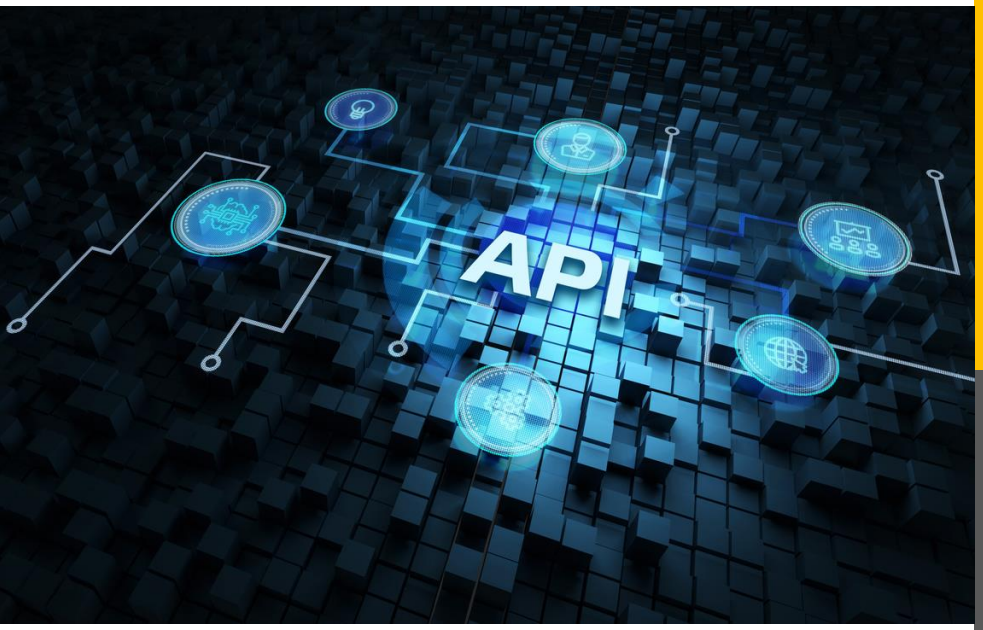
COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI



comillas.edu

Universidad Pontificia Comillas
ICAI

Analítica Social y de la Web
Social and Web Analytics

2.1 APIs, autenticación y parseo de datos

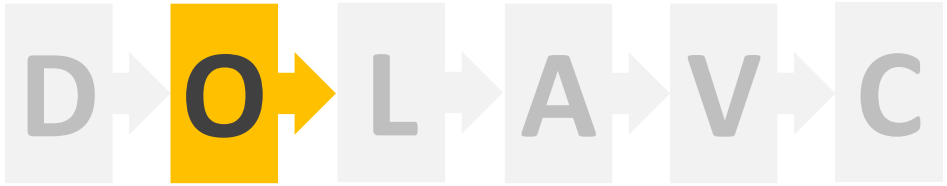
Javier Ruiz de Ojeda

Curso 2023-24
Segundo semestre

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

Uso de APIs

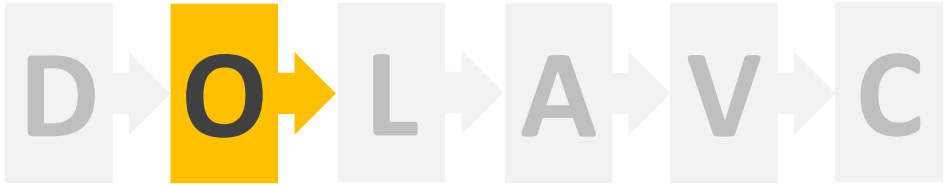
Uso de APIs
Tipos de APIs



- ***RESTful APIs***:
  - El tipo más común entre las redes sociales.
  - Su información es estática y proviene del histórico de datos.
  - Se basa en el protocolo HTTP y está diseñado para simplificar el envío de datos.
  - Los comandos principales son `Get` y `Post`.
- ***Stream APIs***:
  - Se emplean solo cuando hay un requisito específico para la colección de datos en tiempo real (en lugar de históricos)
  - Una de las más usadas es la de Twitter, por su naturaleza.
  - El *output* es muy similar al de las REST APIs.

Uso de APIs

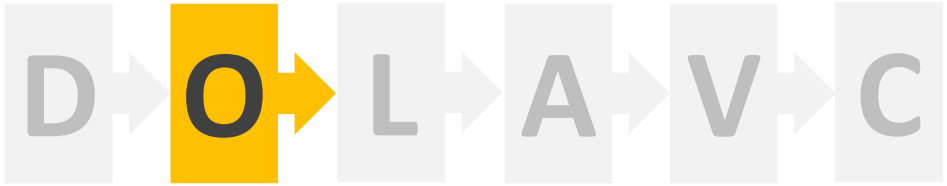
Ventajas de las APIs



- **Datos sociales**: Permiten extraer datos valiosos sobre los usuarios de redes sociales y el contenido que publican, que pueden usarse para hacer análisis de comportamiento y de la forma de pensar y de sentir.
- **Desarrollo de aplicaciones**: Miles de aplicaciones se construyen empleando APIs de redes sociales para proporcionar servicios complementarios a los de dichas redes.
- **Marketing**: Las APIs de redes sociales permiten automatizar algunas actividades de marketing y publicar posts en plataformas dirigidos a sus usuarios. También permite enriquecer y mejorar los datos de marketing sobre estos usuarios.

Uso de APIs

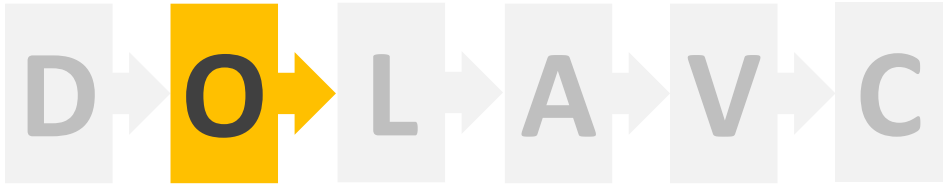
Limitaciones de las APIs



- **Límites en las tasas**: Tanto por límites en su infraestructura como por sus objetivos de negocio, no siempre podremos obtener todos los datos que queramos a alta velocidad, debemos tenerlo e cuenta en nuestra estrategia.
- **Cambios en las APIs**: Igual que los algoritmos de las redes sociales, su forma de compartir datos varía con el tiempo (o puede incluso detenerse), impactando a nuestras estrategias si no lo hemos tenido en cuenta y lo estamos vigilando.
- **Consideraciones legales**: La regulación sobre redes sociales suele ser estricta sobre el uso que admiten de sus datos y servicios. Deberemos analizar el contexto legal antes de considerar una aplicación de los datos.

Uso de APIs

Principios de conexión con APIs



- **Registro en la app**: Casi todas las redes sociales requerirán un registro en su plataforma, incluyendo información personal y el uso previsto de sus servicios API, generándose claves de autenticación y de consumo.
- **Autenticación**: Uso de las claves generadas en el paso anterior para autenticar tu aplicación.
- **API *endpoint hunting***: Los *endpoints* de la API variarán entre proveedores, así que será necesario revisar la documentación que ofrecen para decidir qué *endpoint* se ajusta mejor a nuestras necesidades.

Developer Documentation

Learn the basics of how to send and receive data from the Facebook Social Graph and how to implement the APIs, Platforms, Products, and SDKs to fit your application needs.

App Development
Register as a developer, configure your app's settings in the App Dashboard, and build, test, and release your app.

Docs

Graph API
The primary way for apps to read and write to the Facebook Social Graph.

Docs

App Review
Verify that your app uses our products and APIs in an approved manner.

Docs

App Integrations

- App Events
- App Links
- Graph API
- Groups API
- Pages
- Webhooks

Authentication

- Facebook Login
- Limited Login
- Login Connect with Messenger

Business Messaging

- WhatsApp Business Platform
- Messenger Platform
- Workplace

Marketing and Commerce

- Facebook App Ads
- Automotive Ads
- Catalog
- Commerce Platform
- Conversions API
- Cross-Channel Conversion

Videos

- Developer Documentation for Reels
- Developer Documentation for Videos
- Video API
- Live Video API

Gaming

- Games
- Game Payments

Artificial Intelligence

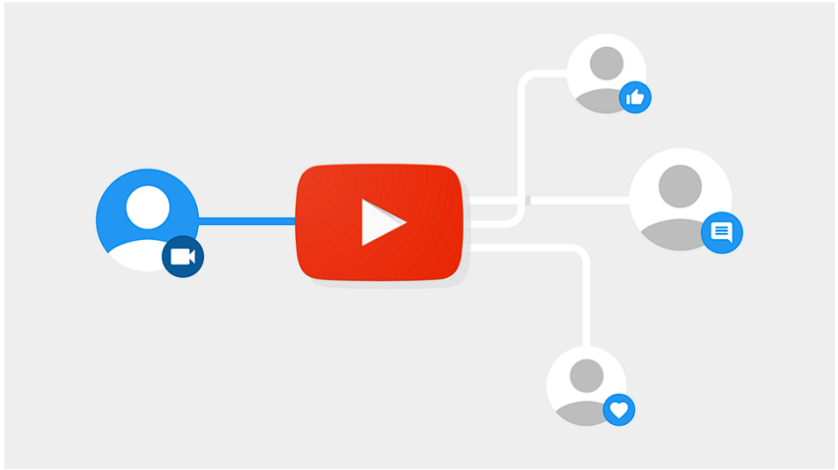
SDKs

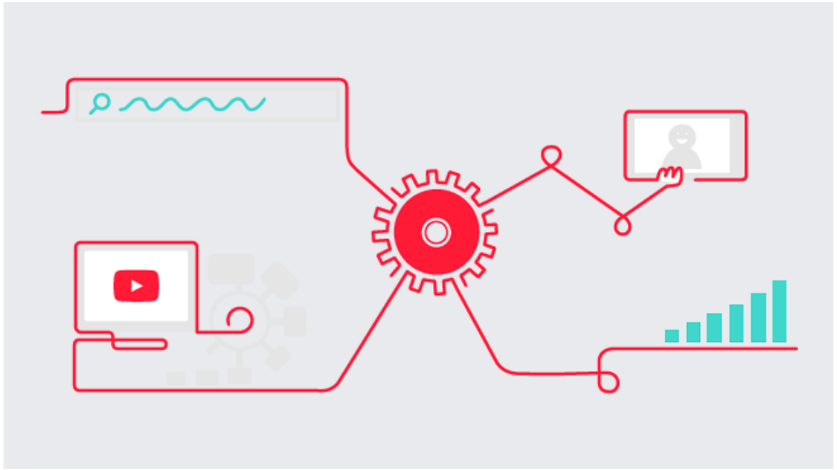
- Facebook SDK for Android
- Facebook SDK for iOS
- Facebook SDK for JavaScript
- Facebook SDK for PHP
- Facebook SDK for tvOS
- Unity SDK
- Facebook Business SDK

Publishing

- Audience Network
- Instant Articles

Add YouTube functionality to your app





Add YouTube functionality to your site

With the YouTube Data API, you can add a variety of YouTube features to your application. Use the API to upload videos, manage playlists and subscriptions, update channel settings, and more.

Get started Implementation guide

Search for content

Use the API to search for videos matching specific search terms, topics, locations, publication dates, and much more. The APIs `search.list` method also supports searches for playlists and channels.

Search for content

Other Resources

Tools

The APIs Explorer lets you test unauthorized and authorized requests. The Quota Calculator shows how different requests impact your quota usage.

APIs Explorer Quota calculator

Code Samples

Use our code samples to jump-start your project. Samples are available for Apps Script, Go, Java, JavaScript, .NET, PHP, Python, and Ruby.

Java Python More

Instagram Graph API

The Instagram Graph API allows Instagram Professionals — Businesses and Creators — to use your app to manage their presence on Instagram. The API can be used to get and publish their media, manage and reply to comments on their media, identify media where they have been @mentioned by other Instagram users, find hashtagged media, and get basic metadata and metrics about other Instagram Businesses and Creators.

The API is intended for Instagram Businesses and Creators who need insight into, and full control over, all of their social media interactions. If you are building an app for consumers or you only need to get an app user's basic profile information, photos, and videos, consider the Instagram Basic Display API instead.

The API is built on the Facebook Graph API. If you are unfamiliar with the Facebook Graph API, please read our Facebook Graph API documentation to learn how it works before proceeding.

Common Uses

- getting and managing published photos, videos, and stories
- getting basic data about other Instagram Business users and Creators
- moderating comments and their replies
- measuring media and profile interaction
- discovering hashtagged media
- discovering @mentions
- publishing photos and videos

Limitations

- The API cannot access Instagram consumer accounts (i.e., non-Business or non-Creator Instagram accounts). If you are building an app for consumer users, use the Instagram Basic Display API instead.
- Content Publishing is only available to Instagram Business Users.
- ordering results is not supported
- all endpoints support cursor-based pagination, but the User Insights edge is the only endpoint that supports time-based pagination

Documentation

Search the docs

Getting started
Fundamentals
Tools and libraries
Tutorials
API reference index

Twitter API
Getting started
Tools and libraries
What to build
Migrate
Twitter API v2
Enterprise - Gnip 2.0
Premium v1.1
Standard v1.1
Twitter Ads API

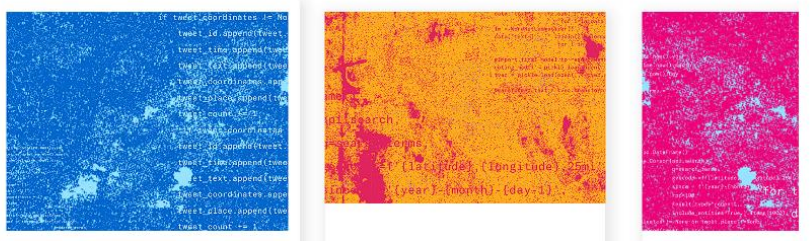
Twitter API

The Twitter API enables programmatic access to Twitter in unique and advanced ways. Tap into core elements of Twitter like: Tweets, Direct Messages, Spaces, Lists, users, and more.

Sign up | API access levels and versions | Try a live request

Twitter API v2

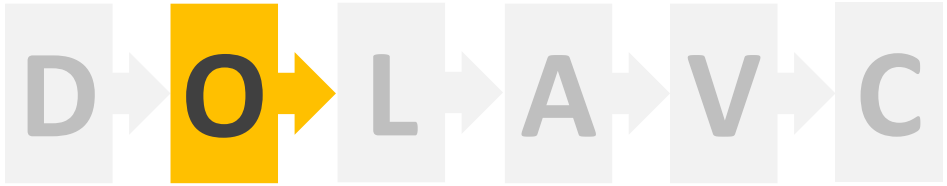
Twitter API v2 is ready for prime time! We recommend that the majority of developers start to think about migrating to v2 of the API, and for any new users to get started with v2. Why migrate?



COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

Técnicas de autenticación

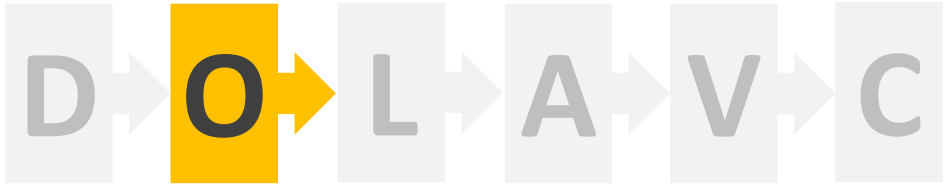
Técnicas de autenticación
OAuth



- OAuth es un protocolo de autenticación que permite a los usuarios intercambiar información con una aplicación sin necesidad de compartir la contraseña.
- Existen dos modelos principales de autenticación contra API:
  - **Autenticación de usuario**: Es el modelo más común, la petición (*request*) identifica tanto la identidad de la aplicación como los permisos otorgados al usuario final, en forma del token de acceso.
  - **Autenticación de aplicación**: En este caso las aplicaciones hacen peticiones a las APIs sin usuarios de por medio, representándose a sí mismas. Este será el modelo que empleemos en la asignatura para crear una aplicación contra cada red social.

Técnicas de autenticación

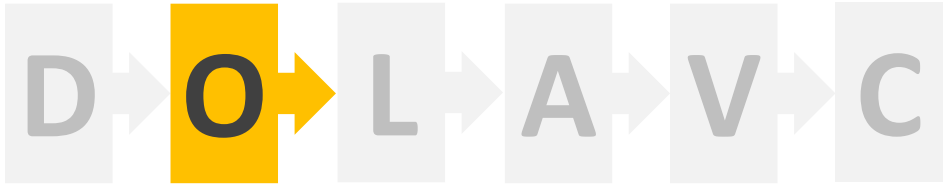
Pasos necesarios



1. **Creación de una cuenta de usuario/desarrollador**: Proporcionando información personal para la verificación.
2. **Creación de una aplicación**: Una vez creada la cuenta, se dispone de acceso a un cuadro de mandos (también llamado consola del desarrollador). Proporciona todas las funcionalidades para la gestión de la cuenta de desarrollador: crear y borrar aplicaciones, monitorizar cuotas, etc. Para la obtención de credenciales se tendrá que crear la primera aplicación mediante este interfaz.
3. **Obtención de los tokens de acceso**: Ahora se pueden generar los tokens de acceso para la aplicación y almacenarlos en un lugar seguro. Se emplearán como parte del código para crear la conexión OAuth con la API.

Técnicas de autenticación

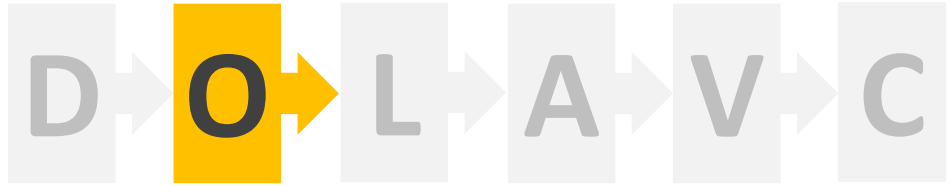
Pasos necesarios



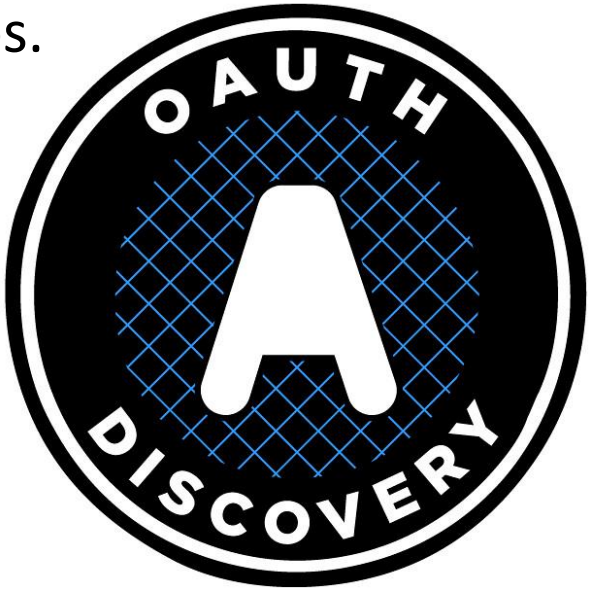
4. **Autorización de peticiones HTTP (opcional)**: Algunas APIS la requieren, que significa que la petición necesita una cabecera de autenticación adicional que proporciona al servidor información sobre la identidad de la aplicación.
5. **Configuración del alcance de permisos (opcional)**: Algunas APIs tienen permisos a varios niveles. En este caso, al generar la clave para la API debe definirse el alcance para esa clave (las acciones que se permitirán). En estos casos, cuando se intente una acción fuera de los permisos se denegará.
6. **Conexión con la API usando los tokens de acceso**: Cuando todos los pasos anteriores se hayan configurado, pueden efectuarse peticiones utilizando los tokens de acceso, quedando limitados por la cuota de peticiones.

Técnicas de autenticación

Por qué usar OAuth



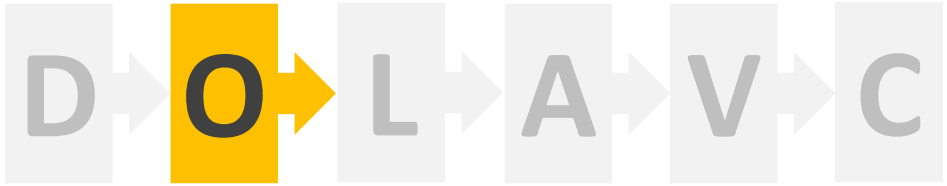
- Permite aplicaciones desatendidas por parte de los usuarios.
- El nivel de seguridad se mantiene a pesar de no incluirse contraseñas.
- Las conexiones se realizan mediante un protocolo estandarizado.
- Proporciona un servicio mejor a nivel de cuota y de endpoints disponibles.
- Se trata de la técnica más garantizada y fiable para cumplir los requisitos (política) de las empresas dueñas de las redes sociales.



Técnicas de autenticación

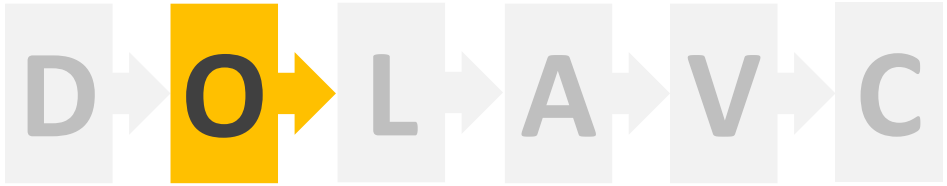
Uso práctico de OAuth

(`requests`)



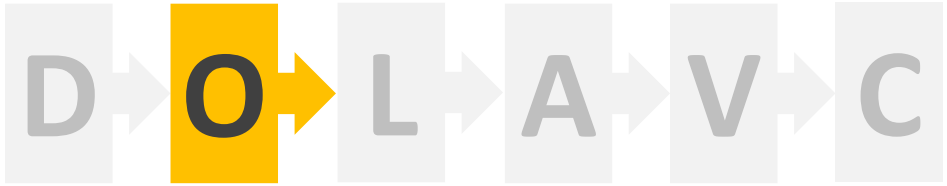
- Importamos: `import requests`
- También la librería concreta: `from requests_oauthlib import OAuth1`
- Creamos la conexión autenticada usando los tokens de acceso y las claves de aplicación creadas en la consola del desarrollador: `auth = OAuth1 ('YOUR_APP_KEY', 'YOUR_APP_SECRET', 'USER_OAUTH_TOKEN', 'USER_OAUTH_TOKEN_SECRET')`

Técnicas de autenticación
Peticiones GET y POST



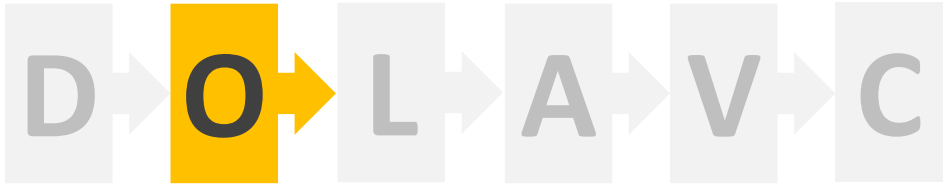
- Podemos hacer **peticiones GET**: `r = requests.get ('https://api.sampleurl.org/get', auth=auth)`
- Paso de parámetros:
  - `payload = {'key1': 'value1', 'key2': 'value2'}`
  - `r = requests.get('http://sampleurl.org/get', params=payload)`
- **Peticiones POST**: `r = requests.post('http://sampleurl.org/post', data = {'key':'value'})`

Técnicas de autenticación
Otras peticiones



- `r = requests.put('http://sampleurl.org/put', data = {'key':'value'})`
- `r = requests.delete('http://sampleurl.org/delete')`
- `r = requests.head('http://sampleurl.org/get')`
- `r = requests.options('http://sampleurl.org/get')`

Técnicas de autenticación
Parseo de los outputs



- Y para parsear los resultados podemos usar métodos como:
  - Para obtener un string: `r.text()`
  - Para obtener un JSON: `r.json()`
  - Para comprobar la codificación: `r.econding()`

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

Ejercicio:

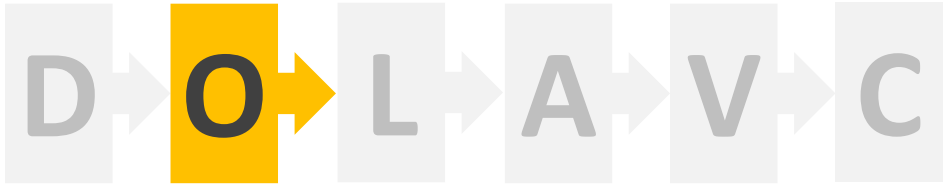
¿Qué RRSS permiten OAuth?

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

Ejemplo

https://www.digitalocean.com/community/tutorials/how-to-get-started-with-the-requests-library-in-python

Ejemplo
Instalación y petición inicial



Código

Instalación de requests (si no la tenemos):

```
pip install requests
```

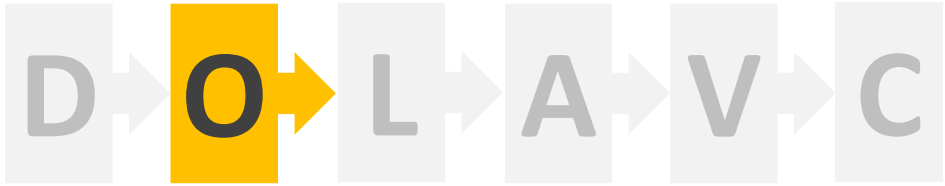
Primera request:

```
import requests

res =
requests.get('https://www.digitalocean.
com/')

print(res)
```

Terminal

```
$ python script.py
<Response [200]>
```

Ejemplo
Códigos de estado



- [1XX] - Information
- **[2XX] - Success**
- [3XX] - Redirect
- **[4XX] - Client Error (you made an error)**
- **[5XX] - Server Error (they made an error)**

`res = true`

`res = false`

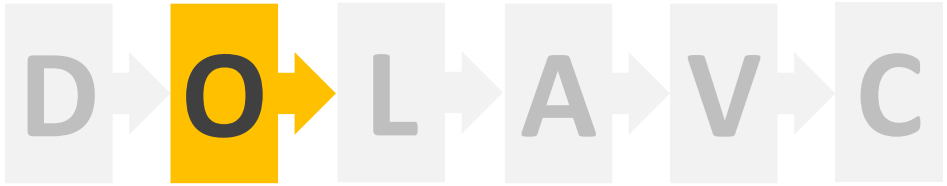
Código

```
if res: print('Response OK’)
else: print('Response Failed')
```

Terminal

```
$ python script.py
<Response [200]>
Response OK
```

Ejemplo
Cabeceras



Código

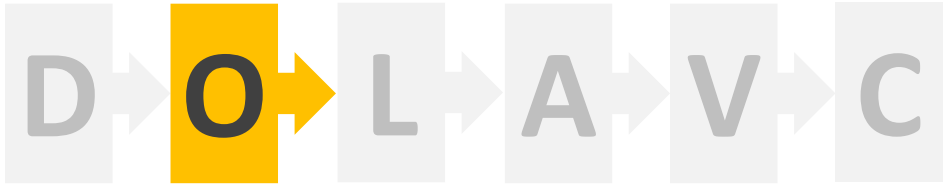
Para leer las cabeceras de la respuesta:

```
print(res.headers)
```

Terminal

```
{'Date': 'Thu, 26 Jan 2023
13:05:08 GMT', 'Content-Type':
'text/html', 'Transfer-
Encoding': 'chunked',
'Connection': 'keep-alive',
'last-modified': 'Wed, 25 Jan
2023 21:15:44 GMT', 'etag':
'W/"63d19c00-28cf8"',
'expires': 'Thu, 26 Jan 2023
13:04:12 GMT', 'cache-control':
'max-age=0, public, max-age=0,
s-maxage=300, must-revalidate',
'x-frame-options': 'DENY', 'x-
xss-protection': '1; …
```

Ejemplo
Código HTML de la página



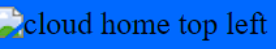
Código

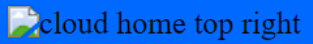
Obtención del texto de respuesta:

```
print (res.text)
```

Terminal

```
…
<!DOCTYPE html><html lang="en"
dir="ltr"><head><meta
charSet="utf-8"/><link
rel="apple-touch-icon"
sizes="180x180"
href="/_next/static/media/apple
-touch-
icon.d7edaa01.png"/><link
rel="icon" sizes="192x192"
href="/_next/static/media/andro
id-chrome-
192x192.f09059d8.png"/><link
rel="icon" sizes="512x512" …
```

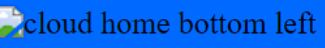


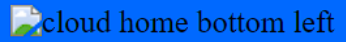


The simple cloud that drives business growth

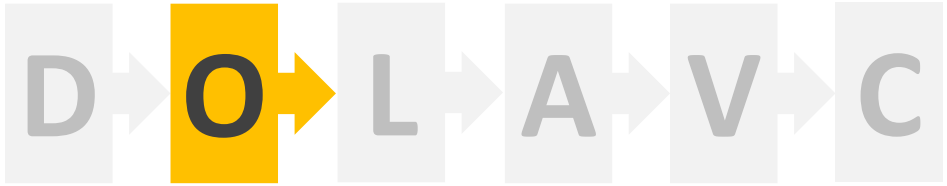
Businesses grow faster when developers can build on the simple, affordable cloud they love. DigitalOcean has the cloud computing services you need, with predictable pricing, robust documentation, and scalability to support your growth at any stage.

Sign up with Google | Sign up with GitHub | Sign up with email





Ejemplo
API de traducción de Yandex



Código

Almacenamiento de la clave:

```
API_KEY = 'your yandex api key'
```

API Yandex: Sintaxis Request

```
https://translate.yandex.net/api/v1.5/
tr.json/translate
 ? key=<API key>
 & text=<text to translate>
 & lang=<translation direction>
 & [format=<text format>]
 & [options=<translation options>]
 & [callback=<name of the callback
function>]
```

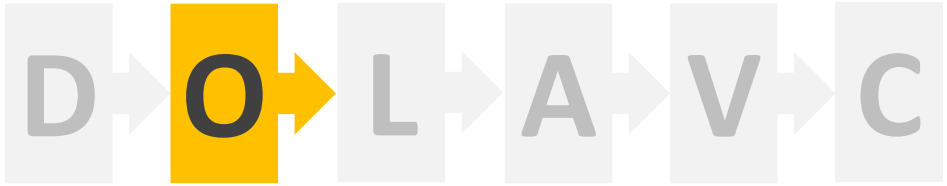
Documentación Yandex

Nueva request:

```
url = 'https://translate.yandex.net/api/v1.5/tr.json/translate'
res = requests.get(url)
```

COMILLAS UNIVERSIDAD PONTIFICIA ICAI

Ejemplo
Web traducida



Código

Los parámetros se pueden añadir al final de la URL directamente, o podemos pedirle a `requests` que lo haga por nosotros:

```
params = dict(key=API_KEY,
text='Hello', lang='en-es’)

res = requests.get(url, params=params)

print (res.text)
```

Terminal

```
…
{"code': 200, "lang": "en-es",
"text": ["Hola"]}
```

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

Parseo de datos

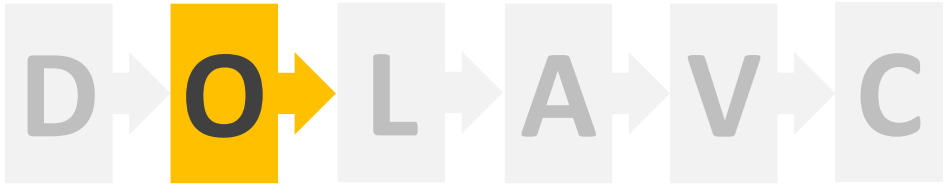
COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

Ejercicio:

Parseo de resultados

Ejercicio

Parseo de resultados



1. Crear una cuenta de desarrollador en la plataforma.
2. Crear una aplicación en la plataforma de desarrollador. Esta aplicación será a la que autorizaremos para hacer consultas de nuestros datos mediante las APIs de la plataforma.
3. Generar los tokens de acceso. Este paso requiere la salvaguarda de los tokens en un lugar seguro. Normalmente guardaremos estas claves en un fichero tipo `JSON`.
4. [*Opcional*] Autenticar las peticiones HTTP. Algunas APIs requerirán que además autoricemos las peticiones HTTP mediante una cabecera (típicamente Bearer Token).
5. [*Opcional*] Seleccionar los permisos correctos. Algunas APIs están desarrolladas con el concepto de permisos multicapa/multinivel. Si es así, nos solicitarán determinar el ámbito (scope) de validez de la clave, donde ámbito/scope no es más que un conjunto de acciones. Esta granularidad de permisos permite controlar muy bien cualquier fuga indeseada, ya que si la aplicación intentase acceder mediante a un método que no está en el ámbito en el que está autorizada, se le rechazará la petición.
6. Conectar con la API con los credenciales obtenidos, una vez los pasos anteriores han sido satisfactoriamente realizados.

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

¡Muchas gracias!