

Part3 REST API와 테스트 코드 작성하기

10장 REST API와 JSON

출처: 코딩 자율학습 스프링부트3 자바 백엔드 개발 입문, 홍팩, 길벗, 2023

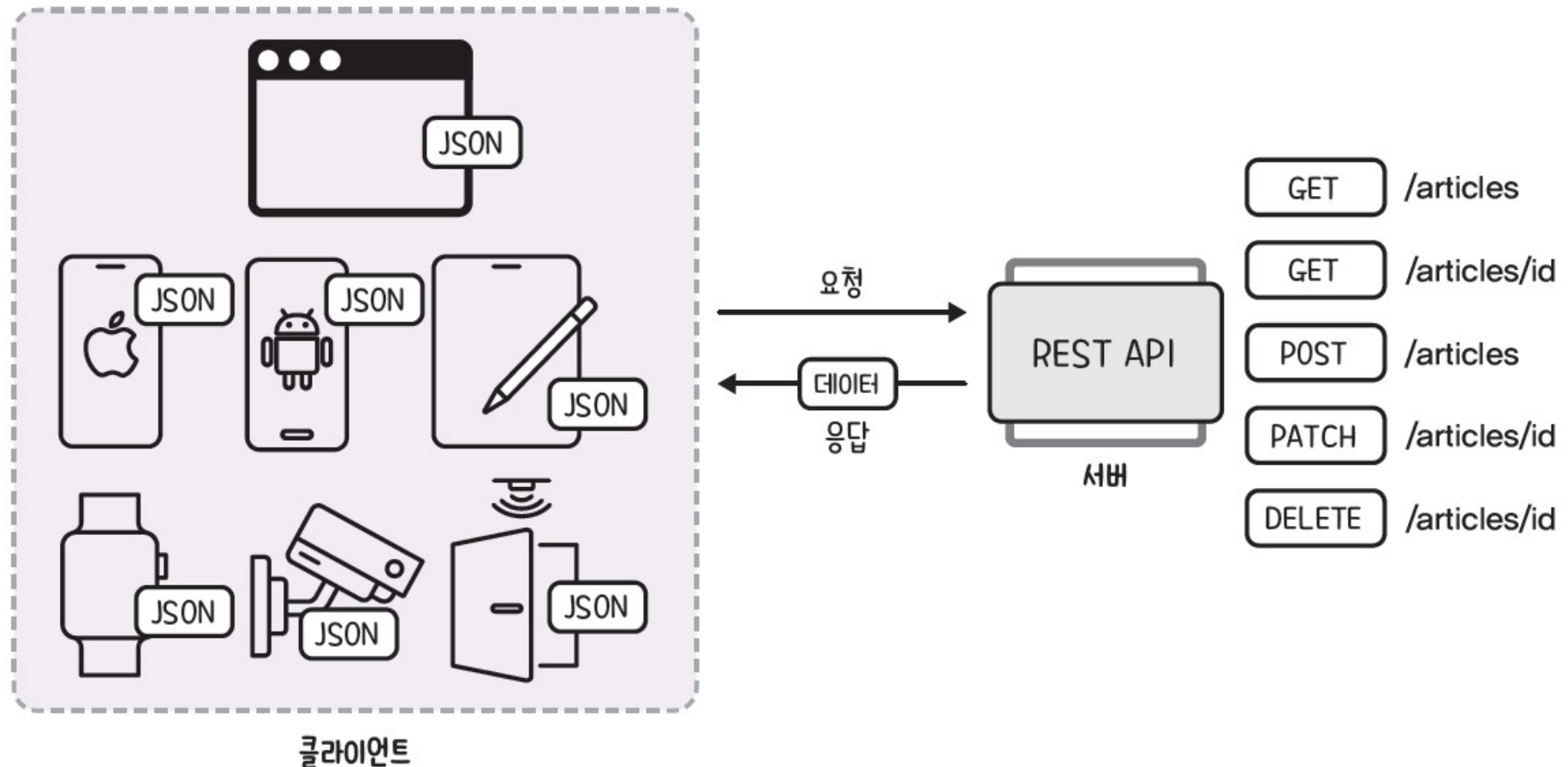
REST API

- **REST API(Representational State Transfer API)**
 - 서버의 자원을 클라이언트에게 구매받지 않고 사용할 수 있게 하는 설계 방식
 - HTTP 요청에 대한 응답으로 서버의 자원을 반환
 - 서버에서 보내는 응답이 특정 기기에 종속되지 않도록 모든 기기에서 통용될 수 있는 데이터를 반환

REST API

- 서버는 클라이언트의 요청에 대한 응답으로 뷰(view)가 아닌 데이터를 전송
- 응답 데이터는 JSON(JavaScript Object Notation)
 - Cf. XML

그림 10-2 REST API의 동작: 화면이 아닌 JSON 데이터 반환



XML vs. JSON

- 데이터를 구조화하여, 교환하기 위한 포맷
- XML (eXtensible Markup Language)
 - 사용자 정의형 마크업 언어
 - 태그 기반의 계층 구조, 확장성이 뛰어남
 - 상대적으로 데이터 크기가 크고, 처리 속도가 느림
 - 복잡한 문서 구조나 스키마 검증이 필요할 때 적합
- JSON (JavaScript Object Notation)
 - 키-값으로 구성, 객체와 배열을 표현
 - 간결하고 직관적
 - 상대적으로 데이터 크기가 작고, 처리속도 빠름
 - 웹, 모바일, JavaScript 환경에 최적화

XML vs. JSON

- XML

```
<articleForm>
  <id>1</id>
  <title>미래</title>
  <content>오늘을 보면 내일이 보인다.</content>
</articleForm>
```

- JSON

```
{
  "id": 1,
  "title": "미래",
  "content": "오늘을 보면 내일이 보인다."
}
```

JSON

- 키-값으로 구성, 객체와 배열 표현

- {}
 - ✓ 객체를 표현
- []
 - ✓ 배열을 표현
- “Key”
 - ✓ 키는 문자열로 표현
 - ✓ 큰 따옴표 사용: “ “
- :
 - ✓ 키와 값을 사이 구분자
- 값
 - ✓ 문자열, 숫자, 불리언, 배열, 객체 등
 - ✓ 값이 문자열일 때, 큰 따옴표 사용
- , (쉼표)
 - ✓ 항목 구분
 - ✓ 마지막 항목 뒤에 쉼표(,) 금지

```
{  
  "name": "홍길동",  
  "age": 20,  
  "isDeveloper": true,  
  "skills": ["JavaScript", "Python", "SQL"],  
  "address": {  
    "city": "서울시",  
    "country": "대한민국"  
  }  
}
```

REST API 동작 살펴보기

- 연습용 REST API 서버에 접속하여 HTTP 요청과 응답 실습
- {JSON} Placeholder 사이트
 - <https://jsonplaceholder.typicode.com>
 - 가짜(fake) API를 사용해 무료로 각종 테스트를 진행하나는 서비스 제공
 - 사이트에서 제공되는 자원

Resources

JSONPlaceholder comes with a set of 6 common resources:

<u>/posts</u>	100 posts
<u>/comments</u>	500 comments
<u>/albums</u>	100 albums
<u>/photos</u>	5000 photos
<u>/todos</u>	200 todos
<u>/users</u>	10 users

REST API 동작 살펴보기

- HTTP 메서드와 URL 경로

Routes

All HTTP methods are supported. You can use http or https for your requests.

GET	<u>/posts</u>
GET	<u>/posts/1</u>
GET	<u>/posts/1/comments</u>
GET	<u>/comments?postId=1</u>
POST	/posts
PUT	/posts/1
PATCH	/posts/1
DELETE	/posts/1

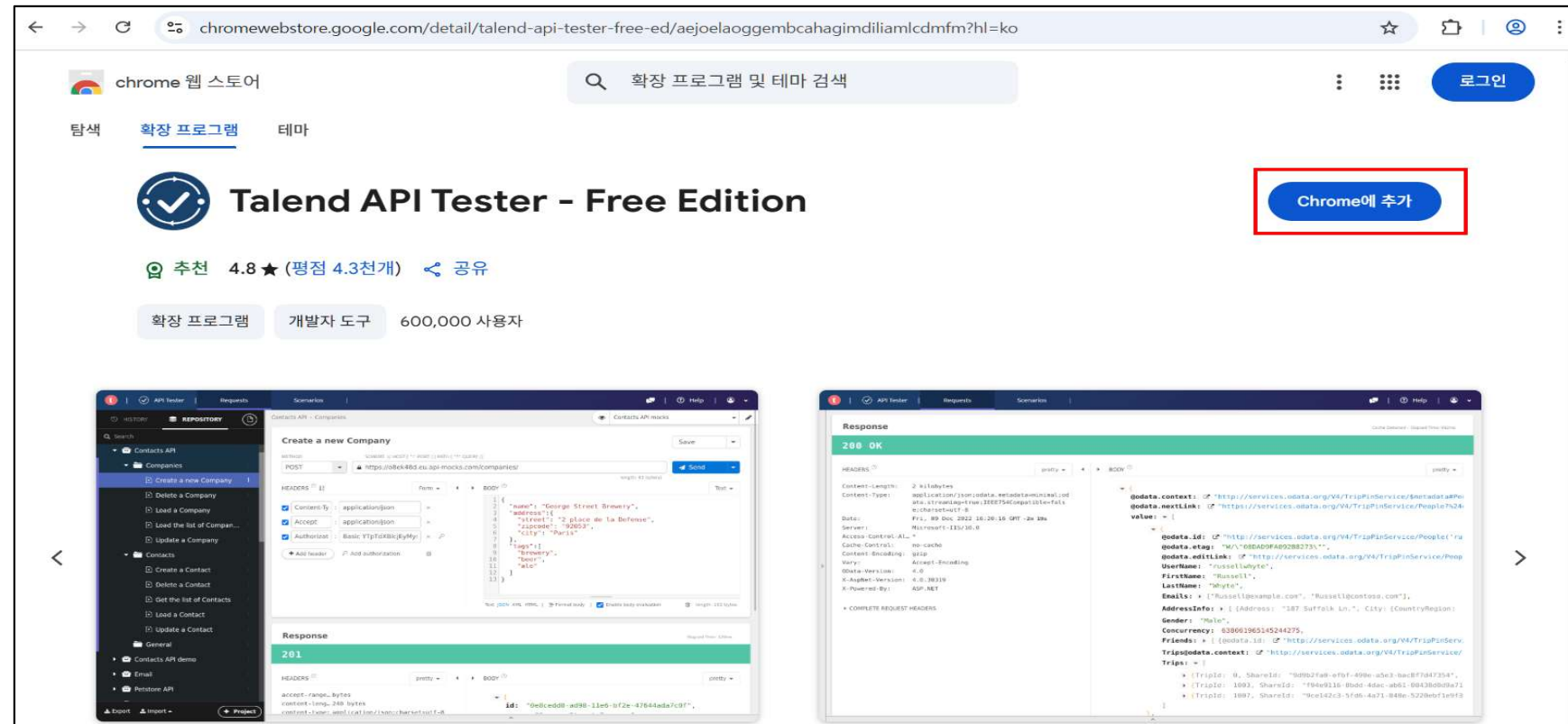
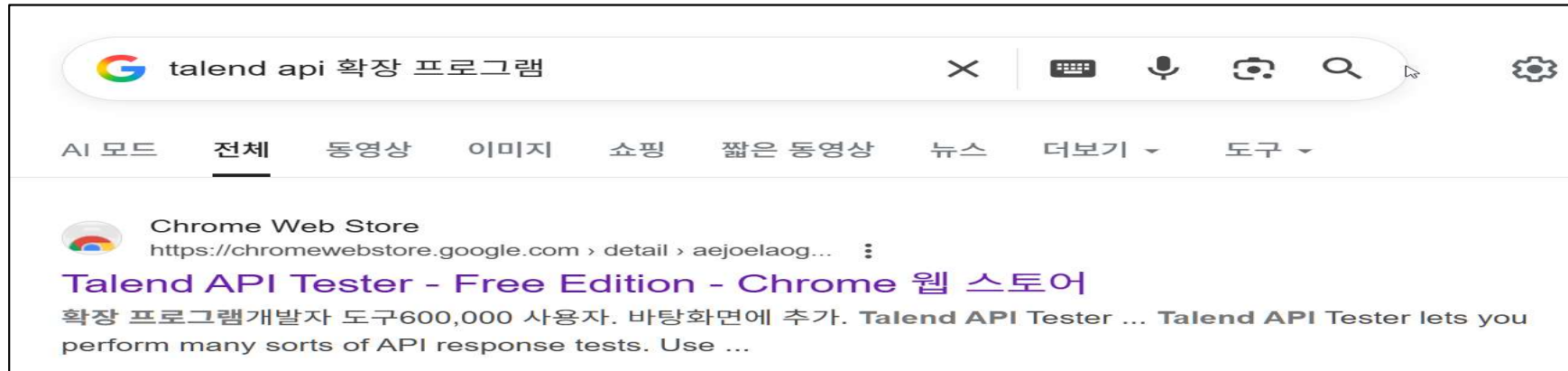
Note: see guide for usage examples.

REST API 동작 살펴보기

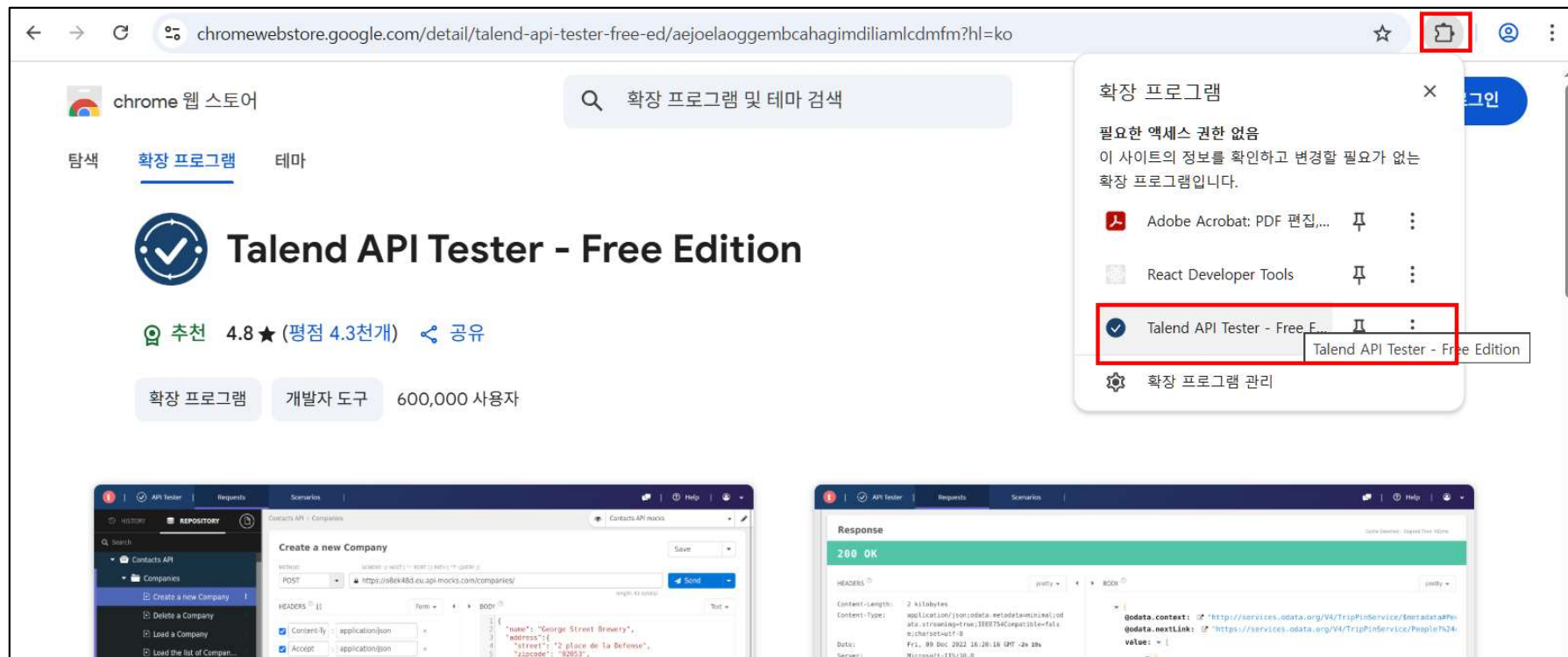
- REST API 사용 예
 - <https://jsonplaceholder.typicode.com/guide/>
 - PUT과 PATCH의 차이점
 - ✓ PUT
 - 기존 데이터를 전부 새 내용으로 변경
 - 기존 데이터가 없다면 새로 생성
 - ✓ PATCH
 - 기존 데이터 중에서 일부만 새 내용을 변경

Talend API Test 설치

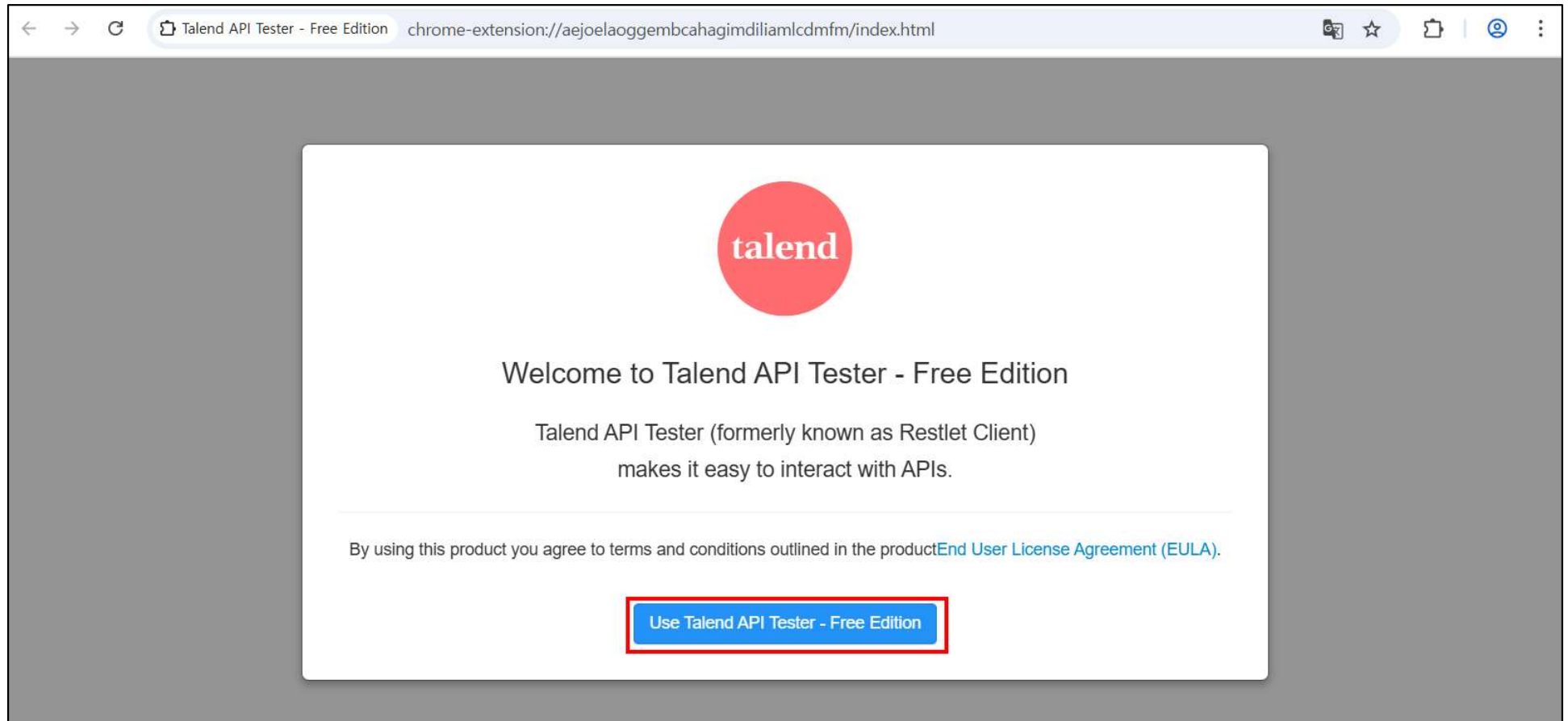
- 구글 검색창에서 “talend api 확장 프로그램” 검색



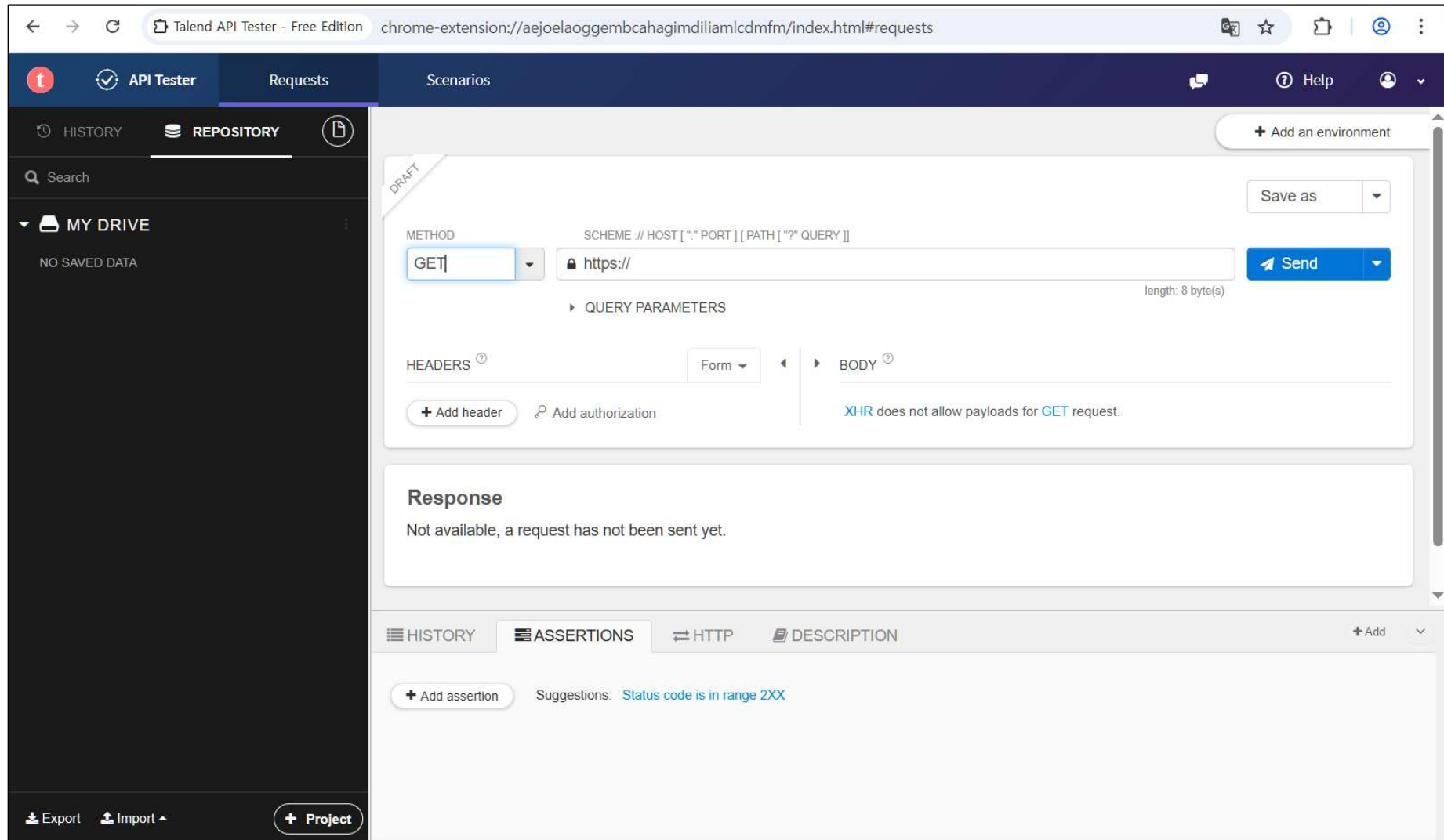
Talend API Test 설치 및 실행



Talend API Test 실행



Talend API Test 실행



HTTP 상태 코드

- 클라이언트가 보낸 요청이 성공했는지, 실패했는지 알려주는 코드

상태코드	설명
1XX(정보)	요청이 수신돼 처리 중입니다.
2XX(성공)	요청이 정상적으로 처리됐습니다.
3XX(리다이렉션 메시지)	요청을 완료하려면 추가 행동이 필요합니다.
4XX(클라이언트 요청 오류)	클라이언트의 요청이 잘못돼 서버가 요청을 수행할 수 없습니다.
5XX(서버 응답 오류)	서버 내부에 에러가 발생해 클라이언트의 요청에 대해 저절히 수행하지 못했습니다.

HTTP 메시지 구성

- 시작 라인

- HTTP 요청 또는 응답 내용, 시작 라인은 항상 한 줄로 끝남
- 헤더: HTTP 전송에 필요한 부가 정보(metadata)
- 빈 라인: 헤더의 끝을 알리는 빈 줄, 헤더가 모두 전송되었음을 알림
- 본문: 실제 전송하는 데이터