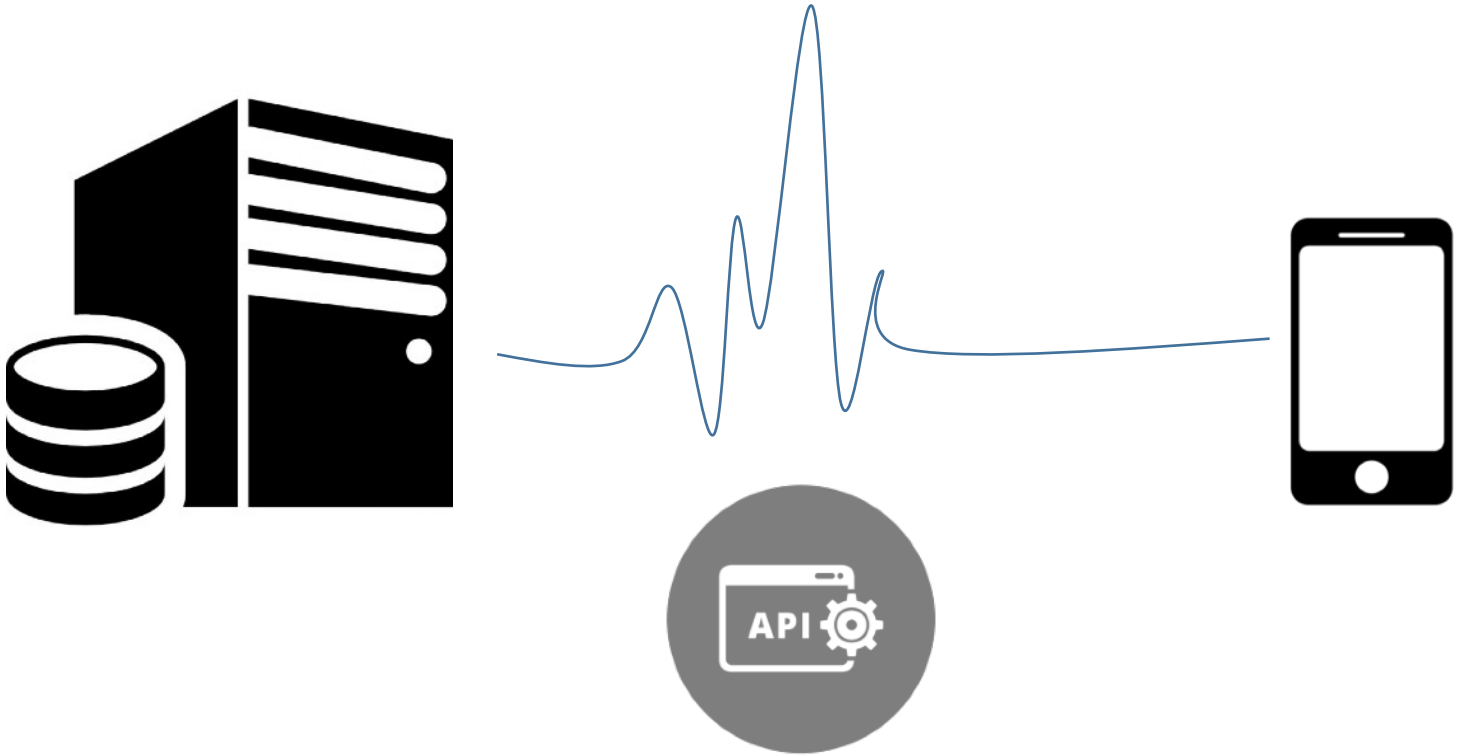
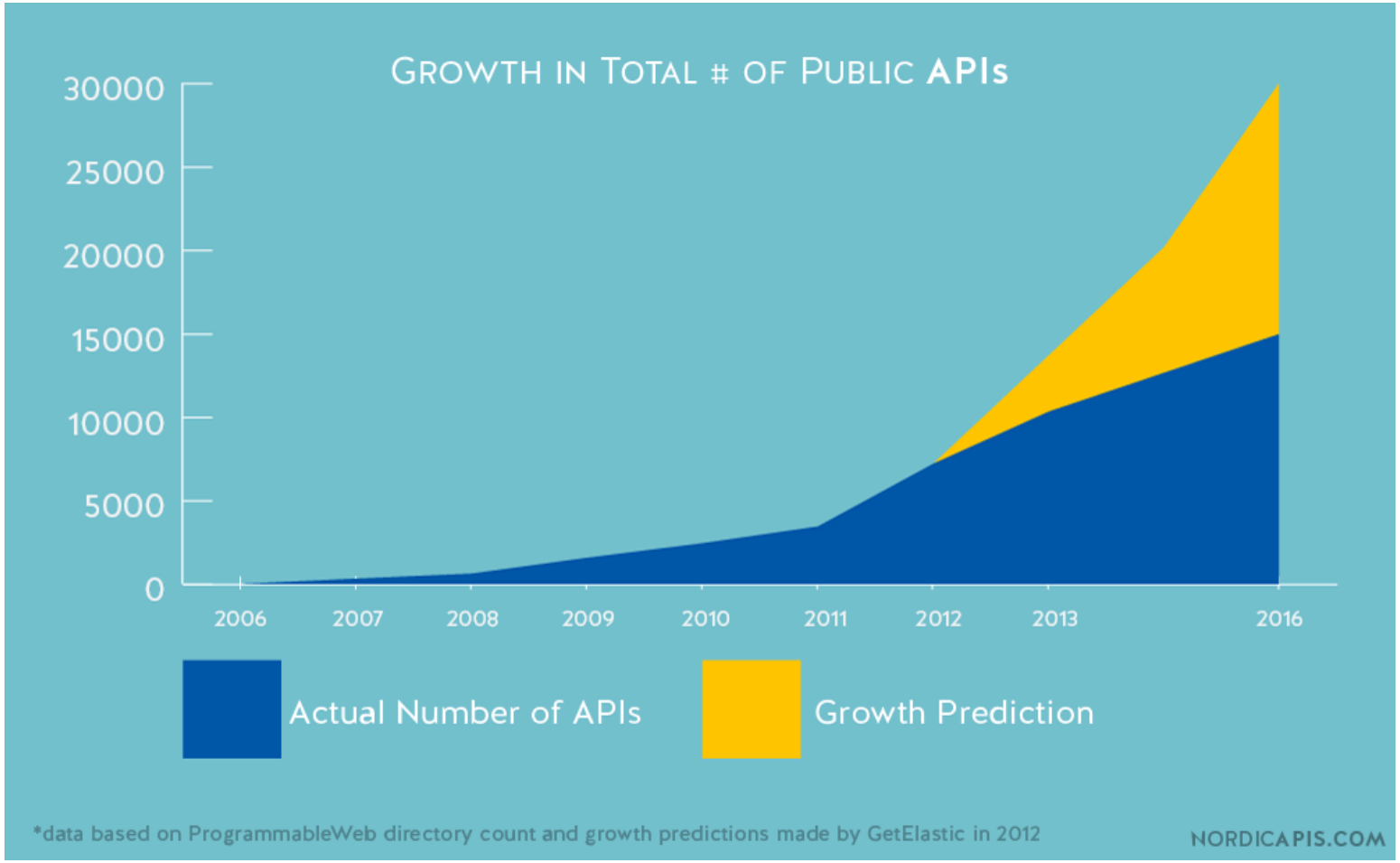
Understanding and Executing on API Developer Experience

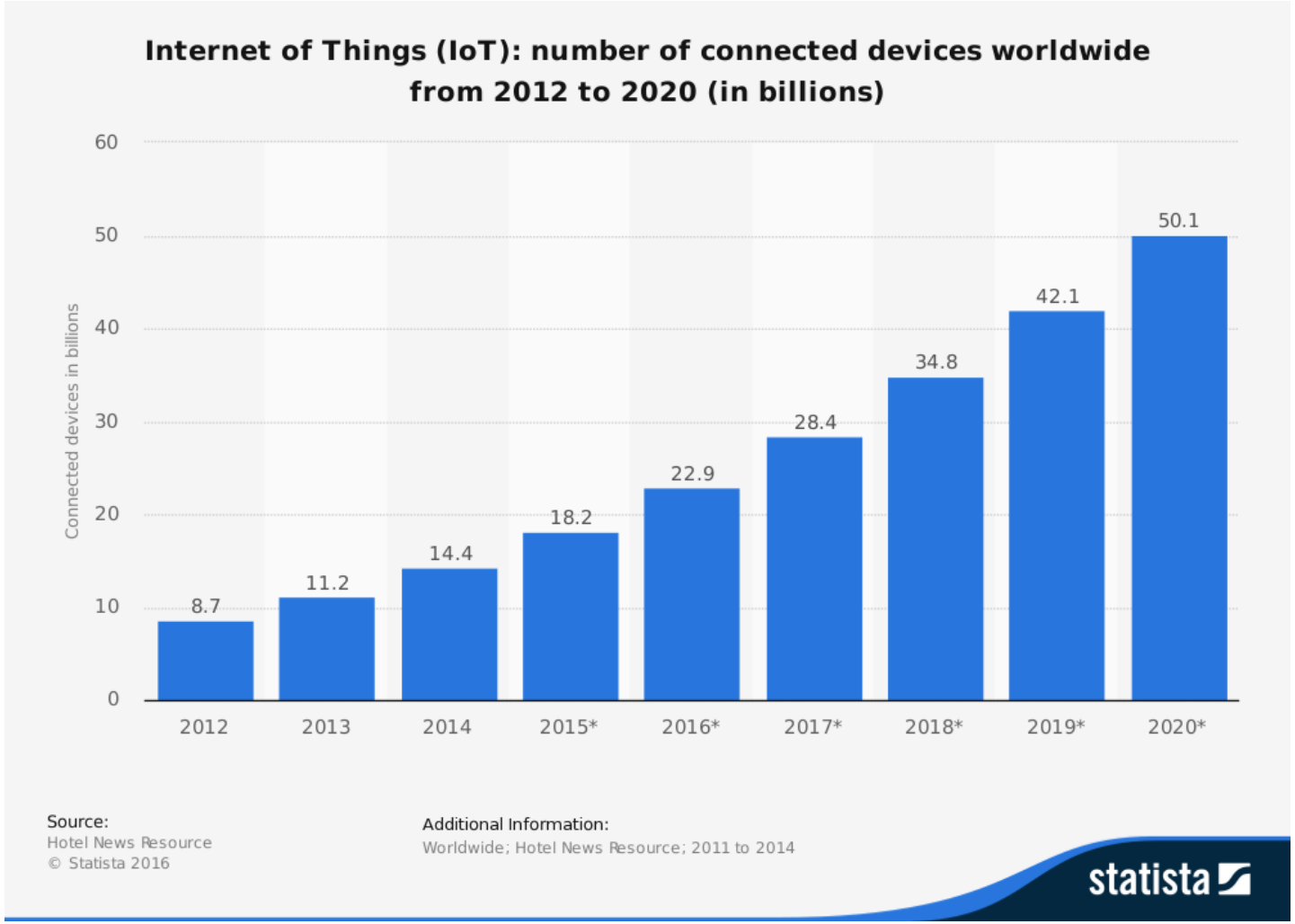
What is an API?



APIs are Everywhere!



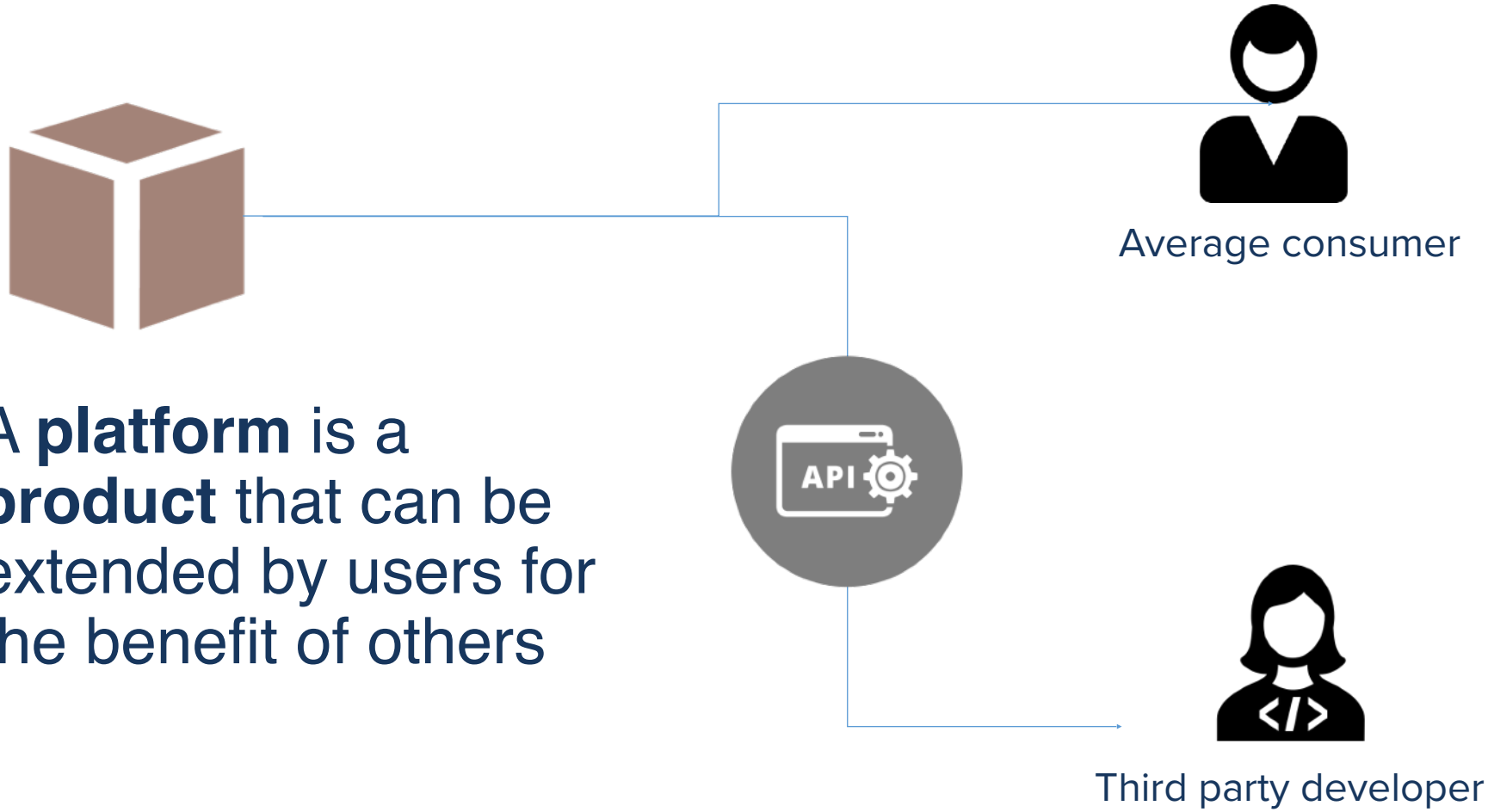
Number of Connected Devices



No. of devices/ person

7 devices

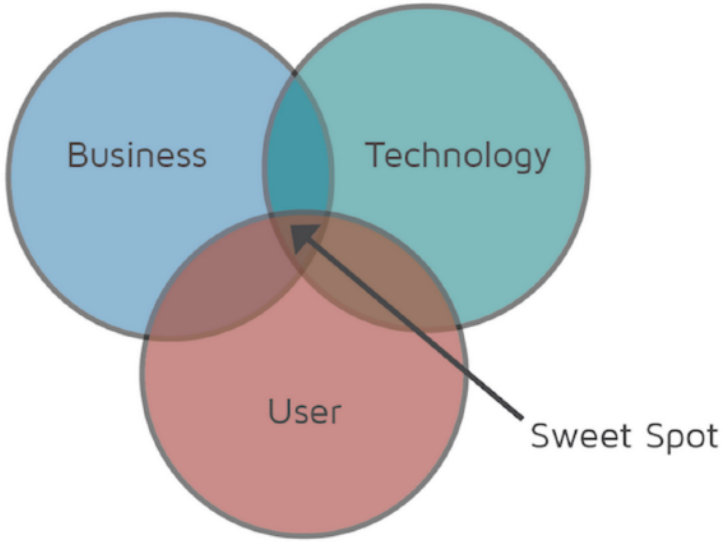
The Multi Platform Ecosystem



A **platform** is a **product** that can be extended by users for the benefit of others

Developers Are Humans Too

DX is the aggregate of all the experiences a developer has when interacting with your Platform, usually through an API



Why DX Matters

We do better with products and platforms we enjoy using

Why DX Matters

Growth and Adoption: The value of an API increases when users reach critical mass

Prevents Switch: Makes APIs differentiable and competitive

Helps drive business and technological goals

Speaking of Business and Technological Goals

Public APIs

Priority: High Adoption

Private APIs

Priority: Low Cost

Partner APIs

Priority: High Adoption, Low Cost

How To Think About DX

Goals for Good DX

Solve a Consumer Need

Help the Business

Built for sustainability

Three Step Process for Good DX

Step 1

Understand the API's audience

Step 3

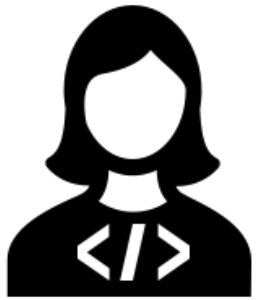
Map the audience to the journey

Step 2

Understand the audience journey

Step 1: Understand the API's Audience

API Users



Newcomer - First Time User

Debugger – addressing a specific issue

API Decision Makers

Evaluator-

Deciding whether to use the API

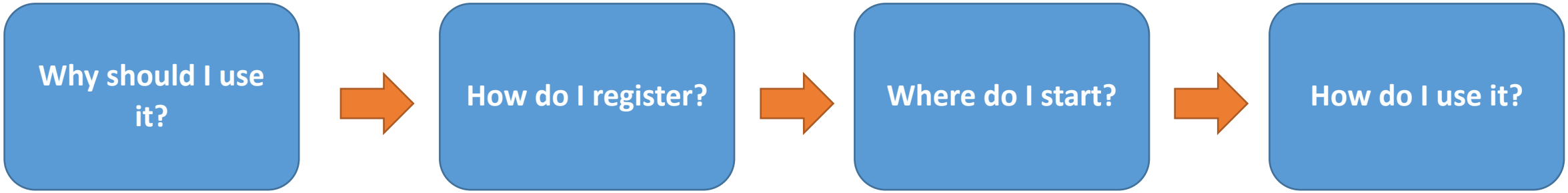
Eg: CTO

Problem Solver –

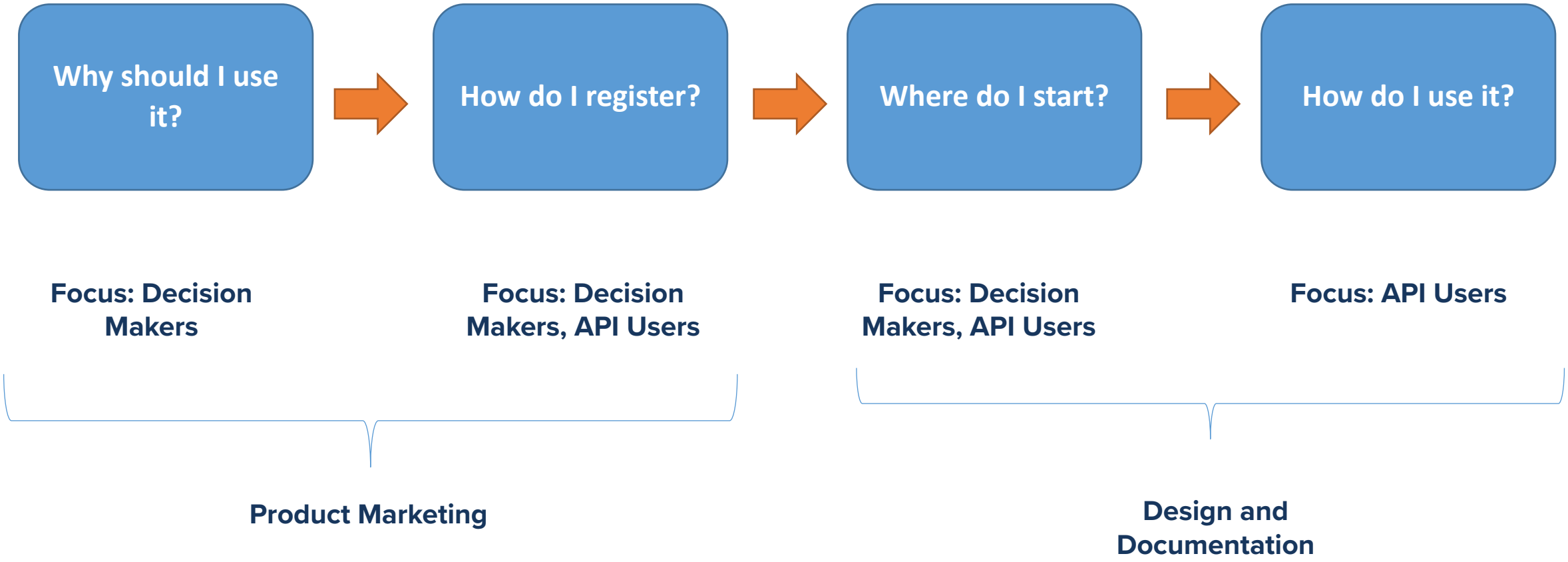
Is X possible with this API

Eg: PM

Step 2: Understand their Journey



Step 3: Map Journey to the Right Audience



Recommendations to Optimize Each Phases

Follow the 3:30:3 Rule

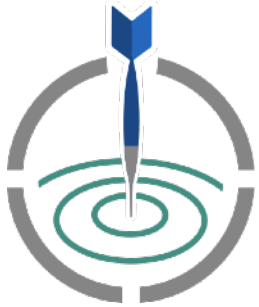
3:30:3 Rule

- 3 seconds to understand API's purpose
- 30 seconds to identify entry point into system
- 3 minutes to use the API's result

3.a. “Why Should I Use It”?

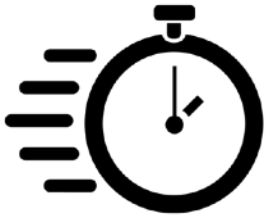
Why should I use it? → Focus: Decision Makers

Recommendations



Position based on Industry

Concise messaging of core value

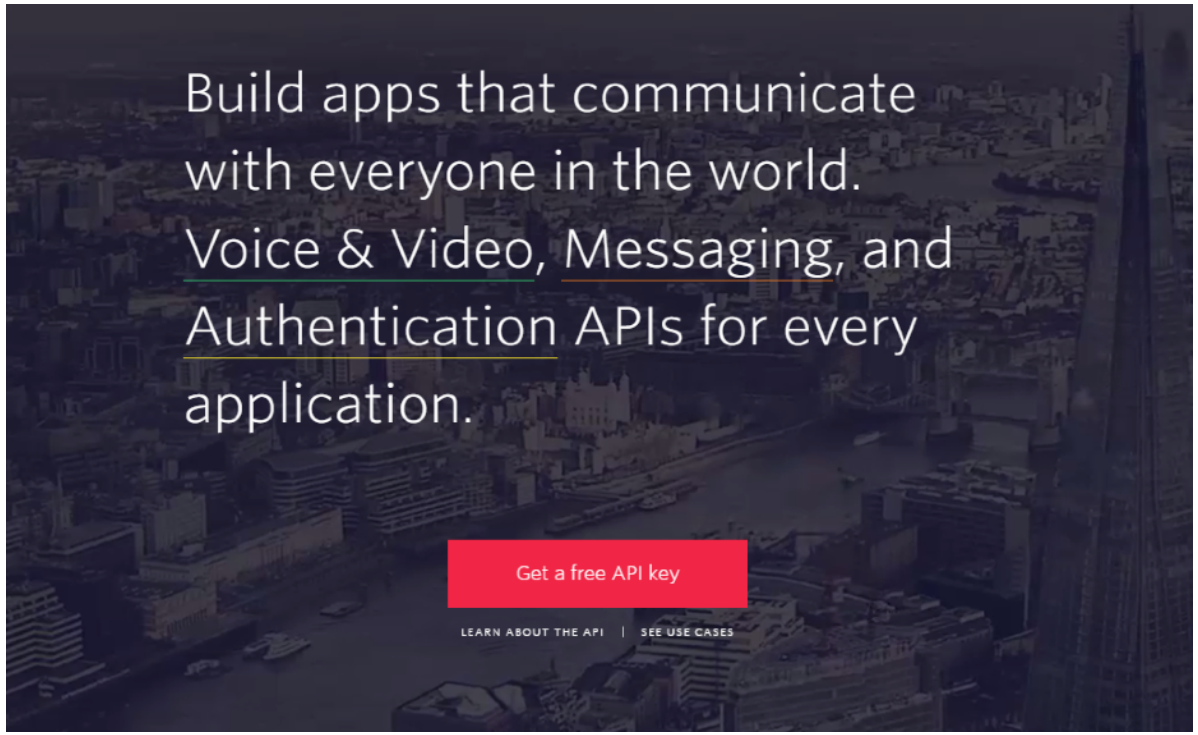


Understand value in 3 seconds

Examples

Twilio

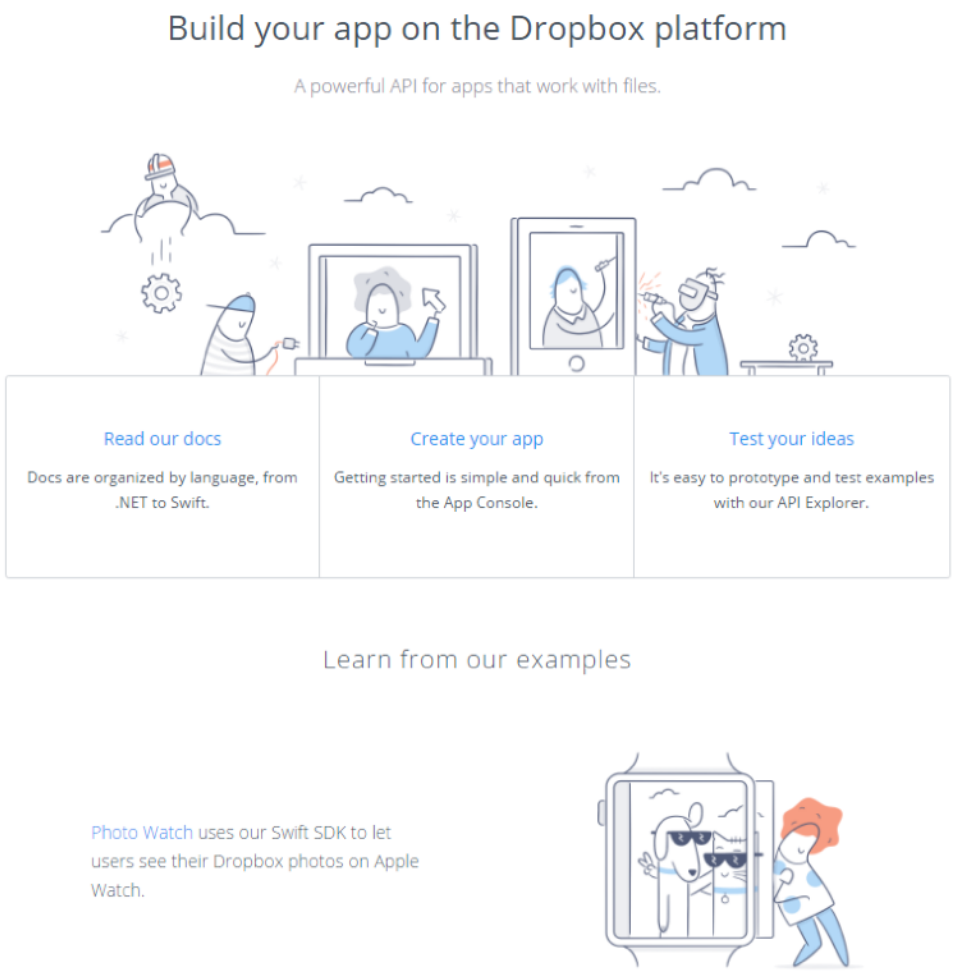
Know exactly what the value is, and what it's mean to be used for



Examples

Dropbox

Pleasant to read, easy to understand, with example use cases, all in the first section of the landing page

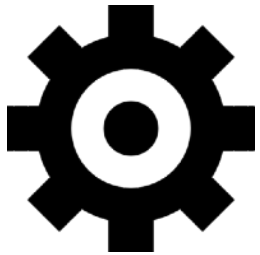


3.b. “How Should I Register?”

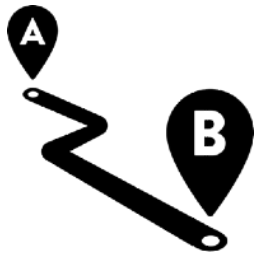
How do I register?

Focus: Decision Makers, API Users

Recommendations



Explain setup in comprehensive fashion



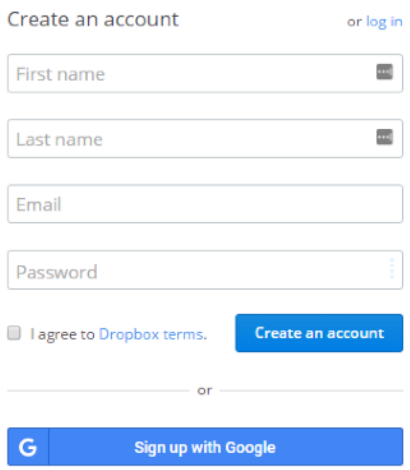
Minimal steps to register – 30 seconds

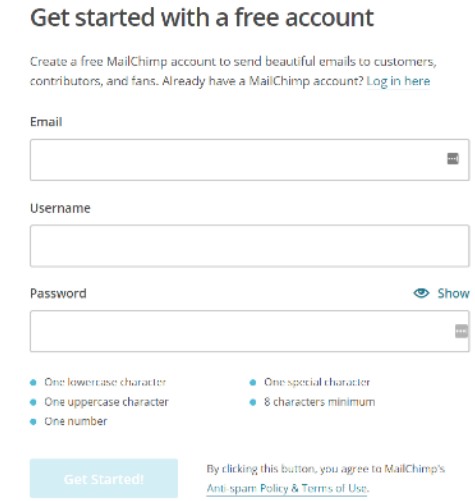
Provide API console

Examples

Make Sign up straightforward

Dropbox asks you for your name, email and password for the free API key. It also allows Google sign in. Straightforward process with Mailchimp as well.

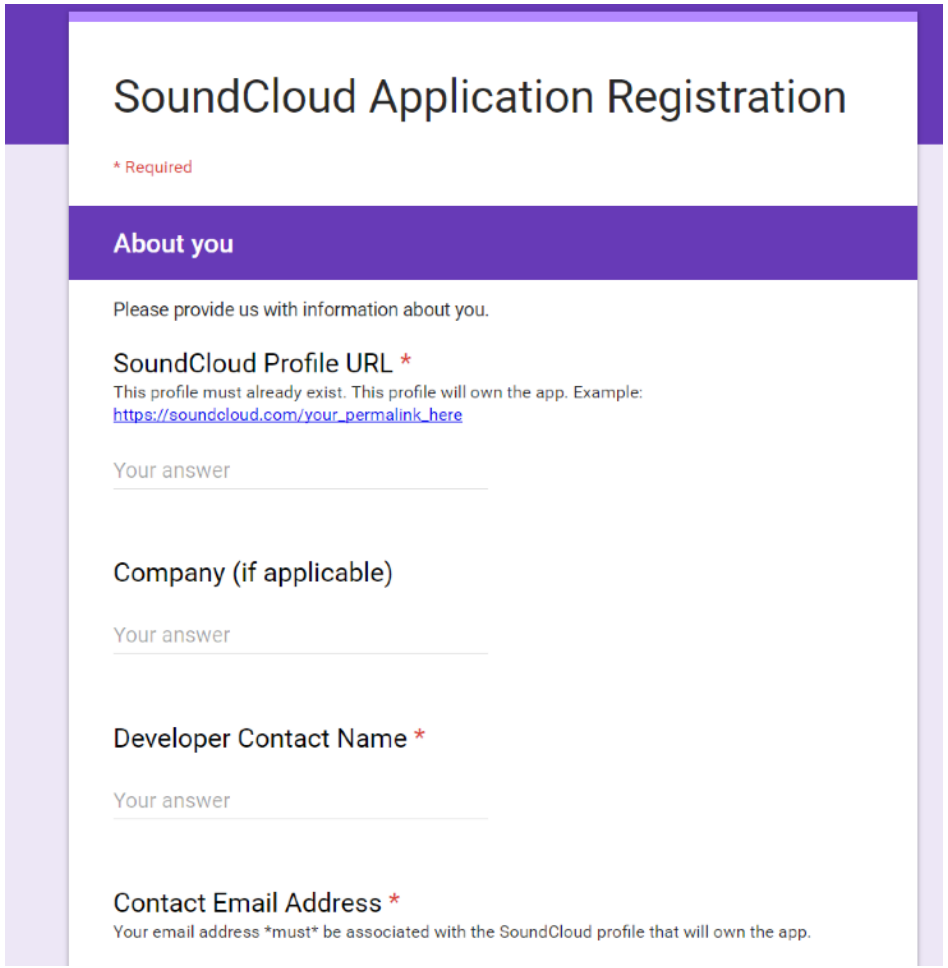




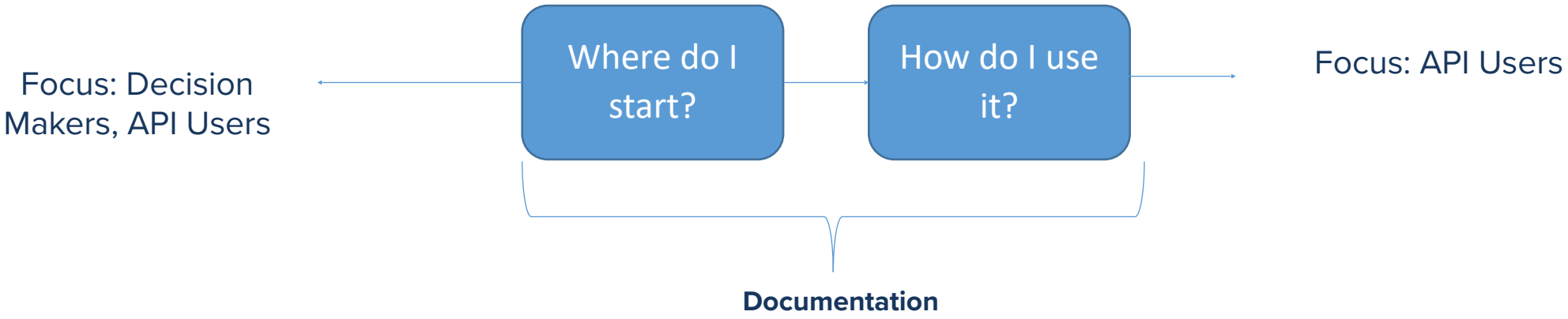
Bad Examples

Jumping through hurdles - SoundCloud

SoundCloud asks users to fill up a 3 page Google form, with details on the app idea, before awarding the API key. Waiting period is typically 2 weeks



3.c. “Where and How Do I Use it?”



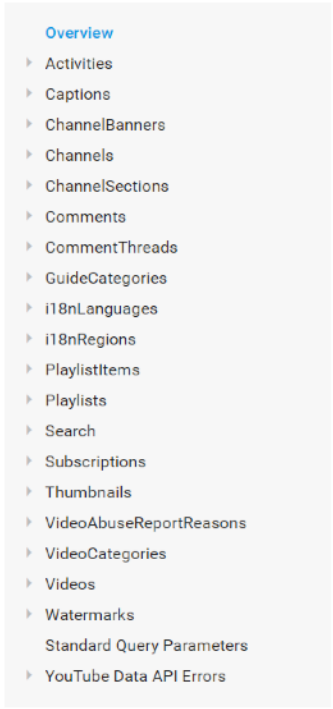
API Documentation and the OpenAPI Specification

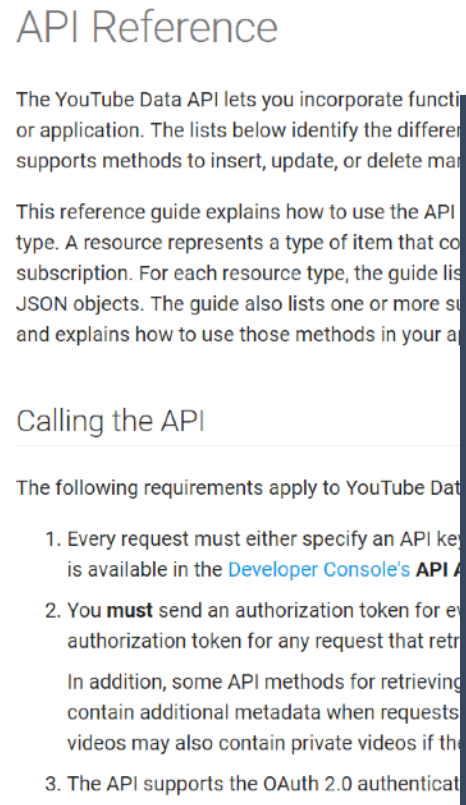
What is API Documentation

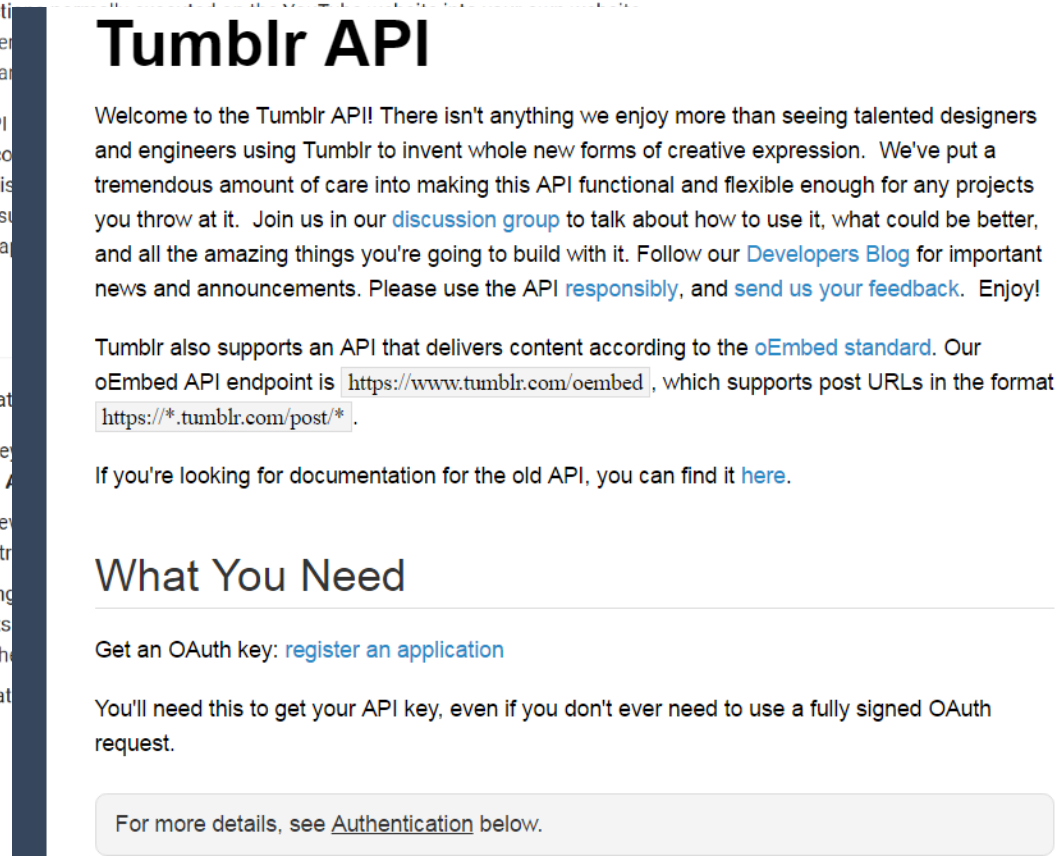
- Documentation is a deliverable of technical content, containing instructions on how to effectively use your API –> usage manual
- Good documentation can be a determining factor is API adoption and growth

An API is only as good as its documentation

Examples


Overview
Activities
Captions
ChannelBanners
Channels
ChannelSections
Comments
CommentThreads
GuideCategories
i18nLanguages
i18nRegions
PlaylistItems
Playlists
Search
Subscriptions
Thumbnails
VideoAbuseReportReasons
VideoCategories
Videos
Watermarks
Standard Query Parameters
YouTube Data API Errors

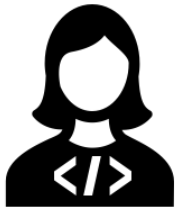

API Reference
The YouTube Data API lets you incorporate functi
or application. The lists below identify the differe
supports methods to insert, update, or delete ma
This reference guide explains how to use the API
type. A resource represents a type of item that co
subscription. For each resource type, the guide lis
JSON objects. The guide also lists one or more s
and explains how to use those methods in your a
Calling the API
The following requirements apply to YouTube Dat
1. Every request must either specify an API ke
is available in the Developer Console's API A
2. You must send an authorization token for e
authorization token for any request that retr
In addition, some API methods for retrievin
contain additional metadata when requests
videos may also contain private videos if th
3. The API supports the OAuth 2.0 authenticat


Tumblr API
Welcome to the Tumblr API! There isn't anything we enjoy more than seeing talented designers and engineers using Tumblr to invent whole new forms of creative expression. We've put a tremendous amount of care into making this API functional and flexible enough for any projects you throw at it. Join us in our discussion group to talk about how to use it, what could be better, and all the amazing things you're going to build with it. Follow our Developers Blog for important news and announcements. Please use the API responsibly, and send us your feedback. Enjoy!
Tumblr also supports an API that delivers content according to the oEmbed standard. Our oEmbed API endpoint is https://www.tumblr.com/oembed, which supports post URLs in the format https://*.tumblr.com/post/*.
If you're looking for documentation for the old API, you can find it here.
What You Need
Get an OAuth key: register an application
You'll need this to get your API key, even if you don't ever need to use a fully signed OAuth request.
For more details, see Authentication below.
Contents
What You Need
Overview
Console
Clients
Authentication
Blog Methods
Blog Info
Avatar
Likes
Following
Followers
Posts
Text
Photo
Quote
Link
Chat
Audio
Video
Answer
Queue
Drafts
Submissions

Traditional API Development and Documentation

Architect

Technical Writer



Developer

Tester

Request

Response

Major Problems

Maintaining documentation, tests and implementation is hard

Implementation comes at a cost of user experience

Communication between services built in different programming languages is not easy

RESTful Interface

Architect

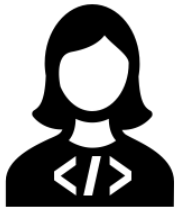
Technical Writer

Restful Interface

Call the API (request)

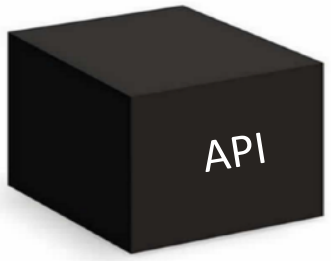
Obtain data (response)

Keeps in Sync



Developer

Tester



OpenAPI Specification is the Answer

Allow machines/tools to integrate with API

Allow humans to implement API code

Allow humans to read and generate API documentation and test cases

The OpenAPI Spec is the API Contract

- The OAS is an API description format to Design and Document REST APIs
- The OAS defines your API's contract
- Connects computers, technology stacks and humans in one unified language

Evolution of the OAS

- 2010 – Wordnik created and open sourced Swagger Specification
- 2010 to 2015 – Massive adoption of the Swagger Specification, with Swagger tooling developed by community
- 2015 - Swagger Spec and core tooling acquired by **SmartBear**

- 2015 – **Open API Initiative** was create to standardize REST design. Swagger Spec was renamed as **OAS**

Example of an OAS Defined API

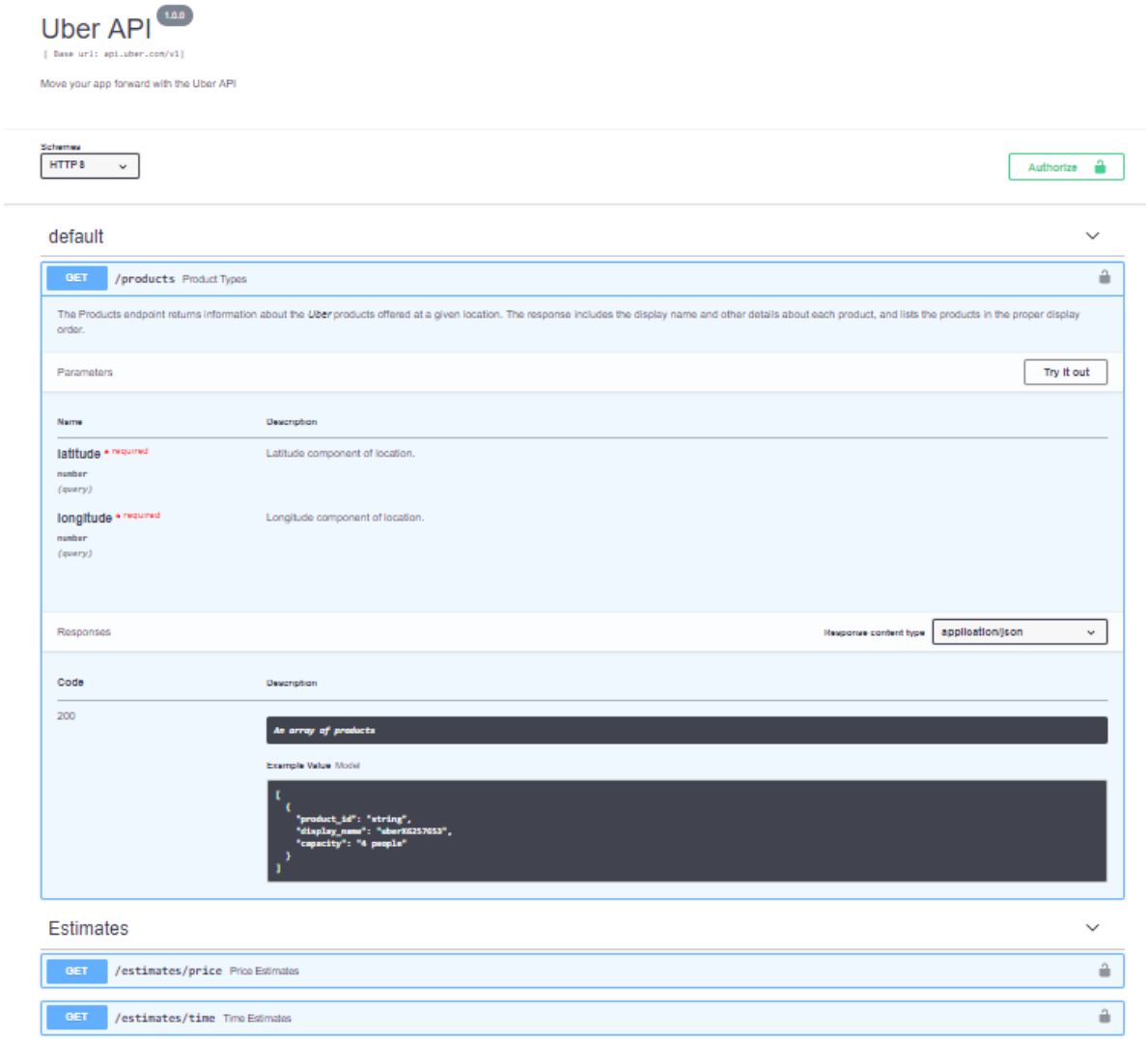
API Overview

Requests

Responses

```yaml
swagger: '2.0'
info:
  title: Uber API
  description: Move your app forward with the Uber API
  version: "1.0.0"
# the domain of the service
host: api.uber.com
# array of all schemes that your API supports
schemes:
  - https
# will be prefixed to all paths
basePath: /v1
produces:
  - application/json
paths:
  /products:
    get:
      summary: Product Types
      description: |
        The Products endpoint returns information about the *Uber* products   offered at a given location. The response includes the display name and other details about each product,
        and lists the products in the proper display order.
      parameters:
        - name: latitude
          in: query
          description: Latitude component of location.
          required: true
          type: number
          format: double
        - name: longitude
          in: query
          description: Longitude component of location.
          required: true
          type: number
          format: double
      responses:
        200:
          description: An array of products
          schema:
            type: array
            items:
                  type: object
                  properties:
                    product_id:
                      type: string
                      description: Unique identifier representing a specific product for a given latitude & longitude. For example, uberX in San Francisco will have a different product_id
                        than uberX in Los Angeles.
                    description:
                      type: string
                      description: Description of product.
                    display_name:
                      type: string
                      description: Display name of product.
                    capacity:
                      type: string
                      description: Capacity of product. For example, 4 people.
                    image:
                      type: string
                      description: Image URL representing the product.
```

Example of Documentation



Hosted on SwaggerHub

Creating Effective Documentation

Effective Documentation Tips

- Documentation plays a central role in the way APIs are perceived by a User

- Tip 1: List the Fundamentals
- Tip 2: Write for Humans
- Tip 3: Explain your Request-Response Cycles
- Tip 4: Empower with Experimentation

Tip 1: List the Fundamentals

Fundamental sections in every documentation

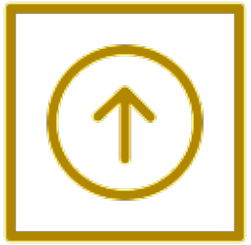
Authentication

Errors

End Points

Examples

Terms of Use



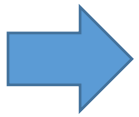
Changelog

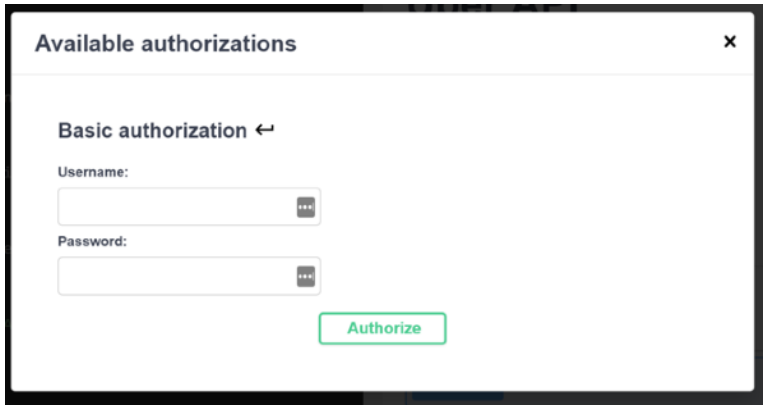
Authentication in OAS

Authorize

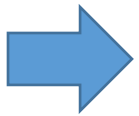
- Basic Auth

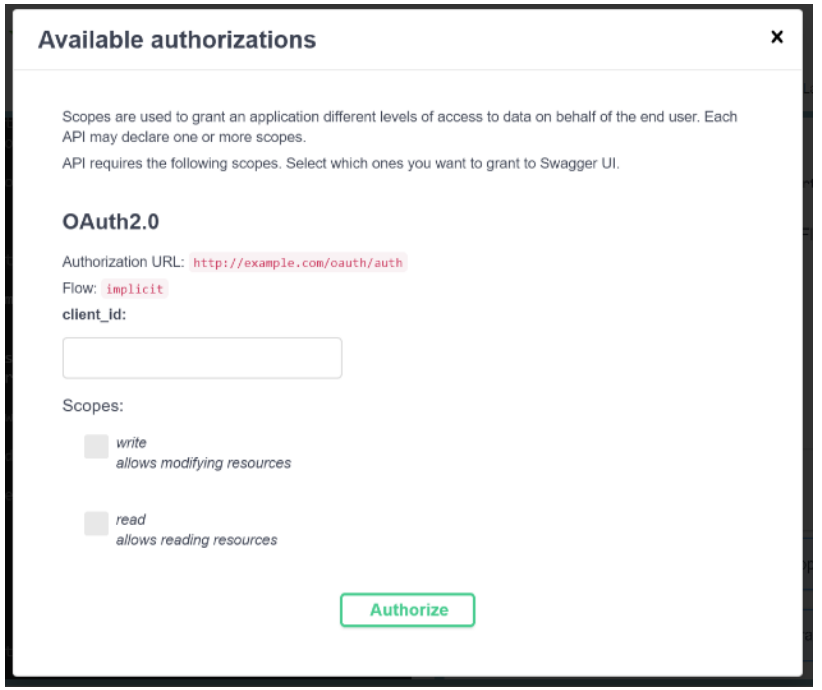
```
securityDefinitions:
  basicAuth:
    type: basic
```



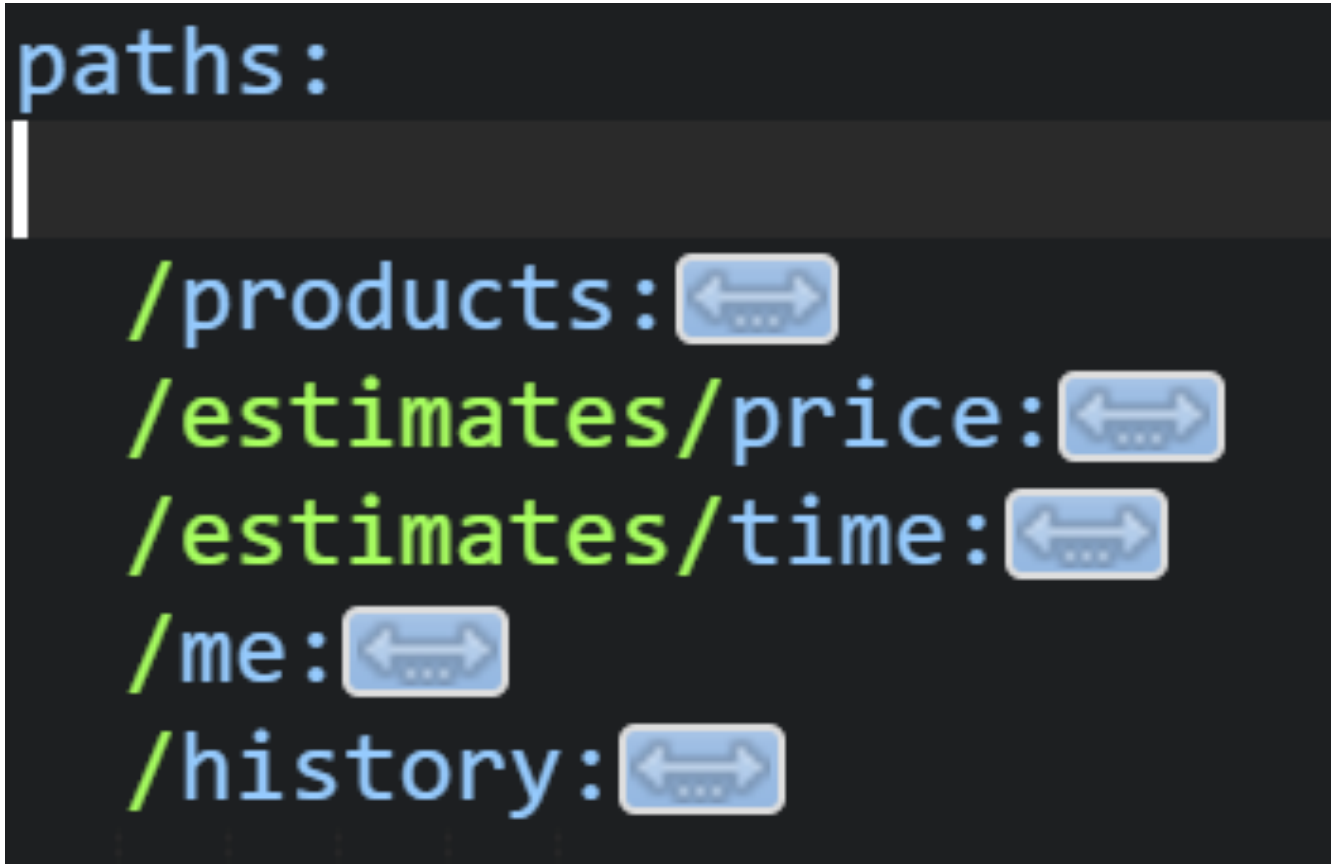


- OAuth 2
  - –Implicit
  - –Password
  - –Access Code





End Points in OAS



Errors and Examples in OAS

```yaml
responses:
  200:
    description: An array of products
    schema:
      type: array
      items:
            type: object
            properties:
              product_id:
                type: string
                description: Unique identifier representing a
                  specific product for a given latitude & longitude.
                  For example, uberX in San Francisco will have a
                  different product_id than uberX in Los Angeles.

              display_name:
                type: string
                description: Display name of product.
                example: uberX6257653
              capacity:
                type: string
                description: Capacity of product. For example, 4
                  people.
                example: 4 people
```

Changelog and Terms of Use

Box

TABLE OF CONTENTS

- 2017-07-13 Token Exchange
- 2017-06-07 Recent Items API
- 2017-06-05 Chunked Upload API
- 2017-06-01 New Metadata Template Editing Options; Changes to Enterprise Events API
- 2017-05-03 Auto-Generation of RSA Key Pairs and Exportable App Settings
- 2017-05-03 The New Box Developer Console is Now Generally Available
- 2017-05-01 Deprecation: TLS 1.0
- 2017-04-28 Increase in Metadata Template Limit for an Enterprise

Google

Google APIs Terms of Service

Last modified: December 5, 2014 (view archived version)

Thank you for using Google's APIs, other developer services, and associated software (collectively, "APIs"). By accessing or using our APIs, you are agreeing to the terms below. If there is a conflict between these terms and additional terms applicable to a given API, the additional terms will control for that conflict. Collectively, we refer to the terms below, any additional terms, terms within the accompanying API documentation, and any applicable policies and guidelines as the "Terms." You agree to comply with the Terms and that the Terms control your relationship with us. So please read all the Terms carefully. If you use the APIs as an interface to, or in conjunction with other Google products or services, then the terms for those other products or services also apply.

Under the Terms, "Google" means Google Inc., with offices at 1600 Amphitheatre Parkway, Mountain View, California 94043, United States, unless set forth otherwise in additional terms applicable for a given API. We may refer to "Google" as "we", "our", or "us" in the Terms.

Section 1: Account and Registration

Tip 2: Write for Humans

Never Assume

Audience is 100% developers

Consumers are fully familiar with API/ Domain Jargon

- Strive to make documentation readable by Decision Makers
- Start writing in plain English without jargon
- Provide context and information for jargon and domain specific words
- Maintain consistency of tone

Examples

Coupons

A coupon contains information about a percent-off or amount-off discount you might want to apply to a customer. Coupons only apply to invoices; they do not apply to one-off charges.

The coupon object

ATTRIBUTES

id
string

object
string , value is **"coupon"**

Invoices

Invoices are statements of what a customer owes for a particular billing period, including subscriptions, invoice items, and any automatic proration adjustments if necessary. Note that invoices at Stripe are part of the recurring billing process and are not intended for one-time charges.

Once an invoice is created, payment is automatically attempted. Note that the payment, while automatic, does not happen exactly at the time of invoice creation. If you have configured webhooks, the invoice will wait until one hour after the last webhook is successfully sent (or the last webhook times out after failing).

Any customer credit on the account is applied before determining how much is due for that invoice (the amount that will be actually charged). If the amount due for the invoice is less than 50 cents (the minimum for a charge), we add the amount to the customer's running account balance to be added to the next invoice. If this amount is negative, it will act as a credit to offset the next invoice. Note that the customer account balance does not include unpaid invoices; it only includes balances that need to be taken into account when calculating the amount due for the next invoice.

Tip 3: Explain your Request-Response Cycles

- Your API Users should know exactly what to expect from a successful call
- Describe full sample response body in every supported format
- Focus not just on the response format, but also HTTP response headers and error codes
- Provide enough context using examples
    1. Examples in Use Cases
    2. Examples in Request-Response Cycles

Examples

HTTP status code summary

200 - OK	Everything worked as expected.
400 - Bad Request	The request was unacceptable, often due to missing a required parameter.
401 - Unauthorized	No valid API key provided.
402 - Request Failed	The parameters were valid but the request failed.
404 - Not Found	The requested resource doesn't exist.
409 - Conflict	The request conflicts with another request (perhaps due to using the same idempotent key).
429 - Too Many Requests	Too many requests hit the API too quickly. We recommend an exponential backoff of your requests.
500, 502, 503, 504 - Server Errors	Something went wrong on Stripe's end. (These are rare.)

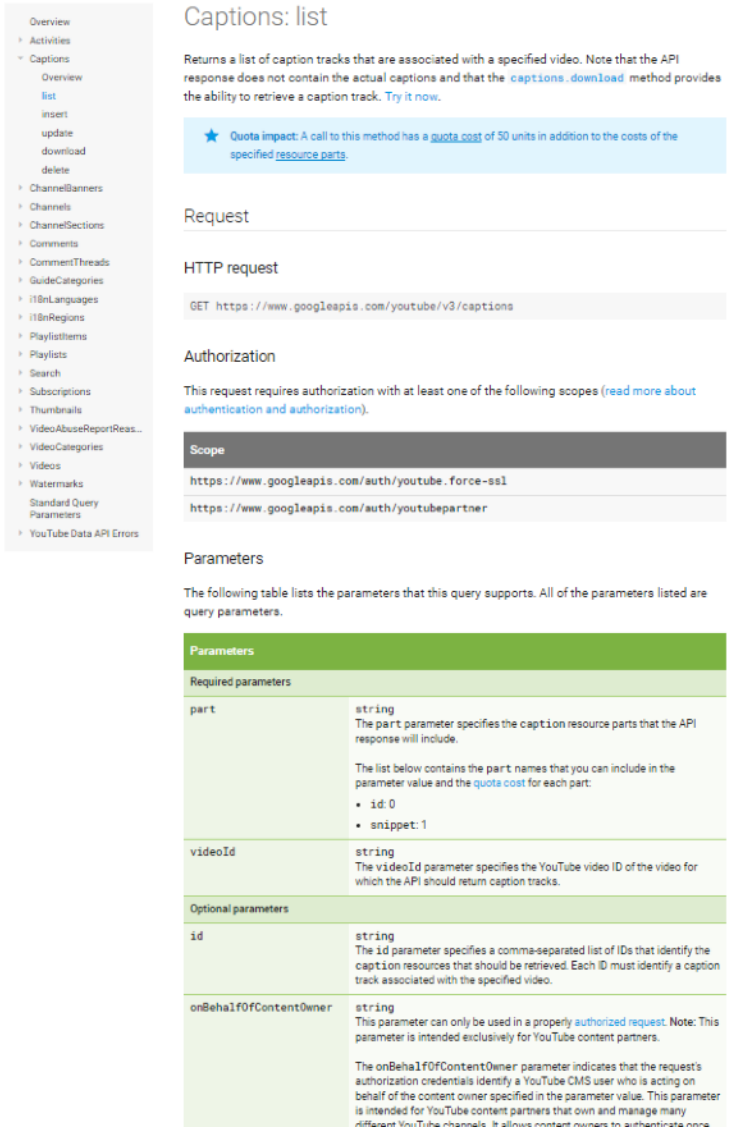
Stripe

All the Error Codes are described in a separate section for users to follow

Examples

YouTube

Well detailed parameters, with an overview section of every resource. These resources are all organized in an easy to follow navigation pane



Tip 4: Experimentation is Power

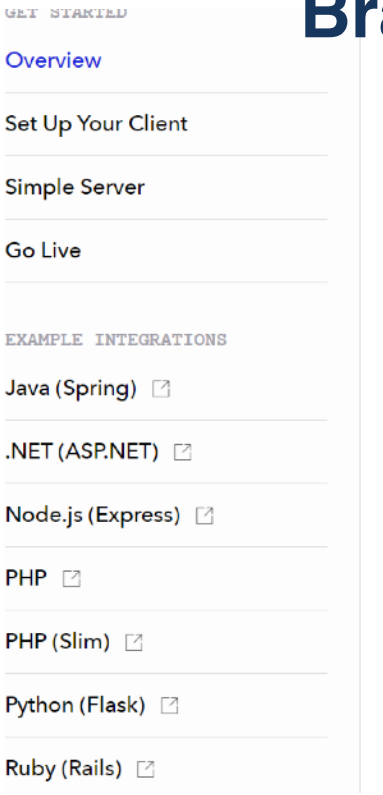
The difference between documentation and good documentation is a little more effort with these Nice-to-haves.

Allow developers to experiment with your API

1. Getting Started Guides
2. Interactive Docs and Console
3. SDKs
4. Tutorials

Examples

Braintree



Get Started

Braintree provides complementary client and server SDKs to complete your integration:

- The client SDKs enable you to collect payment method (e.g. credit card, PayPal) details
- The server SDKs manage all requests to the Braintree gateway

Before we get started, there are two key concepts to introduce: the client token and the payment method nonce.

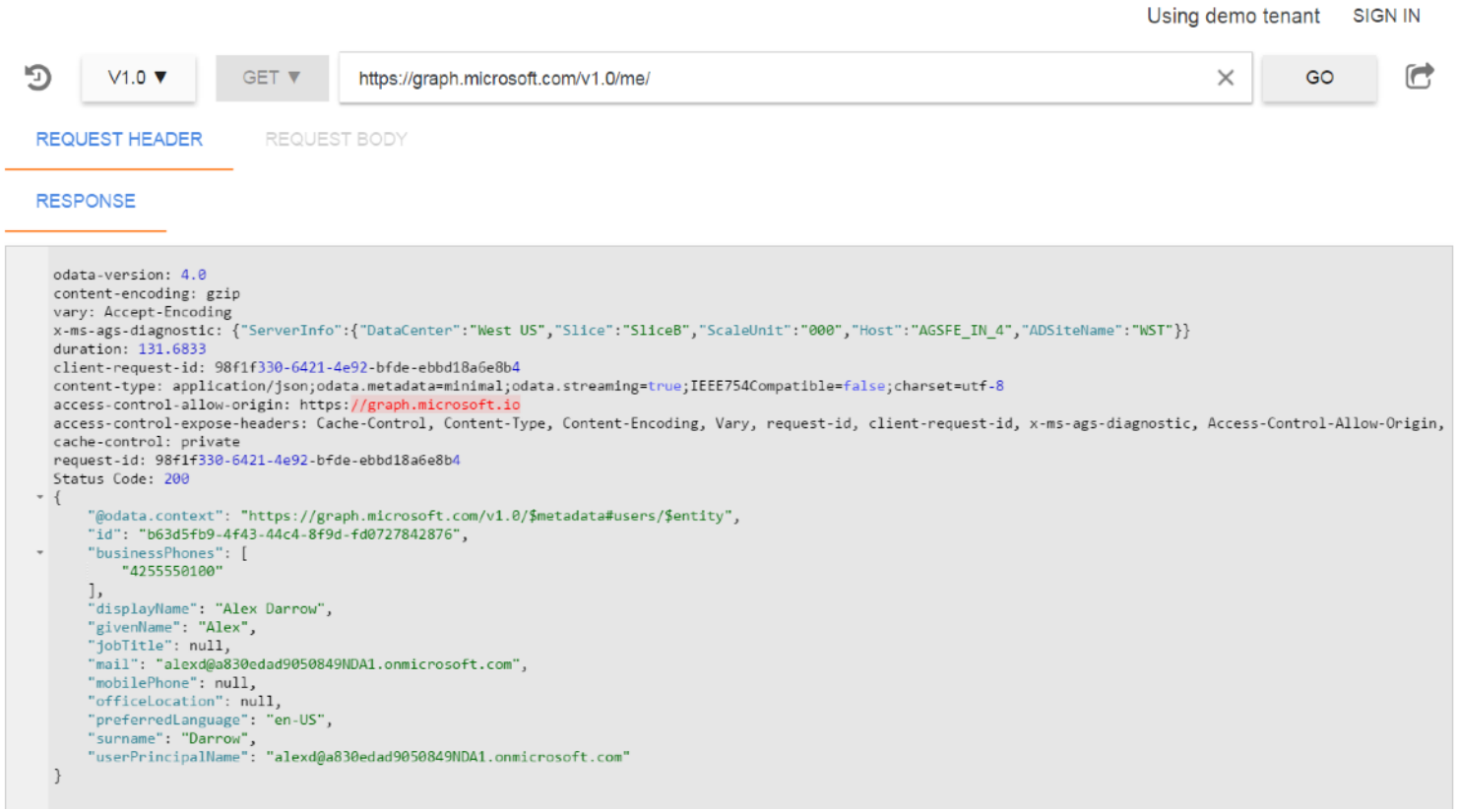
Client token

A client token is a signed data blob that includes configuration and authorization information required by the Braintree client SDK. These should not be reused; a new client token should be generated for each request that's sent to Braintree. For security, we will revoke client tokens if they are reused excessively within a short time period.

Examples

API console

Microsoft Graph Explore lets you try various end points and see what the response packet is



Summary

- Good API Developer Experience can help drive business and tech goals
- Three step framework for understanding API DX
  - Understand audience
  - Understand journey
  - Map journey to audience
- Tips for Good API Documentation
  - List the Fundamentals
  - Write for Humans
  - Explain your Request-Response Cycles
  - Empower with Experimentation

Questions?