

Public APIs: Extending access to the UK Data Service

John Shepherdson

Associate Director Technical Services
UK Data Service

IASSIST 2015 Minneapolis

4 June 2015

UK Data Service



Structure

- UK Data Service Overview
- What?
- Why?
- How?
- Where?



UK Data Service - Overview

- Led by experts at Universities of Essex, Manchester, Leeds, Southampton, Edinburgh and UCL (University College London)
- Source of guidance, training, and support for data users in UK and around the world
- Currently serves approx. 24,000 registered users
- Funded to coordinate the Administrative Data Research Network, and support the Big Data Network
- Website: ukdataservice.ac.uk



UK Data Service - Overview

- Indexes all data collections in the UK Data Archive
 - all catalogued at thematic level
 - many indexed at variable level
- Also provides access to UK Census data (1971 to 2011)
- Harvests metadata from other sources
- Available for download via Discover search-and-browse catalogue discover.ukdataservice.ac.uk



APIs - What and Why?

- RESTful APIs that are publically available
 - An Application Programming Interface (API) is a mechanism for allowing software components to interact
- Programmatic access to some of our metadata and data
- Innovation Fund - working with 3rd parties
 - <http://ukdataservice.ac.uk/news-and-events/newsitem/?id=3874>
 - App Challenge (CreativeGrid)
 - Leaderboard (AudienceNet)
- Thin end of wedge – much more to come
 - Secure/control own use APIs



RESTful APIs

- Representational State Transfer (REST)
 - “software architecture style consisting of guidelines and best practices for creating scalable web services” [1]
 - developed by W3C Technical Architecture Group
- How to CRUD with REST/HTTP
 - Create - POST method
 - Read - GET method
 - Update - PUT method
 - Delete - DELETE method

• [1]: http://en.wikipedia.org/wiki/Representational_state_transfer



How?

- Approach
- Design
- Implementation



Approach: Open, but not uncontrolled

- Quotas and rate limits - making sure individual and/or collective use of an API does not exceed a pre-defined limit
- Authentication - making sure only registered third-parties can make use of the API
- Authorization - making sure registered third-parties can only use the parts of the API they are entitled to access
- Usage monitoring - providing information on who used what, when and how much



Approach

- How to design an API?
 - Lots of advice, but of course no single way
 - Reviewed several 'big' APIs and tried to distill some common approaches
 - Some mantras 'Once the developer has learned a small portion of the API, they should be able to guess a large portion of it'
 - Implies repeating patterns, consistency
- Produced design guidelines (and tried to stick to them)
 - <https://drive.google.com/file/d/0Bwk0RK5TDo6iVWZLemVpR0dnNUE/view?usp=sharing>
- Ignorance was bliss (but see 'lessons learned')
 - i.e. simple on paper to design for reuse



Design - App Challenge API (subset)

List Datasets

Return list of datasets for specified datasetType.

Format

/<Version>/<Resource>?datasetType=<DatasetType>

Example

/v1/Datasets?datasetType=Open

Dataset Details

Return all metadata for specified dataset.

Format

/<Version>/<Resource>/<DatasetCode>

Example

/v1/Datasets/EQLS

Dataset Topics

Return all topics for specified dataset.

Format

/<Version>/<Resource>/<DatasetCode>/<Scope>

Example

/v1/Datasets/EQLS/Topics

Dataset Keywords

Return all keywords for specified dataset.

Format

/<Version>/<Resource>/<DatasetCode>/<Scope>

Example

/v1/Datasets/EQLS/Keywords



Guidelines - increase API adoption

- Need Developers' portal, which should include:
 - videos
 - client examples in several coding languages
 - API documentation
 - free use (limited hits/functionality)
 - sign-up
 - support (user forum, blog, etc)
 - what does X do? (reference documents – exhaustive)
 - how do I do Y? (tutorials – interesting)
 - what can I do in Z minutes? (quickstart guide - very simple)



Design - components

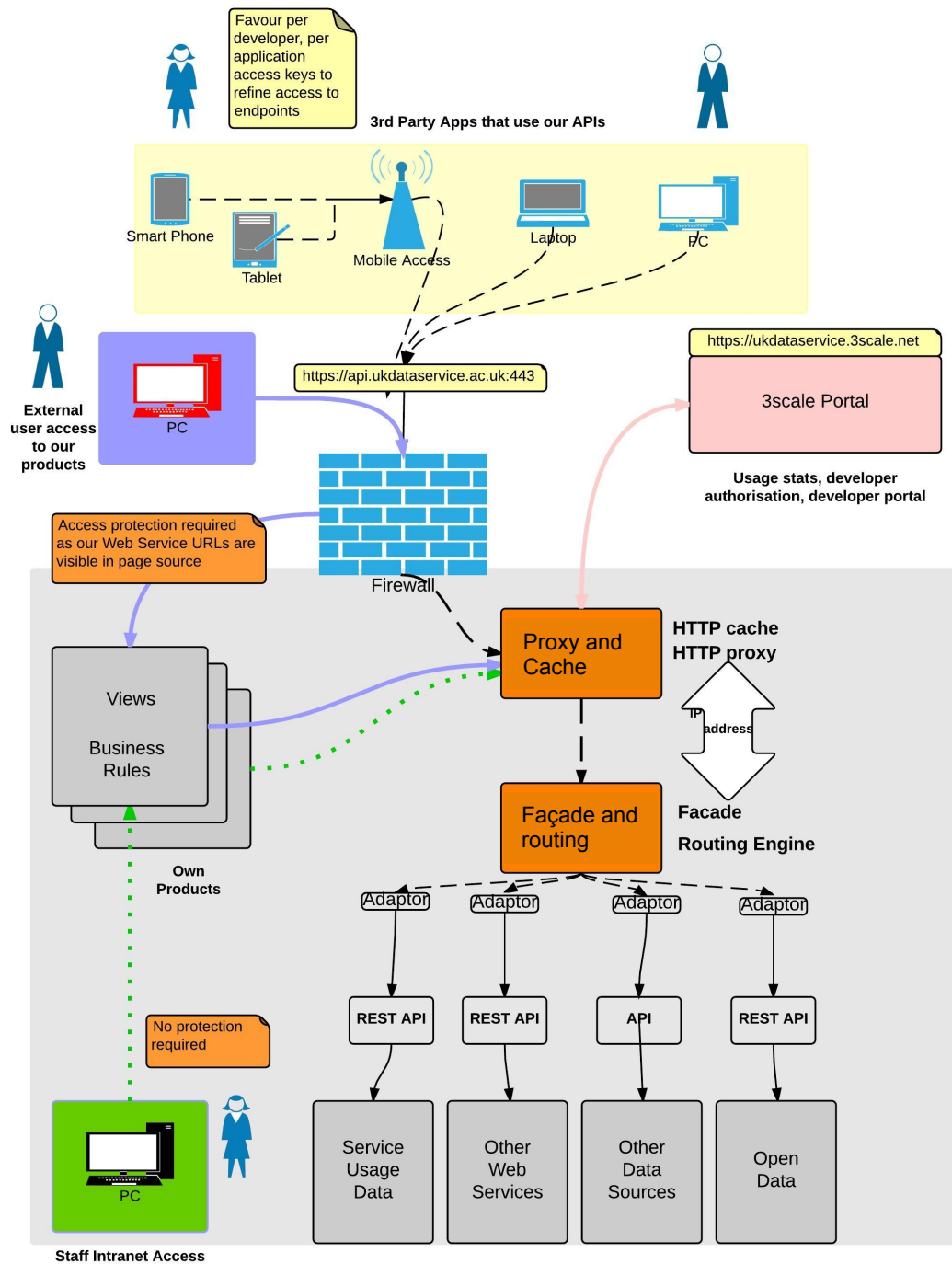
- API Management Service
 - An API management service typically provides protection (authentication and usage control mechanisms), as well as developer account management, technical documentation and forums and reporting services
- HTTP Cache
 - This improves performance by reducing latency by one or more orders of magnitude
- HTTP Proxy
 - Implements the authentication and usage control mechanisms (which in some cases are developed and tested via the API management service)
- Facade
 - A mechanism for implementing a consistent, predictable and easy to use REST API design. It allows us to re-purpose existing web APIs whilst leaving them intact and usable by existing products
- Routing Engine
 - The plumbing that connects the APIs of the backend services to the Facade, allowing replication and locational transparency



Implementation - components

- API Management Service
 - 3scale portal
- HTTP Cache and Proxy
 - Publicly accessible server running OpenResty (based on Nginx HTTP proxy)
- Façade and Routing Engine
 - OWIN/Katana
 - Open Web Interface (OWIN) specification
 - Katana - set of open-source OWIN components developed by Microsoft
 - <http://www.asp.net/aspnet/overview/owin-and-katana>





Where?

- Where can I find the APIs?
- How do I get access?



Implementation – API Documentation

- Used Swagger to annotate the API code
 - Documentation generated automatically
 - imported in to the Developers' Portal
- Swagger is 'a specification and complete framework implementation for describing, producing, consuming, and visualizing RESTful web services.'
- <http://swagger.io>

Swagger – Active API Documents

Documentation

Use our live documentation to learn about the App Challenge API

Operations

App-Challenge Api (VF)

DataSets

/V1/datasets **GET**

Description

DataSets

PARAMETER

VALUE

DESCRIPTION

user_key

(required)

Your API access key

datasetType

Data Set Type (default is 'Open')

Send Request

UK Data Service



Swagger – Active API Documents

Request

```
curl -v -X GET "https://api.ukdataservice.ac.uk/v1/datasets?user_key=[REDACTED]&datasetType=OPEN"
```

Response Body

```
{
  "DataSets": [
    {
      "DatasetId": 2,
      "DatasetCode": "EQLS",
      "DatasetLabel": "European Quality of Life Time Series: Open Access"
    }
  ],
  "Error": {
    "message": "",
    "exceptionMessage": "",
    "exceptionType": ""
  }
}
```

Response Code

200

Response Headers

```
Content-Encoding: gzip
Content-Type: application/json; charset=utf-8
Date: Thu, 21 May 2015 11:31:49 GMT
Server: openresty
Content-Length: 179
Connection: keep-alive
```



Sign up – via 3scale portal

- Apply for account online
- Linked to application plan
 - Limits total no of hits in given period
 - Limits access to various parts of API
- May be subject to authorisation by Administrator
 - Has associated user key
 - All API calls must include valid key

https://api.ukdataservice.ac.uk/v1/Datasets/EQLS?user_key=xxxxxxxxxxxxxxxxxxx
- Usage stats available on per account basis



Learning Points

- Release API design c/w JSON schemas for payloads
 - consumers can prepare for use, prior to implementation
- Return rich error messages
 - support localisation
- What do differently next time?
 - Quite hard to design API structure and JSON payload that is reusable/extensible
 - So design in parallel where possible



Learning Points – general vs. explicit

Too general

List Datasets

Return list of datasets

Format

/<Version>/<Resource>

Example

/v1/Datasets



Learning Points – general vs. explicit

Too general

List Datasets

Return list of datasets

Format

/<Version>/<Resource>

Example

/v1/Datasets

Context is explicit

List Datasets

Return list of datasets for specified
datasetType

Format

/<Version>/<Resource>?
datasetType=<DatasetType>

Example

/v1/Datasets?datasetType=Open



Thanks to

- UKDS Colleagues

Amir Moradi

Deirdre Lungley

Sirisha Kakarla



Questions

- Rich error handling
- App Challenge details
- UKDS Open Data
- More use cases
- Exemplar API designs
- API management platforms



Rich Error Handling

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Error",
  "description": "Detailed error message, can be localised by using the messageIds",
  "type": "object",
  "properties": {
    "message": {
      "type": "string"
    },
    "messageId": {
      "type": "integer"
    },
    "exceptionMessage": {
      "type": "string"
    },
    "exceptionMessageId": {
      "type": "integer"
    },
    "exceptionType": {
      "type": "string"
    },
    "exceptionTypeId": {
      "type": "integer"
    },
    "stackTrace": {
      "type": "string"
    }
  },
  "required": [
    "message",
    "exceptionMessage",
    "exceptionType"
  ],
  "additionalProperties": false
}
```

```
{
  "message": "An error has occurred.",
  "messageId": 2,
  "exceptionMessage": "Object reference not set to an instance of an object.",
  "exceptionMessageId": 41,
  "exceptionType":
    "System.NullReferenceException",
  "exceptionTypeId": 16,
  "stackTrace": "at
    UKDS.AppChallenge.Service.Models.ErrorModel..ctor(Exception exp)"
}
```



UKDS Innovation Fund – App Challenge

- “The App Challenge project will help to identify datasets that can be opened up to the outside world via an Open Data Institute (ODI) compliant API (Application Programming Interface). This API will then be used to crowdsource apps and new services from the App Challenge community around the world.”
- Cash prize for best apps.
- <http://www.appchallenge.net/uk-data-service-open-innovation-project/>



UKDS Open Data

For more details see

<http://ukdataservice.ac.uk/get-data/open-data.aspx>

Also see Margherita Ceraolo's 'Finding space in an Open Data World' presentation from track A3 on Wednesday



Other Use Cases for Open APIs

- CESSDA-ELSST
 - ensure only registered users that have accepted the license conditions can access a machine-readable version of the thesaurus;
 - ensure usage is monitored so it can be analyzed and reported on a regular basis.
- Producer accounts
 - allow producers to harvest statistics relating to the usage of their data collections.
- Catalogue record metadata harvesting
 - restrict number of simultaneous requests to preserve the integrity of the service.



Exemplar APIs

- transportAPI
- Guardian Content Platform
- BBC Developer Portal
- Apigee's API Facade Pattern eBook, (or their [API Best Practices Blog](#) for bite-size chunks)
 - explains the motivation and approach to designing from the outside in
 - (start with the User Experience, then handle the plumbing in order to connect to existing systems).



Some API Management platforms

- Mashery (<http://www.mashery.com/>)
- Apigee (<http://apigee.com>)
- 3scale (<http://www.3scale.net/>)

