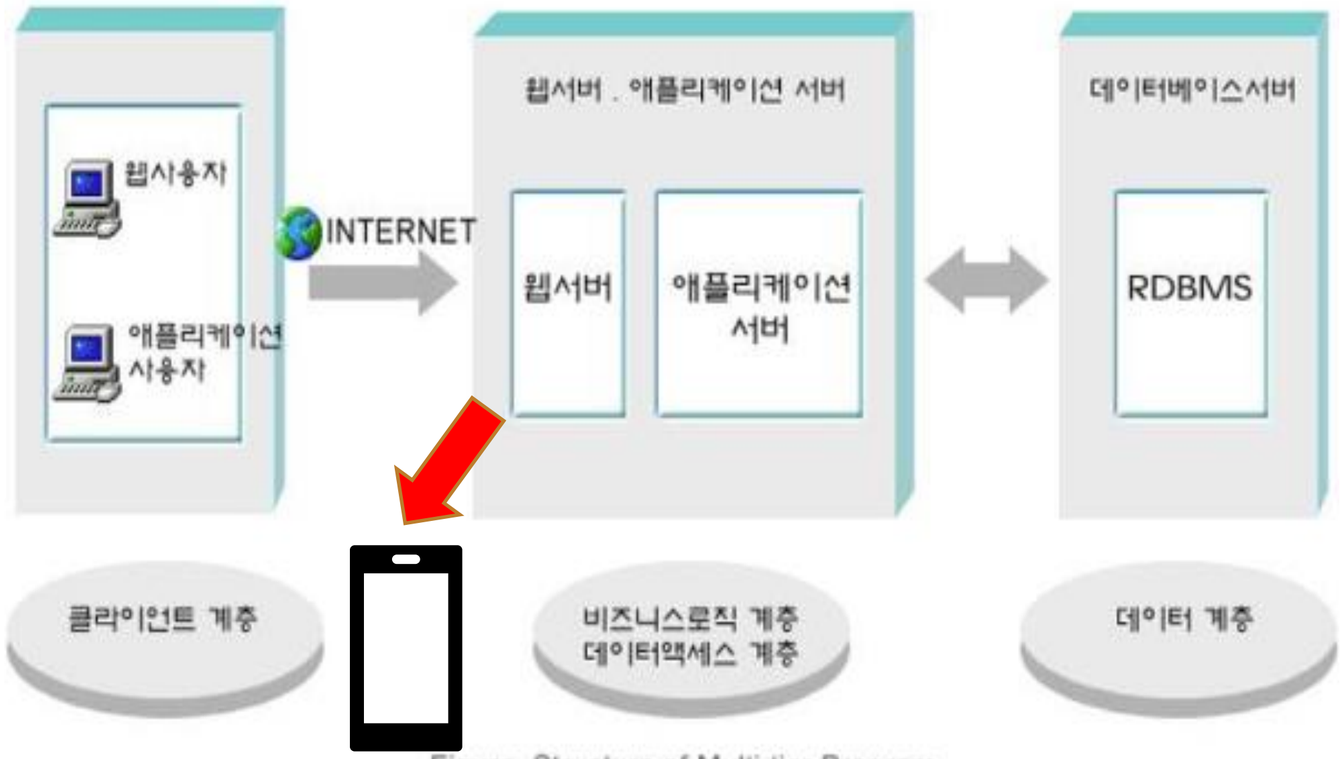


Figure. Structure of Multi-tier Program

REST

Rest 란 ?

REST(Representational State Transfer)는 인터넷 상의 컴퓨터 시스템간 상호 운용성을 제공하는 방법 중 하나입니다.

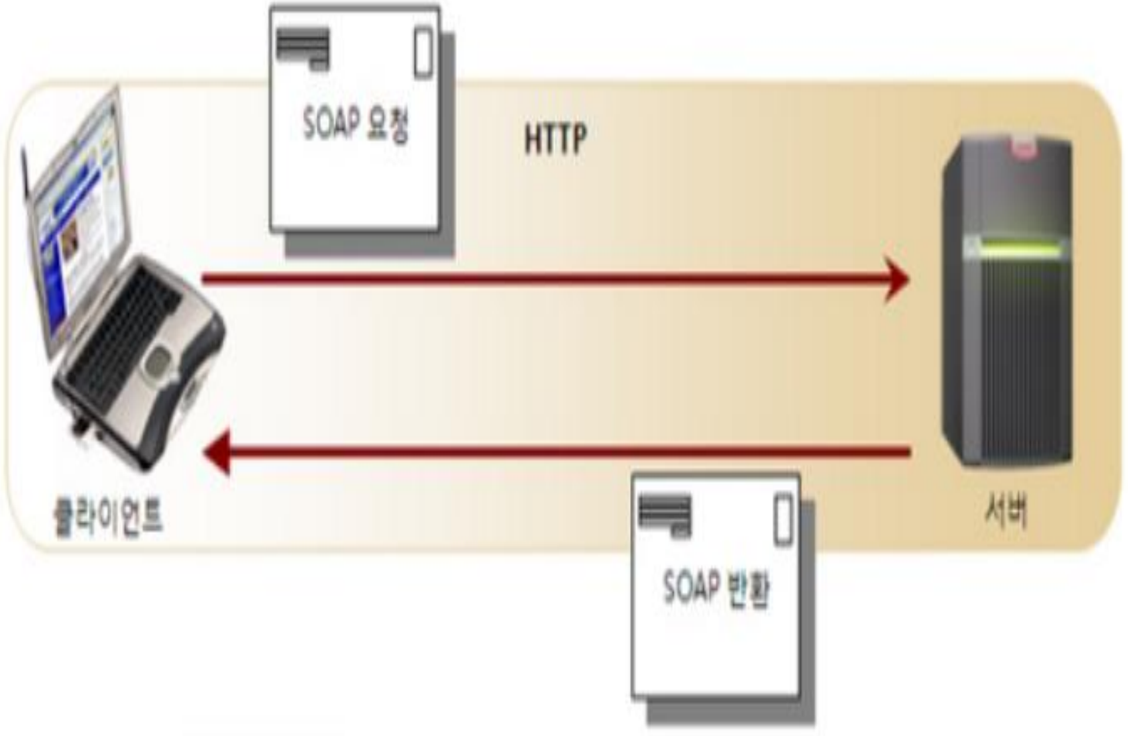
REST는 HTTP 기반으로 필요한 자원에 접근하는 방식을 정해 놓은 네트워크 아키텍처입니다. 여기서 자원이란, 저장된 데이터(DBMS 등)는 물론, 이미지/동영상/문서(PDF 등)와 같은 파일, 서비스(이메일 전송, 푸쉬 메시지 등) 등을 모두 포함합니다.

REST는 HTTP의 주요 저자 중 한사람인 로이 필딩의 2000년 박사학위 논문에서 처음 소개되었습니다.

•위키백과 - REST : https://ko.wikipedia.org/wiki/REST

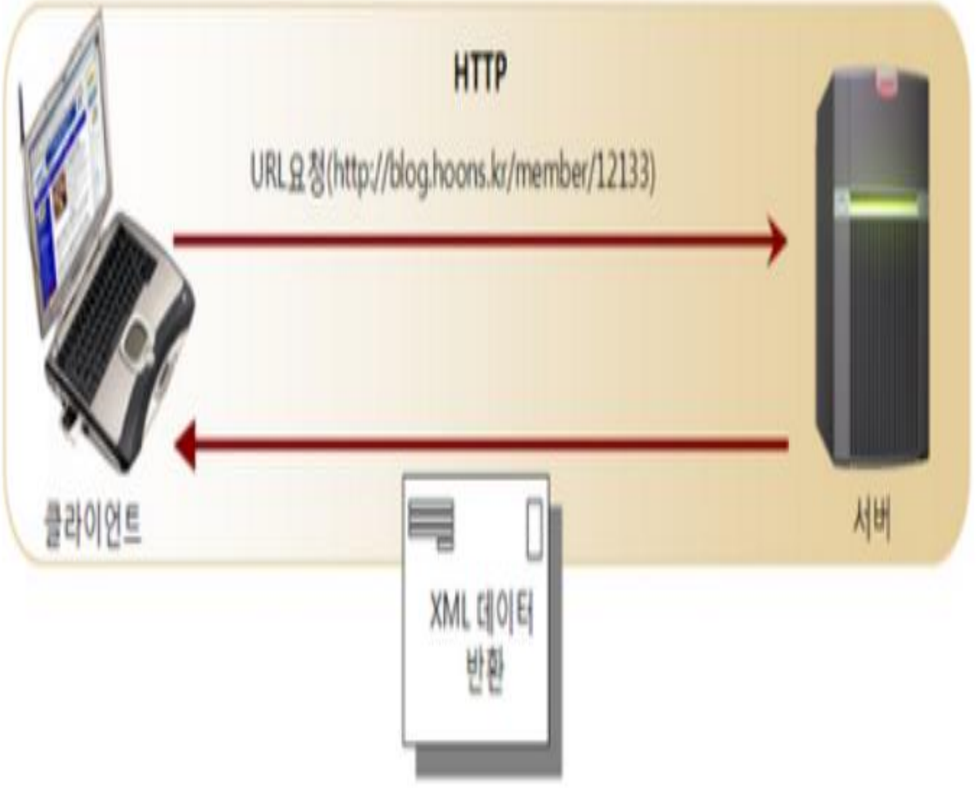
https://tech.devgear.co.kr/delphi_news/433404

(1) 기존의 SOAP을 이용한 일반적인 웹 서비스 방식은 다음과 같습니다.



(2) 반면, REST는 URL을 요청하게 됩니다.

복잡한 SOAP 메시지를 호출하지 않아도 URL을 이용해서 데이터를 간단하게 요청한다는 것이죠.



- **REST API**

REST 기반으로 서비스 API를 구현한 것을 REST API라고 합니다.

최근 OpenAPI(누구나 사용할 수 있도록 공개된 API: 구글 맵, 공공 데이터 등), 마이크로 서비스(하나의 큰 애플리케이션을 여러 개의 작은 애플리케이션으로 쪼개어 변경과 조합이 가능하도록 만든 아키텍처) 등을 제공하는 업체 대부분은 REST API를 제공합니다.

또, 사내 시스템들도 REST 기반으로 시스템을 분산해 확장성과 재사용성을 높여 유지보수 및 운용을 편리하게 할 수 있습니다.

REST는 HTTP 표준을 기반으로 구현하므로, HTTP를 지원하는 프로그램 언어로 클라이언트, 서버를 구현할 수 있습니다.
즉, REST API를 제작하면 델파이 클라이언트 뿐 아니라, 자바, C#, 웹 등을 이용해 클라이언트를 제작할 수 있습니다. 물론 반대의 방법도 가능합니다.

REST 주요 구성요소

REST 주요 구성요소는 리소스, 메소드, 메시지 3가지 입니다.

- **리소스** : 접근할 대상, URI를 통해 식별
- **메소드** : 리소스에 대한 행위, 표준 HTTP 메소드에 따라 자원에 접근(생성, 조회, 수정, 삭제)
- **메시지** : HTTP 해더와 바디에 포함된 메시지는 메시지를 처리하기 위한 충분한 정보를 포함

즉, REST는 어떤 자원(리소스)에 어떤 행위(메소드)를 어떻게(메시지) 할지 HTTP 기반으로 정해놓은 아키텍처입니다.

- **리소스(자원)**

리소스는 URI를 통해 정의합니다.

URI	의미
http://api.domain.com/books/	도서정보 콜렉션
http://api.domain.com/books/1/	1번 도서 정보
http://api.domain.com/books/1/photo/	1번 도서의 사진

예를 들면, 아래와 같은 URI는 다음과 같은 의미를 갖습니다.

리소스명은 동사보다 명사를 활용해 어떤 자원인지 표현하는데 집중해야 합니다.(/getBooks/와 같은 리소스는 적절하지 않습니다.)

슬래시(/)는 계층 관계를 나타내며, URI 앞쪽부터 넓은 의미로 사용합니다.
일반적으로 계층은 컬렉션(목록) 하위에 아이템을 지정하는 방식으로 정의합니다.

아래와 같이 URI를 구성할 수 있습니다.
/sports/soccer/players/1/
/(컬렉션)/(아이템)/(컬렉션)/(아이템)/

- # 메소드

REST에서는 메소드를 통해 리소스에 대한 행위를 정의합니다.

표준 HTTP 메소드 중 Get, Post, Put, Delete를 통해 자원의 CRUD를 정의합니다.

HTTP 메소드	자원에 대한 행위
POST	자원 생성(Create)
GET	자원 조회(Read)
PUT	자원 수정(Update)
DELETE	자원 삭제(Delete)

- # Endpoint

다음과 같이 메소드와 URI를 이용해 리소스에 접근합니다. URI 별 HTTP 메소드로 구현된 항목을 Endpoint라고 합니다.

HTTP 메소드	URI(자원)	Endpoint의 행위
POST	http://api.domain.com/books/	새로운 도서정보 생성
GET	http://api.domain.com/books/	도서정보 목록 조회
GET	http://api.domain.com/books/1/	1번 도서정보 조회
PUT	http://api.domain.com/books/1/	1번 도서정보 수정
DELETE	http://api.domain.com/books/1/	1번 도서정보 삭제

- 메시지

REST에서 자원에 대한 정보는 HTTP 바디와 HTTP 해더, 응답 상태코드를 활용합니다.

HTTP 바디

HTTP 바디에 포함된 데이터를 통해 자원에 대한 정보를 전달합니다.

HTTP 해더

HTTP 해더에는 HTTP 바디의 컨텐츠 종류를 명시할 수 있습니다. 해더에 정의된 컨텐츠 타입에 따라 데이터를 분석하도록 구현해야 합니다.
요청 HTTP 해더는 "Accept" 항목으로, 응답 HTTP 해더는 "Content-type"으로 컨텐츠 타입을 설명합니다.

몇가지 컨텐츠 타입은 다음과 같습니다.
•application/json
•application/xml
•text/plain
•image/jpeg
•image/png

응답 상태코드

리소스 요청에 대한 응답은 응답 상태코드로 설명할 수 있습니다.
대표적인 응답 상태코드는 아래와 같습니다.
•200 - 요청을 정상 수행
•201 - 리소스 생성 요청 성공(Post로 생성 요청 시에 한함)
•400 - 요청이 부적절함
•401 - 인증되지 않은 상태에서 보호된 리소스 요청
•403 - 공개되지 않은 리소스에 접근 요청(인증과 무관)
•404 - 존재하지 않는 리소스 요청
•406 - 지원하지 않는 미디어타입을 요청
•409 - 리소스 상태에 의해 해당 요청을 수행하지 못함

델파이로 REST API 구현

델파이의 HTTP 라이브러리를 이용해서 REST API를 구현할 수 있습니다.
REST API 구현해 특화된 기술은 서버 측 기술로는 EMS 서버 클라이언트 기술로는 REST 클라이언트 라이브러리가 있습니다.

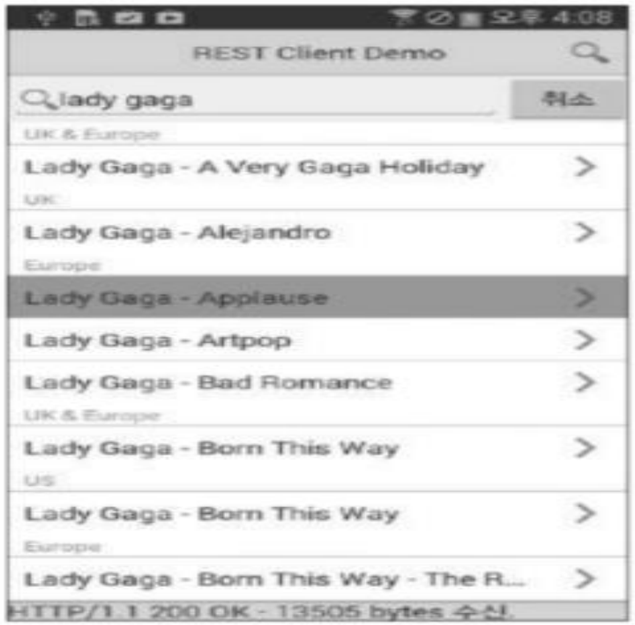
서버(Back-End) 프레임워크

•RAD Server(EMS Server) - REST API Endpoint 제공
•WebBroker - RAD 스튜디오(델파이, C++빌더) 웹 개발 프레임워크
•DataSnap - RAD 스튜디오 멀티티어 개발 프레임워크, TCP/IP, HTTP 프로토콜 제공
•Delphi MVC Framework - 오픈소스 웹서비스 개발 프레임워크
•mORMot - 오픈소스 SOA, ORM 개발 프레임워크

[실습] 웹서비스를 이용한 음반 정보 앱

웹서비스에 대해 이론적으로 충분히 이해되지 않았더라도 관계 없습니다. 델파이의 UI 컨트롤부터 시작하여 여러가지 컴포넌트와 기술을 사용해 보았으므로 따라하기 형식으로 진행 되는 이 실습이 크게 어렵지는 않을 것입니다.

이 실습을 완료하면 다음과 같은 앱이 완성됩니다. 이 앱은 DiscoGS에서 제공하고 있는 음반 정보 웹 서비스를 사용합니다. 날씨 등 여러분이 원하는 공공 데이터에도 마찬가지 방식으로 활용할 수 있습니다.



[실습] 웹서비스를 이용한 음반 정보 앱

실습 대상 사이트 - DiscoGS 실습 대상 사이트는 DiscoGS(http://www.discogs.com)

DiscoGS는 음반 정보 제공 사이트이며, REST 방식의 Open API를 제공하고 있습니다. (지금 설명되는 REST나, JSON, XML 등은 웹 서비스를 위한 국제 표준 규약들입니다. 이론 설명은 이 책의 후속 편 또는 구글 검색을 참고하세요)

API 정보는 개발자 페이지(http://www.discogs.com/developers/)를 통해서 확인 가능하 고, 우리

그 중에서 Search API(http://www.discogs.com/developers/resources/data base/search-endpoint. html)를 대상으로 진행하겠습니다. 검색 요청은 아래와 같은 URL이 사용되며, 중괄호로 감싸진 {query}, {pages는 상황에 맞는 파라미터로 치환됩니다.

- http://api.discogs.com/database/search?q={query}&type=master&per_page={page}

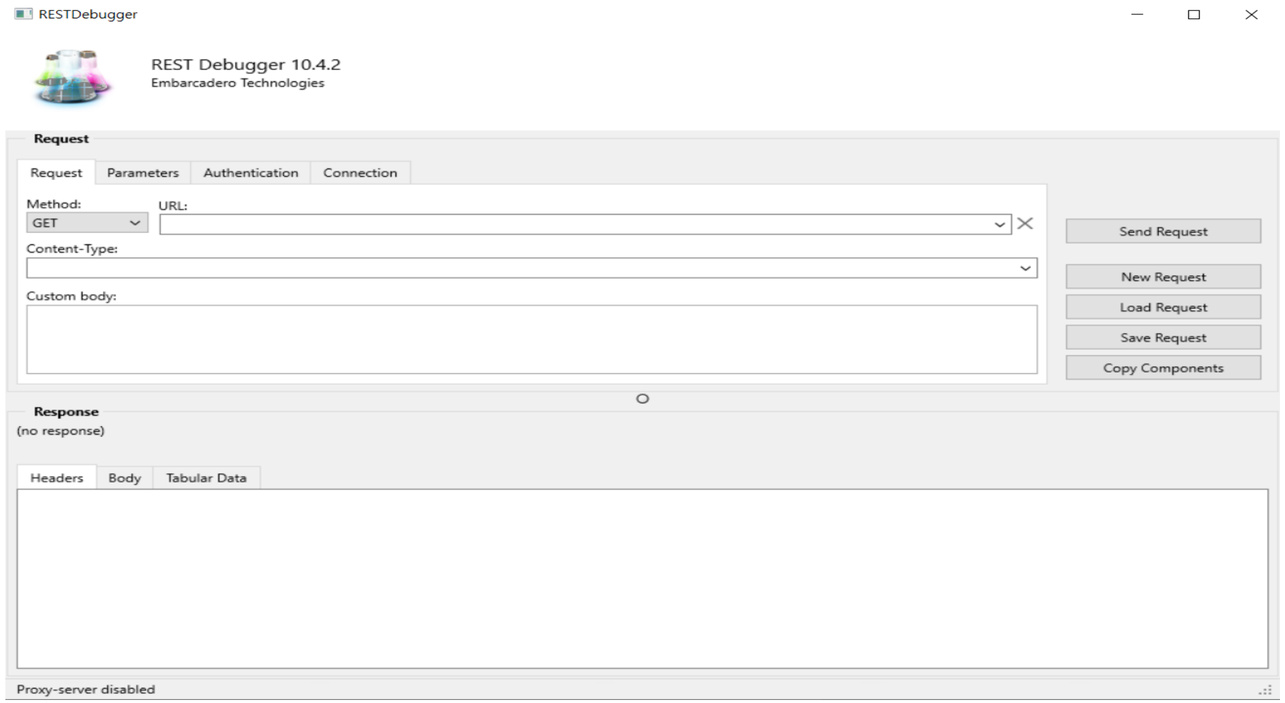
[실습] 웹서비스를 이용한 음반 정보 앱

응답은 JSON 형식으로 수신됩니다. 아래의 샘플데이터 중 "results"를 RootElement로 사용합 니다.
("results"의 자식 노드는 배열([])로 구성됩니다.)


```
{
  "pagination": {
    "per_page": 3,
    "pages": 1206721,
    "page": 1,
    "items": 3620161,
    "urls": {
      "last": "http://api.discogs.com/database/search?per_page=3&page=1206721",
      "next": "http://api.discogs.com/database/search?per_page=3&page=2"
    }
  },
  "results": [
    {
      "style": [ "Techno", "Experimental" ],
      "title": "Ken Ishii - Pneuma",
      "country": "Belgium",
      "format": [ "Vinyl" ],
      "uri": "/Ken-Ishii-Pneuma/release/13863",
      "label": "R & S Records",
      "catno": "RS93025",
      "year": "1993",
      "genre": [ "Electronic" ],
      "type": "release",
      "id": 13863
    },
... 생략 ...
```

[실습] 웹서비스를 이용한 음반 정보 앱
REST 분석 도구(REST Debugger)

REST 디버거는 REST Client를 이용해 REST 컴포넌트를 구성하기 이전에 REST API를 분석하고, 확인하기에 아주 좋은 도구입니다. (REST 00디버거도 델파이의 REST Client 컴포넌트 등을 이용해서 개발되었으며, 테스트 방식이 동일합니다.)
메인 메뉴 > Tools > REST Debugger를 클릭합니다.



RESTful 웹서비스 연동(Rest Client 소개)

이번에 소개해 드릴 컴포넌트는 XE5에서 새로 추가된 REST컴포넌트입니다.

REST 컴포넌트는 RestClient, RestRequest, RestResponse 3가지 주요 컴포넌트를 주축으로 구성되어 있으며, 요청과

응답을 객체화 시키고, 응답의 경우 RESTResponseDataSetAdapter를 연결하여 사용자의 데이터 파싱없이 리스트 및

그리드에서 사용할 수 있는 DataSource로 데이터를 자동변환합니다.