

인터넷에서 사용되는 여러 기술 : REST API 2

조인준
KBS 미디어기술연구소 차장

지난 편에서 REST API의 개요를 알아보았습니다. 이번 편에서는 REST API의 동작에 관한 예를 통해 개요보다는 한 발짝 더 깊이 들어가 보겠습니다.

본론을 시작하기에 앞서 그동안 설명에 URI(Uniform Resource Identifier)와 URL(Uniform Resource Locator) 등을 혼용해 왔습니다. 대략의 맥락을 이해하는 데는 큰 문제가 없을 수 있으나, 추후 독자들께서 더욱 자세한 내용을 개인적으로 찾아보실 때 URI와 URL 등의 구별로 인한 혼동이 없도록 한 번 정리해보는 것이 좋을 것 같습니다.

웹에서 찾을 수 있는 많은 자료에서 URI와 URL의 차이를 설명할 때 꼭 같이 나오는 것이 있습니다. 바로 URN(Uniform Resource Name)이라는 것인데 C군도 웹 관련 전문 업계에 종사하는 사람이 아니다 보니 자료를 통해 개념적으로만 그 존재를 알고 있을 뿐 실제로 어떻게 생겼는지, 어떻게 쓰이는지 모르겠습니다. 그렇지만, 많은 자료에서 이 셋의 상관관계를 설명하고 있고, 이를 포함해야 상호관계가 좀 더 선명해지는 부분도 있으므로 같이 포함하여 다루어보겠습니다.

URI는 Uniform Resource Identifier의 약자로 인터넷상의 리소스(데이터, 이미지, 텍스트 등)의 위치, 이름 또는 이 둘 모두를 이용해 식별하기 위한 문자열입니다. 비유하자면 ‘방송과기술’을 출판하는 회사를 구분하기 위해 회사의 이름인 ‘(주)월간 방송과기술’ 또는 회사 주소인 ‘서울특별시 양천구 목동 923-5’를 오른쪽과 같이 사용하는 것과 같은 것입니다.

회사명 : (주)월간 방송과기술
주 소 : 서울특별시 양천구 목동 923-5

URL은 Uniform Resource Locator의 약자로 인터넷상의 리소스(데이터, 이미지, 텍스트 등)의 위치를 나타내는 문자열입니다. 이는 개념적으로 아래와 같이 방송과기술을 출판하는 회사를 구분하기 위해 회사 주소인 ‘서울특별시 양천구 목동 923-5’를 사용하는 것과 같습니다.

주 소 : 서울특별시 양천구 목동 923-5

마지막으로 URN은 Uniform Resource Name의 약자로 인터넷상의 리소스(데이터, 이미지, 텍스트 등)의 이름을 나타내는 문자열입니다. 이는 개념적으로 그림과 같이 방송과기술을 출판하는 회사를 구분하기 위해 회사의 이름인 ‘(주)월간 방송과기술’을 사용하는 것과 같습니다.

회사명 : (주)월간 방송과기술

각각의 설명을 종합해보면 URI는 [그림 1]과같이 위치와 이름을 사용하여 리소스를 특정할 수 있다고 하였으므로 위치를 나타내는 URL과 이름을 나타내는 URN은 모두 URI가 될 수 있습니다. 반대로 이야기하면 URI는 URL도 될 수 있고 URN도 될 수 있습니다. 그런데 무슨 이유로 이런 헛갈리는 구조가 만들어졌을까요? 뭔가 이유가 있을 거로 생각해서 C군이 또 인터넷을 열심히 뒤지다 보니

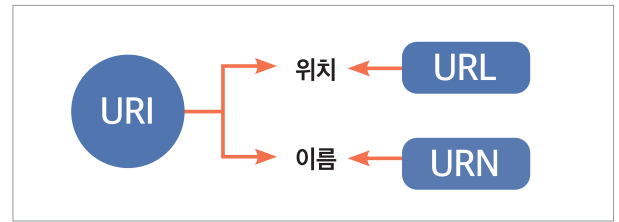


그림 1. URI, URL 및 URN의 관계

대략 다음과 같은 사연이 있었던 것 같습니다. 1990년대에 웹상의 리소스 구별을 위한 체계를 만드는 중에 URI를 ‘위치’와 ‘이름’ 두 가지 클래스로 분리했고 URI가 나타내는 것이 ‘위치’인지 ‘이름’인지 혼동되는 것을 막기 위해 URI 중에 ‘위치’를 나타내는 것을 URL, 이름을 나타내는 것을 URN이라고 부르게 되었다고 합니다. 하지만 동명이인이 있듯이 웹상의 리소스를 특정하기 위해 이름을 사용하는 것은 효과적이지 않아서 주로 URL을 통해서 웹페이지와 같은 웹상의 리소스를 특정하고 있습니다.

구분	항목
필수	프로토콜, 도메인
선택	경로, 포트번호, 기타(매개변수 등)

표 1. URL 구성요소



그림 2. URI 구성요소 예

[그림 2]는 [표 1]의 URL 구성요소를 [그림 3]의 방송과기술 홈페이지의 구독신청 페이지 주소를 예로 들어 표시한 것입니다. 우선 필수항목인 프로토콜과 도메인은 [그림 2]에 녹색 선으로 표시했습니다. html로 되어있는 웹 페이지 요청에는 http나 https(보안이 적용된 http)를 사용하므로 방송과기술 구독신청 페이지 URI의 프로토콜 항목은 http가 됩니다. http 다음에는 ‘:’와 ‘/’의 구분자를 붙인 후 그 뒤에 방송과기술 도메인인 tech.kobeta.com이 들어갑니다. 도메인 뒤에는 ‘:’와 함께 포트 번호가 들어가며, 이후 ‘/’를 구분자로 두고 경로 등이 들어갑니다. 참고로 http는 80번 포트를 기본 포트로 사용하기 때문에 생략해도 웹브라우저가 정상적으로 동작합니다. 브라우저 주소창에 tech.kobeta.com을 입력하면 방송과기술 홈페이지가 문제없이 출력되는 이유는 ‘:80’을 브라우저가 자동으로 처리하여 방송과기술 웹서버에 요청을 보내기 때문입니다. 만약 방송과기술 웹서버가 80번이 아닌 4000번 같은 일반적이지 않은 포트 번호를 사용한다고 하면 꼭 tech.kobeta.com:4000이라고 브라우저 주소창에 입력해야 홈페이지에 들어갈 수 있습니다.



그림 3. 방송과기술 홈페이지의 구독신청 페이지

지금까지 간단하게 URI, URL, 그리고 덤으로 URN에 대해 요점만 간단히 알아보았습니다. 그렇다면 이제 REST API가 URL을 이용하여 어떻게 동작하는지 GET, POST, PUT, DELETE 메서드를 예로 들어 핵심만 간단히 풀어서 설명 드리겠습니다. 설명을 위해 과자나 면 등의 식품을 파는 인터넷 주소(도메인) www.kobetafoods.co.kr를 가진 인터넷 상점이 있다고 가정하겠습니다. 이 상점이 보유한 전체 식품의 리스트를 REST API로 조회하기 위한 URI는 도메인 뒤에 구분자 ‘/’와 상품을 나타내는 경로 명인 products를 붙인 www.kobetafoods.co.kr/products로 정해져 있다고 하면, HTTP 프로토콜을 이용하여 GET 메서드로 www.kobetafoods.co.kr/products를 요청하면 [표 2]와 같은 상점이 보유한 전체 식품의 리스트를 수신하게 됩니다.

```
{ "과자" : [
    { "기호" : "snack_1"
      "이름" : "꿀 바른 강정"
      "수량" : 20}

    { "기호" : "snack_2"
      "이름" : "꿀물 가득 샌드"
      "수량" : 30}
    .....
  ],

  "면류" : [
    { "기호" : "noodle_1"
      "이름" : "불갈비 범벅면"
      "수량" : 15}
    .....
  ],
  .....
}
```

표 2. JSON 포맷의 서버 응답 예시

만약 새로운 과자가 출시되어서 상점의 제품리스트에 이를 추가해야 한다면 어떻게 해야 할까요? 이때 사용하는 REST API 메서드는 POST입니다. ‘말랑 감자 스틱’이라는 과자가 새로 출시되었고, 이 과자 50개가 입고되었다고 가정합니다. 그렇다면 상점의 리소스도 이를 반영해야 합니다. 상점의 과자 품목에 대한 URI가 www.kobetafoods.co.kr/products/snacks라고 한다면 POST 메서드를 사용하여 [표 3]의 JSON 데이터를 URI로 보냅니다. 만약 리소스에 새 과자 데이터 추가가 성공적으로 이루어지면 예전 HTTP 설명 때 알려드렸듯이 처리 성공을 알리는 200번 코드를 수신하게 됩니다.

```
{
  "구분" : "과자"
  "이름" : "말랑 감자 스틱"
  "수량" : 50
}
```

표 3. POST 메서드로 보내는 상품 추가 JSON 데이터

```
{ "과자" : [
    { "기호" : "snack_1"
      "이름" : "꿀 바른 강정"
      "수량" : 20}

    { "기호" : "snack_2"
      "이름" : "꿀물 가득 샌드"
      "수량" : 30}

    { "기호" : "snack_3"
      "이름" : "말랑 감자 스틱"
      "수량" : 50}
    .....
  ],

  "면류" : [
    { "기호" : "noodle_1"
      "이름" : "불갈비 범벅면"
      "수량" : 15}
    .....
  ],
  .....
}
```

표 4. 상품 추가 후 전체 상품 리스트 요청에 대한 JSON 응답

[표 3]의 새 품목 추가 처리가 성공한 후 다시 전체 상품을 GET 메서드로 조회해보면 [표 4]와 같이 새 과자가 추가되어 있는 것을 확인할 수 있습니다.

만약 말랑 감자 스틱이 10개가 팔려서 수량을 조정해야 할 때는 어떻게 할까요? 이 경우에는 리소스를 업데이트할 때 쓰이는 REST API의 PUT 메서드를 사용하면 됩니다. ‘말랑 감자 스틱’의 리소스를 가리키는 URI가 해당 상품의 ‘기호’인 snack_3을 경로 매개변수로 삼은 www.kobetafoods.co.kr/products/snacks/snack_3라고 한다면 PUT 메서드를 사용하여 [표 5]와 같이 새로 조정된 수량을 나타내는 JSON 데이터를 www.kobetafoods.co.kr/products/snacks/snack_3으로 보냅니다. 정상적인 처리가 이루어졌을 경우 서버는 200번의 성공 코드를 보내줍니다.

```
{
  "수량" : 40
}
```

표 5. PUT 메서드로 보내는 상품 수량 조정 JSON 데이터

만약 '말랑 감자 스틱'이 불티나게 팔려서 다음 날 매진되었다면 DELETE 메서드를 사용하여 해당 리소스의 URI인 `www.kobetafoods.co.kr/products/snacks/snack_3`를 가리키면 됩니다. 이번에도 마찬가지로 삭제 성공 후에는 서버가 200번의 성공 코드를 보내줍니다.

마지막으로 URI의 매개변수에 관해 간단히 정리해보겠습니다. 매개변수에는 경로 매개변수(path parameter)와 쿼리 매개변수(query parameter)가 있습니다. 우선 경로 매개변수는 `www.kobetafoods.co.kr/products/snacks/snack_3`의 예와 같이 특정 리소스를 지칭하는 기호 등을 경로처럼 사용한 경우입니다. 쿼리 매개변수의 경우는 다양한 경우가 있을 수 있어 간단한 예를 하나만 들어보겠습니다. 만약 수백 가지의 과자가 존재하고 이를 아주 긴 리스트로 한 번에 보는 것이 불편할 수 있어서 상점의 서버가 임의의 개수씩 여러 페이지로 분리된 리스트로 보여주는 옵션을 제공하고 있다고 가정합니다. 이 경우 리스트의 길이를 지정하는 파라미터는 `length`이고 몇 번째 페이지인지 지정할 수 있는 파라미터는 `page`라고 한다면, 과자의 리스트를 10개씩 분리해서 페이지별로 조회하는 URI는 다음과 같습니다. 물론 사용되는 메서드는 GET 메서드입니다.

```
www.kobetafoods.co.kr/products/snacks?length=10&page=1
```

위 URI를 보시면 과자 리소스를 가리키는 URI인 `www.kobetafoods.co.kr/products/snacks`의 뒤에 '?' 구분자를 두고 쿼리 매개변수를 나열한 것을 볼 수 있습니다. 쿼리 매개변수 사이에는 '&' 구분자가 들어갑니다. 당연한 것이지만 위 URI에서 `length`는 리스트의 길이를 나타내므로 `length=10`에 의해 10개씩 분리된 리스트를 요청하게 되고, `page=1`인 경우는 1번부터 10번까지의 과자 상품 리스트가, `page=2`인 경우는 11번부터 21번까지의 과자 상품 리스트가 응답으로 옵니다.

이것으로 REST API에 관해 간단하지만 어떻게 동작하는지 이해에 필요한 내용은 거의 다 포함된 설명을 드렸습니다. 다음 편에서는 인터넷 보안 프로토콜에 관해 설명 드리겠습니다.

P.S.

C군이 여러분께 전하는 내용 중 전문적 성격이 짙은 것은 엄밀한 언어를 사용하여 설명하기에는 한계가 있습니다.

본 내용은 설명하는 대상에 대한 전체적 맥락의 이해에만 이용하시고, 그 이상은 권위 있는 전문자료를 참고하시기 바랍니다. 📖