



Cabinet Office

Rosalie Marshall

Technical Author

Government Digital Service

@Rosaliemarshall

This presentation is in **4**
parts...

- 1. Why this research?**
- 2. Understanding users**
- 3. Delivering on research**
- 4. Next steps**

1. Why this research?

The **Government Digital Service** is
building **Government as a Platform**

We've also been improving how services are delivered

“We don’t want to be building a kitchen every time we need to have a meal,”
GDS Director of Digital, Chris Ferguson.



Government as a Platform

Good API
documentation is ~~a~~
~~waste of time~~ essential
to the success of
Government as a
Platform and code
sharing more generally





@z3r0fox



Follow

Development: 10% inspiration, 90% trying to decipher poorly written/incomplete API documentation.

RETWEETS

4

LIKES

5



2:26 PM - 9 Mar 2016



Reply to @z3r0fox

2. Understanding users

Content

Design

Writing Style

Who are the users of API docs?

1. Service Managers
2. Technical Architects
3. Developers

20 out of 24 service managers
said they never read API
documentation..

We quizzed 30 technical architects
and developers on what they
wanted from API documentation

We looked at API documentation from a range of different places asking them how it could be improved...

August 4, 2015

IER: INDIVIDUAL ELECTORAL REGISTRATION.

IER EMS CLIENT VERIFICATION API SPECIFICATION

*Version: 2.**

*Release date: ******

[View the release notes](#) for the current release:

CONTENTS

1. [Intended Audience](#)
2. [Conventions](#)
3. [Domain Language](#)
4. [Using the IER API](#)
5. [Using the IER Movers API](#)
6. [Notes](#)



BETA This is a new service.

[Forum](#) [Sign in / Register](#)

Companies House API

- [Getting started](#)
- [Quick start](#)
- [Overview](#)
- [Introduction](#)
- [Authorisation](#)
- [Developer guidelines](#)
- [Rate limiting](#)
- [Search](#)
- [Company profile](#)
- [Registered office address](#)
- [Officer list](#)
- [Filing history](#)
- [Insolvency](#)
- [Charges](#)
- [Officer Appointment List](#)
- [Officer disqualifications](#)
- [Document API](#)

Companies House API

Beta release

The Companies House API and Developer Hub are currently at a [beta release](#). We are going through a process of user engagement to improve the API and documentation.

We have a proposed approach to versioning the API and documentation that is currently being [discussed in the Developer Hub Forum](#). In the meantime existing resources will remain 'as is' and can be considered as at 'version 0'.

Please refer to the [Developer Hub Forum](#) for ongoing feature and documentation discussions, and announcements on upcoming feature releases.

Getting started

If you are new to the Companies House API, we recommend you take a



Thanks for supporting Read the Docs!

Get a Performance Platform dashboard

This page explains how to get a dashboard on the [Performance Platform](#).

A published dashboard is a requirement of the [Digital by Default Service Standard](#).

Before you start

Please discuss your dashboard requirements with whoever will be signing off the publication of the dashboard as soon as possible.

How to get a dashboard

A standard dashboard for transactional services includes [four mandatory KPIs](#).

- Cost per transaction (the average cost to the government of each completed transaction)
- User satisfaction (how satisfied users feel after completing a transaction online)
- Completion rate (the percentage of users who complete a digital transaction once they have started it)
- Digital take-up (percentage of all transactions which are completed using digital channels)

How to get a transactional services dashboard

To get a dashboard, email the following information to performance@digital.cabinet-office.gov.uk. You can use the same address for further enquiries about dashboards and the Performance Platform.

About this guide

This is the technical reference guide for GOV.UK Verify.

This guide contains information to help teams who are building secure digital services to integrate with GOV.UK Verify, and should be used together with the [GOV.UK Onboarding Guide](#).

GOV.UK Verify is currently in beta. This guide will be continually improved as we take the service from beta to live. Check the [version history](#) for the latest changes to the guide.

Who this guide is for

This guide is for technical delivery teams who are building secure digital services that integrate with GOV.UK Verify.

Next 

[Home](#) >

Police API Documentation

The API provides a rich data source for information, including:

- Neighbourhood team members
- Upcoming events
- Street-level crime and outcome data
- Nearest police stations

The API is implemented as a standard JSON web service using HTTP GET and POST requests. Full request and response examples are provided in the documentation.

[Authentication](#)

Methods

[Availability](#)

Force related:

[Forces](#)

[Specific force](#)

[Force senior officers](#)

API stands for Application Programming Interface, literally specifying how software components should interact with each other. In our context, this means we have endpoints that return different kinds of data to developers using JSON objects, allowing them to use and manipulate it for their own applications.

Bottom line is; this service creates a lot of opportunities for developers that don't necessarily have the know-how, time or resources to get the data themselves, but are full of ideas for using it. So if you have an idea that only needs a little more data to become a reality, don't hesitate to contact us or even contribute yourself to our github repository.

What about version control?

At the moment, including an `accept-version` header is not necessary, since all endpoints are at version 1, **but we strongly recommend it**, especially in production environments. Once an endpoint has been updated, the default version to be returned is the most recent, and this documentation will be updated accordingly. Changes in functionality between versions will be outlisted.

information on individual endpoints visit our github repository and look at the source code.

How to get started?

This documentation includes simple jQuery.ajax demos. Feel free to copy these to your projects, but remember that they are provided as is, and you will most certainly have to adjust them to your own needs.

How is all this possible? And for free?

APIs.is is completely open source and developed by several contributors working on their own personal time. All this runs in the cloud, available to you at all times.



Icelandic Addresses



Icelandic Bus System



Cars in Iceland



Icelandic companies

Data

[By Country](#) [By Topic](#) [Indicators](#) [Data Catalog](#) [Microdata](#) [Initiatives](#) [What's New](#) [Support](#) [Products](#)This page in [English](#) | [中文](#)

API Documentation

[+ SHARE](#)

How to use the API

The pages listed on the right hand side of this page describe the different types of requests you can make against the API. Besides making calls to the API using an application or custom program, you can also put the "Example calls" URLs listed in the documentation or your own custom calls into a web browser and view the results. If you choose to receive the result in JSON format you can use the JSON View Firefox plugin for easily viewing JSON results directly in Firefox.

There are also third party applications and libraries that can make using the API easier depending on your objectives. See [Application Showcase](#) for more details.

API Access / Authentication

Until recent it was necessary to use an API key to make calls to the API. This has changed. You no longer need an API key.

About this documentation

The majority of the documentation in this section, the contents of which you can see to the right side of the page, explains the different types of call that can be made and what they're for, how to structure a specific call, what the response looks like, and some examples. More general rules about query call structures can be found [here](#)

For Developers

- **API Documentation**
 - Basic Call Structure
 - Catalog Sources Queries
 - Country Queries
 - Income Level Queries
 - Indicator Queries
 - Lending Type Queries
 - Topic Queries
 - Aggregates - Regions and Income Levels
 - Error Codes
- API Sources
- App Competitions
- Application Showcase
- Apps for Development Competition
- Data Catalog API

Our user research also involved card sorting exercises and diary studies.

Lastly we reached out to
some companies producing
great API docs..

API Reference

The Stripe API is organized around [REST](#). Our API has predictable, resource-oriented URLs, and uses HTTP response codes to indicate API errors. We use built-in HTTP features, like HTTP authentication and HTTP verbs, which are understood by off-the-shelf HTTP clients. We support [cross-origin resource sharing](#), allowing you to interact securely with our API from a client-side web application (though you should never expose your secret API key in any public website's client-side code). [JSON](#) is returned by all API responses, including errors, although our [API libraries](#) convert responses to appropriate language-specific objects.

To make the API as explorable as possible, accounts have test mode and live mode API keys. There is no "switch" for changing between modes, just use the appropriate key to perform a live or test transaction. Requests made with test mode credentials never hit the banking networks and incur no cost.

Authentication

Authenticate your account when using the API by including your secret API key in the request. You can manage your API keys in the [Dashboard](#). Your API keys carry many privileges, so be sure to keep them secret! Do

API libraries

Libraries for the Stripe API are [available in several languages](#).

API Endpoint

```
https://api.stripe.com
```

Example Request

```
\Stripe\Stripe::setApiKey("sk_test_BQokikJOvBiI2HlWgH4oLfq2");
```

- OVERVIEW
- Supported Direct Debit Schemes
- Getting started
- Restrictions
- Anatomy
- Backwards compatibility
- Changelog
- Client Libraries
- API USAGE
- Making requests
- Time zones / dates
- Cursor Pagination
- Response codes
- Errors
- CORE ENDPOINTS
- Customers**
- Create a customer
- List customers
- Get a single customer
- Update a customer
- Customer Bank Accounts
- Events

Customers

Customer objects hold the contact details for a customer. A customer can have several [customer bank accounts](#), which in turn can have several Direct Debit [mandates](#).

Note: the `swedish_identity_number` field may only be supplied for Swedish customers, and must be supplied if you intend to set up an Autogiro mandate with the customer.

Properties

<code>id</code>	Unique identifier, beginning with "CU".
<code>address_line1</code>	The first line of the customer's address.
<code>address_line2</code>	The second line of the customer's address.
<code>address_line3</code>	The third line of the customer's address.
<code>city</code>	The city of the customer's address.
<code>company_name</code>	Customer's company name. Required unless a <code>given_name</code> and <code>family_name</code> are provided.

[Home](#)
[News](#)
[Developer Showcase](#)
[My Applications](#)
[Web API](#)
[User Guide](#)
[Beginner's Tutorial](#)
[API Endpoint Reference](#)
[API Console](#)
[Object Model](#)
[Authorization Guide](#)
[Using Scopes](#)
[Code Examples & Libraries](#)
[Playlist Guide](#)
[Track Relinking Guide](#)
[Migration Guide](#)
[Change Log](#)

Web API User Guide

Our Web API provides client applications with fast and reliable access to Spotify data.

Note that by using Spotify developer tools, you accept our [Developer Terms of Use](#).

About the Web API

Through the Spotify Web API your applications can retrieve and manage Spotify content. The base address of the API is <https://api.spotify.com>.

There are several **endpoints** at that address, each with its own unique path. Many endpoints are open and yours do not need any special permissions to access them. To access *private* data through the Web API, such as user profiles and playlists, an application must get the user's permission to access the data. **Authorization** is via the Spotify Accounts service at <https://accounts.spotify.com>.

Note: Our **Metadata API** is now deprecated. All its functionality is found in the new Web API. For now, the Metadata API endpoints are available, but be aware they will be removed in the near future. We recommend you **migrate** your existing applications to the Web API as soon as possible.

Requests

The Spotify Web API is based on **REST** principles: data resources are accessed via standard HTTPS requests in UTF-8

Topics

- [About the Web API](#)
- [Requests](#)
- [Spotify URIs and IDs](#)
- [Rate limiting](#)
- [Responses](#)
- [Timestamps](#)
- [Pagination](#)
- [Conditional requests](#)
- [Response Status Codes](#)
- [Error Details](#)
- [Authentication](#)
- [Integration with Echo Nest](#)



API VERSION

V1

V2

Introduction

Making requests

Errors

Links

OAuth

SDK Support

Accounts

Customers

Documents

Funding sources

Transfers

Events

Webhook subscriptions

Webhooks

OPTIONS

Get some help

Sandbox Environment

Introduction

Welcome to the Dwolla API V2 documentation, with ongoing updates as functionality is released to the API. We plan to implement API V1 functionality in API V2, but in the meantime, the two versions will operate in parallel.

The initial focus of API Version 2 centers around a premium product: [white label](#), and provides different functionality from API Version 1. Over time, we are adding the same functionality currently available in V1 to V2.

Official SDKs for Java, Node.JS, PHP, Ruby, and Python are being actively developed.

Please note: white label is a premium product that cannot be activated in our production environment until you've received our approval to use it and have entered into an agreement with us. Please [contact a sales representative](#) to find a package that best meets your needs.

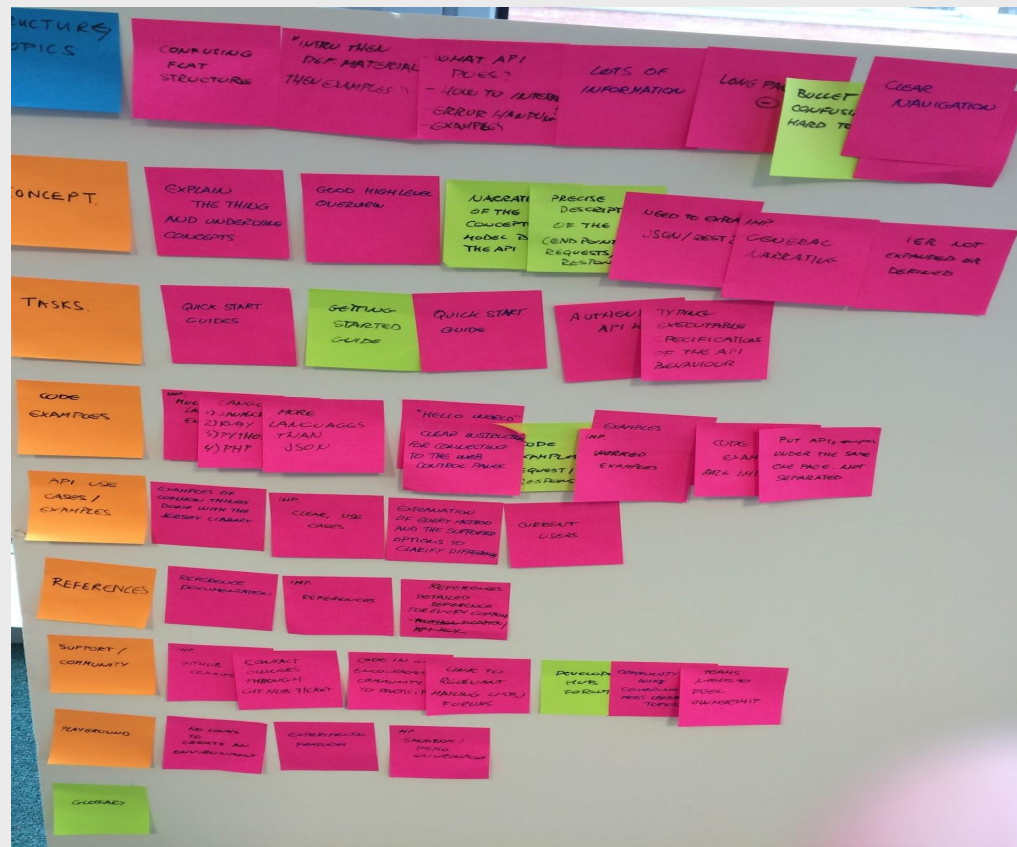
Making requests

All requests should supply the `Accept: application/vnd.dwolla.v1.hal+json` header. `POST` requests must have a JSON encoded body and the `Content-Type: application/vnd.dwolla.v1.hal+json` header.

Requests must be made over HTTPS. Any non-secure requests are met with a redirect (HTTP 302) to the HTTPS equivalent URI.

```
POST https://api.dwolla.com/customers
Content-Type: application/json
Accept: application/vnd.dwolla.v1.hal+json
```

We then analysed all our research....



Our findings.....

Good docs need to
communicate concepts AND
help you get stuff done

Devs and Tech Archs differ on
the info they want from API
docs

Devs consume content
differently to citizens but they
still want friendly content

Devs and Tech Archs worry
government documentation is
out of date

Devs want good signposting:
“ I don't like the
documentation having both
an 'introduction' and
'overview' - that's very
confusing,” Developer.

“Generally interacting with http isn't rocket science but examples are nice to have,”
Developer.

A good design is essential,
eg auto loading features, 3
columns, index that's always
visible, no flat content

We found out much more...

3. Putting our user research into practice



Travis CI API

Overview

Making Requests

External APIs

Authentication

Entities

Other Endpoints

API Clients

Travis CI for Open Source

Travis CI for Private Projects

© 2015 Travis CI GmbH

Riggaerstr. 8,

10247 Berlin, Germany

Overview

Welcome to the Travis CI API documentation. This is the API used by the official Travis CI web interface, so everything the web ui is able to do can also be accomplished via the API.

The first thing you will have to find out is the correct API endpoint to use.

- **Travis CI for open source:** For open source projects tested on travis-ci.org, use <https://api.travis-ci.org>.
- **Travis Pro:** For private projects tested on travis-ci.com, use <https://api.travis-ci.com>.
- **Travis Enterprise:** For projects running on a custom setup, use <https://travis.example.com/api> (where you replace travis.example.com with the domain Travis CI is running on).

Note that both Pro and Enterprise will require almost all API calls to be [authenticated](#).

Making Requests

i If you do not set the **Accept** header, you might retrieve our old API formats. These are deprecated and will be removed soon.

When you write your own Travis CI client, please keep the following in mind:

http shell ruby

```
GET / HTTP/1.1
User-Agent: MyClient/1.0.0
Accept: application/vnd.travis-ci.2+json
Host: api.travis-ci.org
```

```
GET / HTTP/1.1
User-Agent: MyClient/1.0.0
Accept: application/vnd.travis-ci.2+json
Authorization: token "YOUR TRAVIS ACCESS TOKEN"
Host: api.travis-ci.com
```

```
GET /api HTTP/1.1
User-Agent: MyClient/1.0.0
Accept: application/vnd.travis-ci.2+json
Authorization: token "YOUR TRAVIS ACCESS TOKEN"
Host: travis.example.com
```

```
GET / HTTP/1.1
User-Agent: MyClient/1.0.0
Accept: application/vnd.travis-ci.2+json
Host: api.travis-ci.org
```




URL

TOKEN



API REFERENCE

default

Creates a new payment

**Government Digital
Service Payments
Api - Swagger
Documentation** BASE URL:

API VERSION: 1.0.0

Creates a new payment

POST /v1/payments

Creates a new payment for the given account ID, provided the following keys are provided in the request payload: 'amount' and 'return_url'

Parameters

body

Content type:

application/json

CreatePaymentRequest requestPayload

Test this endpoint

TRY

Response Messages

201 Created

400 Bad request.

DOCUMENTATION

This documentation

About GOV.UK Pay

Before you start

Quick start guide to the API

GOV.UK Pay API Overview

Architecture

Testing GOV.UK Pay

Security

Versioning

Support, contact and more information

Contribute

Case studies

Glossary of terms

API

▸ V1

This documentation

This documentation is for developers, technical architects and service managers interested in using the GOV.UK Pay platform to process payments for their online service.

About GOV.UK Pay

GOV.UK Pay is a cross-government platform, currently in beta, that lets departments and agencies take payments easily.

The platform is capable of connecting your government service to a number of different [payment service providers \(PSPs\)](#), although during Beta, the platform will be connected to a single PSP. The providers will support both credit and debit card payments. Over time, further payment methods will be supported, eg direct debit or eWallet.

Your service will only need to integrate with GOV.UK Pay once to let your users make credit and debit card payments. When the platform is expanded in 2016 to accept other payment methods, your service will not have to undertake any new integration. Service teams will find GOV.UK Pay a convenient alternative to supporting multiple PSPs and methods themselves.

4. Next steps?

- Guidance
- Template
- Discovery
- Research on other docs



Cabinet Office

Thanks!

Rosalie Marshall
@rosaliemarshall

