
The AR Form Project

What is the AR Form Project?

- Create an online form for students to fill out when they are planning on taking a test in AR
 - Confirm the date and time of the test with teacher, student, and AR staff
 - Organize this info
-

Our Options:

Microsoft and/or Java SQL:

- Advantages:
 - Local server
 - Local management
- Disadvantages:
 - Hard to scale
 - No ability to utilize online applications or http protocols

Outlook Calendar REST API:

- Advantages:
 - Stored in cloud
 - Can use java
 - Can use existing email addresses and Outlook calendar and email API
 - Automate processes easily
 - Disadvantages:
 - Java has limited native support for networking
-

—

**We chose to use the
Outlook REST API
with Java**

REST API

Connecting Online Resources to Applications

—

REST: Representational State Transfer

Why I'm using it

AR form project:

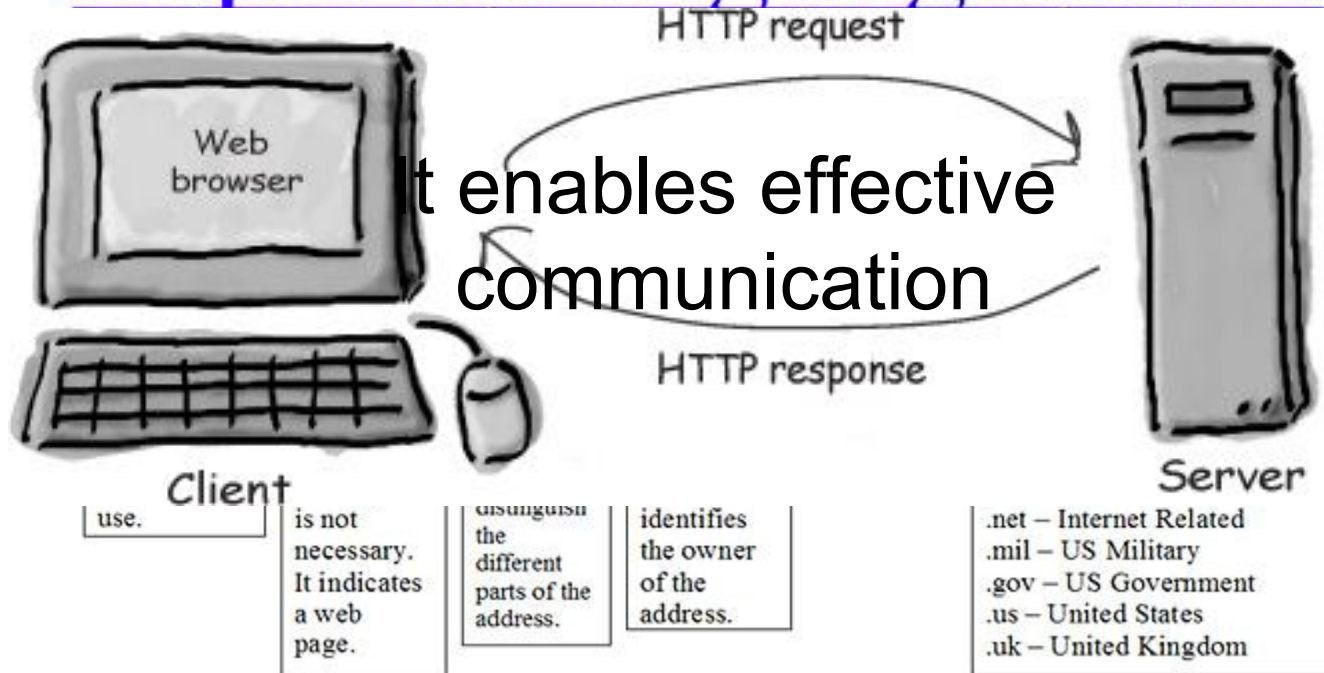
- Outlook calendar api
 - Populate calendar with information automatically
 - Confirmation emails
-

—

**Before REST:
HTTP(S),
HyperText Transfer Protocol
(Secure)**

What is HTTP Protocol?

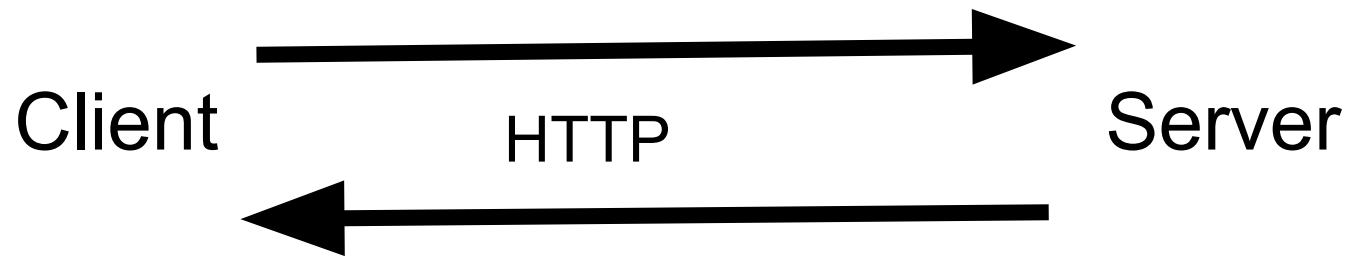
<http://www.google.com>



GET = Consume data

POST = Write data

What does this have
to do with REST?

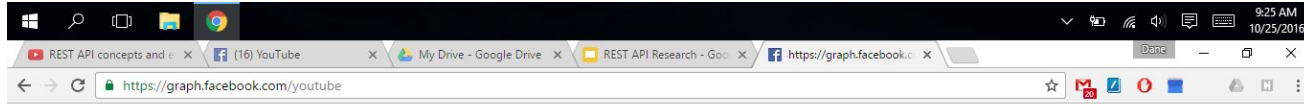


Example:

<http://www.facebook.com/youtube> brings you the facebook youtube page

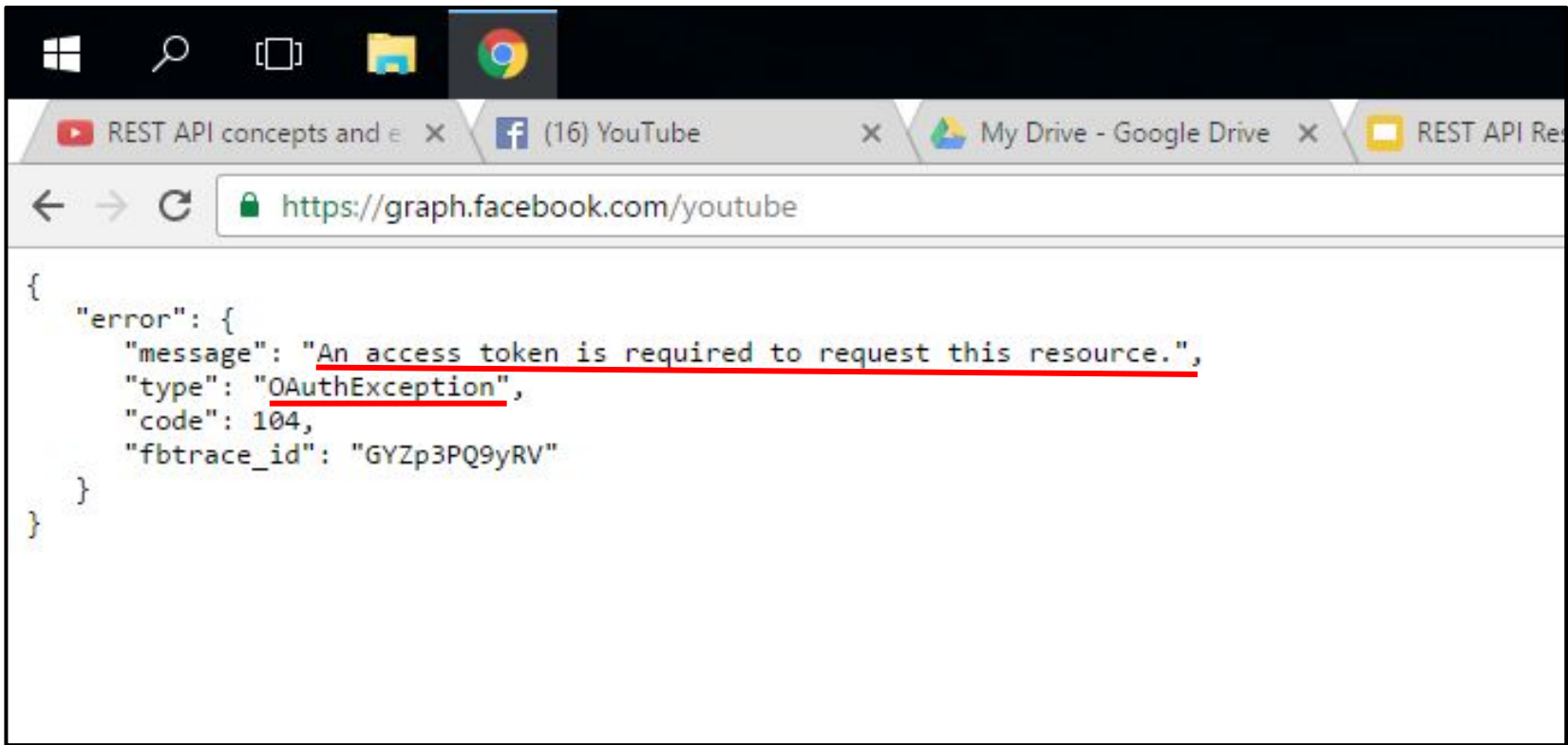
Or you could try a different url:

<http://graph.facebook.com/youtube> which gives you the facebook REST response for the youtube facebook page.

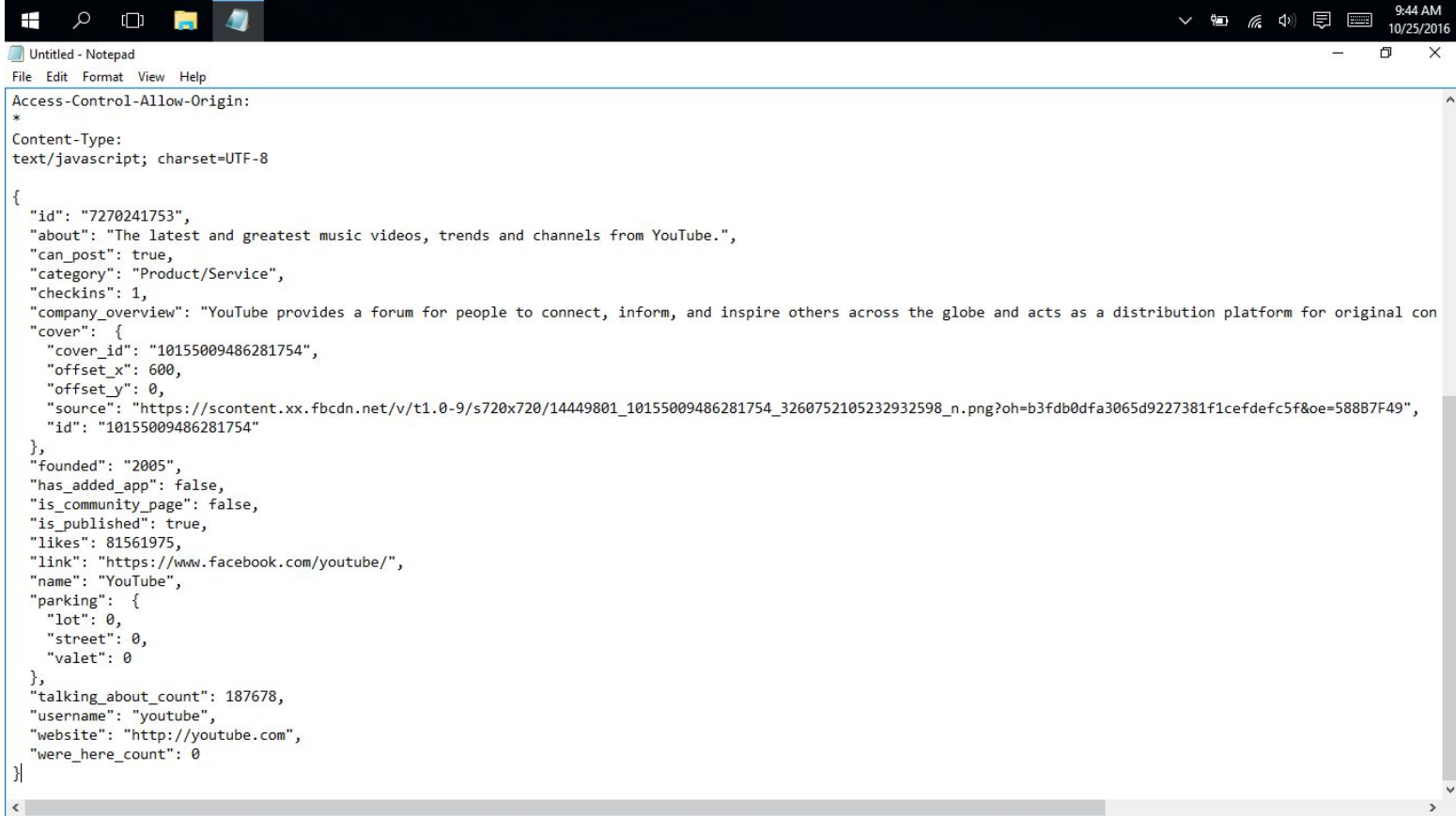


```
{
  "error": {
    "message": "An access token is required to request this resource.",
    "type": "OAuthException",
    "code": 104,
    "fbtrace_id": "GYZp3PQ9yRV"
  }
}
```

What Happened!?



Facebook is secured by OAuth2, this way you need to login before being able to use the API, otherwise people would be able to start posting things without permission



```
Access-Control-Allow-Origin:
*
Content-Type:
text/javascript; charset=UTF-8

{
  "id": "7270241753",
  "about": "The latest and greatest music videos, trends and channels from YouTube.",
  "can_post": true,
  "category": "Product/Service",
  "checkins": 1,
  "company_overview": "YouTube provides a forum for people to connect, inform, and inspire others across the globe and acts as a distribution platform for original con",
  "cover": {
    "cover_id": "10155009486281754",
    "offset_x": 600,
    "offset_y": 0,
    "source": "https://scontent.xx.fbcdn.net/v/t1.0-9/s720x720/14449801_10155009486281754_3260752105232932598_n.png?oh=b3fdb0dfa3065d9227381f1cefdefc5f&oe=58887F49",
    "id": "10155009486281754"
  },
  "founded": "2005",
  "has_added_app": false,
  "is_community_page": false,
  "is_published": true,
  "likes": 81561975,
  "link": "https://www.facebook.com/youtube/",
  "name": "YouTube",
  "parking": {
    "lot": 0,
    "street": 0,
    "valet": 0
  },
  "talking_about_count": 187678,
  "username": "youtube",
  "website": "http://youtube.com",
  "were_here_count": 0
}
```

With authentication, we get information on the facebook.com/youtube webpage organized in json formatting

**Try Spotify API
Console...**

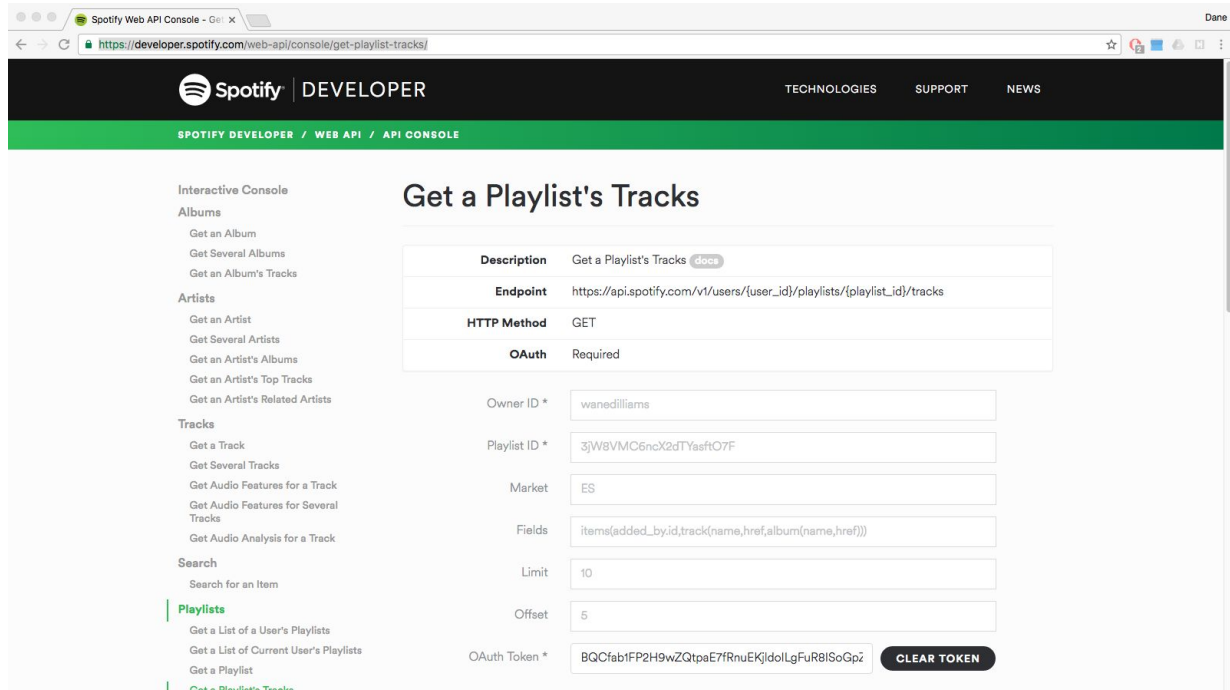
Let's do a GET

[Spotify API console](#) lets us try out the Spotify REST api without needing to write any code! Let's give Mr. Srock an upgrade to his playlist :)

Today's Top Hits:

Owner ID:
spotify

Playlist ID:
5FJXhjdILmRA2z5bvz4nzf



The screenshot shows the Spotify Web API Console interface. The browser address bar displays `https://developer.spotify.com/web-api/console/get-playlist-tracks/`. The page has a dark header with the Spotify logo and 'DEVELOPER' text, and a green navigation bar with links for 'SPOTIFY DEVELOPER', 'WEB API', and 'API CONSOLE'. On the left, a sidebar lists various API endpoints under categories like 'Albums', 'Artists', 'Tracks', 'Search', and 'Playlists'. The 'Playlists' section is active, showing 'Get a Playlist's Tracks' as the selected endpoint. The main content area displays the endpoint details in a table and a form below it.

Description	Get a Playlist's Tracks docs
Endpoint	<code>https://api.spotify.com/v1/users/{user_id}/playlists/{playlist_id}/tracks</code>
HTTP Method	GET
OAuth	Required

Owner ID *	<input type="text" value="wanedilliams"/>
Playlist ID *	<input type="text" value="3jW8VMC6ncX2dTYesftO7F"/>
Market	<input type="text" value="ES"/>
Fields	<input type="text" value="items[added_by.id,track(name,href,album(name,href))]]"/>
Limit	<input type="text" value="10"/>
Offset	<input type="text" value="5"/>
OAuth Token *	<input type="text" value="BQCfab1FP2H9wZQtpaE7fRnuEKjldoLgFuR8lSoGpZ"/> <button>CLEAR TOKEN</button>

And now a POST...

Let's Make Mr. Srock a playlist!

I made an account for this class:

Username / owner ID:
eastsidecatholicpcs

Password: Caroline!

Steps:

- [Create a Playlist](#)
 - Owner ID
 - Public = true
 - [Add Tracks to a Playlist](#)
 - Spotify track uri
 - Right click track in play.spotify.com for code at end of url (not uri)
 - Uri = spotify:track:code
 - Owner ID
 - Playlist ID
 - I will show you this after creating the playlist
-

Let's see the playlist!

Thanks for participating :)
