

ONS OpenAPI Training Slides

July 2014

Contents

1. Introduction
2. Web Services Basics
3. ONS OpenAPI Overview
4. Discovery
5. Delivery
6. Error Codes
7. Client Applications
8. Examples

1. Introduction

- What is the ONS OpenAPI?
- NeSS and NOMIS
- Release Timetable
- Resources

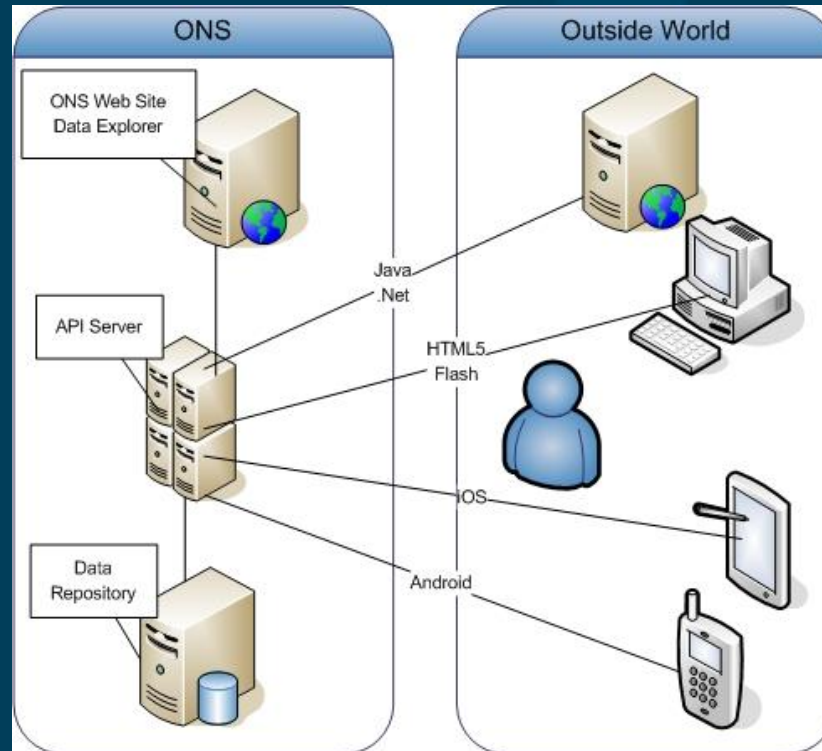
ONS OpenAPI Overview

The ONS OpenAPI allows you to interact with ONS data programmatically from your own application.

Using the API you can:

- Request complete datasets to be downloaded into your own system
- Request specific sets of data items
- Request data to use in real-time, or to store for later
- Discover the data available and geographical areas that are available
- Create something new and exciting!

What is the ONS OpenAPI?



- The ONS OpenAPI is an Application Programming Interface which makes our data available on demand to client applications both internal and external.
- It is implemented as a RESTful web service.

NeSS and NOMIS

- NeSS and NOMIS both have live data web services. Whilst the ONS OpenAPI is intended to be the long-term solution, for the time being we have three complementary services.
- NeSS Data Exchange – older service (established in 2008) giving access to monovariate NeSS data via SOAP or REST. For details go to its [landing page](#)
- NOMIS API - modern RESTful service serving data from NOMIS. For details see the [instructions page](#)

Release Timetable for the ONS API

- Initial Beta release October 2013
- Enhanced Beta release July 2014
- Further updates to follow.
- Mostly Census 2011 data but more being added over time (e.g. Population Projections, Annual Survey of Hours and Earnings)

Resources

- API Service Pages
 - Manual
 - Example clients
 - How To guides
 - Sample output
- Developers' Forum on [Stack Overflow](#) (tag ons-api)
- Data questions on [StatsUserNet](#)

2. Web Services Basics

- What is a Web Service?
- SOAP
- REST
- Security

What is a Web Service?



- The W3C defines a web service as "a software system designed to support interoperable machine-to-machine interaction over a network".
- Most services work using either the SOAP or REST protocols.

SOAP



- SOAP stands for “Simple Object Access Protocol”.
- It uses XML messages transmitted via HTTP or SMTP.
- A SOAP service’s methods are defined in a document called a WSDL (Web Service Definition Language) – pronounced “wizdel”.
- Example method from NeSS Data Exchange: `getAreaChildren('24UE')`. The response is a list of areas in XML.
- SOAP services are traditionally tightly coupled with client applications (via stub generation).

REST

- REpresentational State Transfer (REST) is a style of software architecture for distributed systems such as the World Wide Web. REST has emerged as a predominant web service design model.
- Everything is a resource with a unique URI addressable via an HTTP request.
- REST services are loosely coupled with their client applications.
- The ONS OpenAPI uses REST.



Security

- Some APIs have no security at all – requests are completely anonymous.
- Others require an API Key to identify the user.
- A few have real authentication requiring a username and password to be sent with each request.
- Some APIs encrypt the response using SSL but this slows the service down.
- The ONS OpenAPI (Beta) requires a registration key – mainly for usage monitoring but also useful for debugging.



3. ONS OpenAPI Overview

- Design Principles
- URIs
 - Elements of URI
 - Concepts and References
 - Representations (output formats)
 - XML
 - JSON
 - API Key
 - Context
 - Geography
 - Filter Terms
 - Paging Parameters
- Discovery and Delivery

Design Principles

- The API specification follows recommendations made in the document [Designing URI Sets for the Public Sector](#) which aims to provide a unified approach for all public data services.
- A [new set of guidelines](#) has just been issued by the Cabinet Office, too late for our initial design, but may be useful for later versions. From an initial review the ONS API generally follows these guidelines.
- The ONS API implements [REST](#) using a framework called [Jax-RS](#).
- Linked data ([RDF](#)) output is not directly supported but the design is sympathetic to future use.

URIs

ONS OpenAPI requests all consist of a Uniform Resource Identifier (URI) submitted via an HTTP GET. The URI includes

1. Path `http://data.ons.gov.uk/ons/api/data/dataset/Q5301EW`
2. Representation `.xml`
3. Query String
 - > API Key `apikey=R1UBX9V7`
 - > Context `context=Census`
 - > Geography `geog=2011WARDH`
 - > Slicing `dm/age=42,43,44`
 - > Paging `noobs=100`

Elements of URI

- Domain: <http://data.ons.gov.uk/ons/api>
- Context: `/data/`
- Path To Resource: Points to thing to be retrieved
- Representation: xml, json, csv etc.
- Query: Data Context
- Query: Geographic Hierarchy
- Query: Slice Parameters
- Query: Paging Parameters

Path to Resource

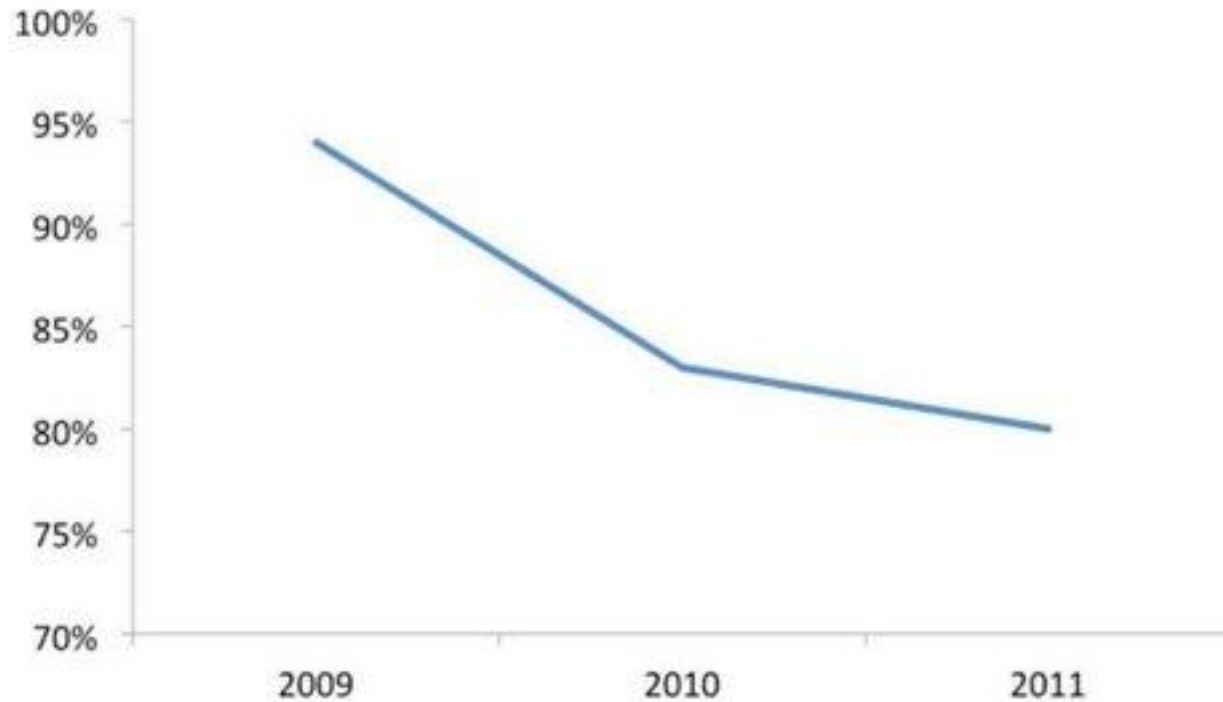
- Usually a concept and a reference in a pair.
- A concept is a category of thing, e.g. dataset.
- A reference is an incidence of a thing, e.g. QS01EW.
- So the path to the resource is dataset/QS01EW.
- The URI Sets paper recommends that omitting the reference is an inferred request for a list. The ONS API uses collection entities instead e.g. datasets

Representation



- For the beta ONS OpenAPI, XML and JSON are supported for direct responses, with CSV and XLS also available for zipped downloads.
- The XML format used for data and dataset structure is [SDMX](#) Generic.
- The JSON format can be the SDMX auto-converted or alternatively the more lightweight [JSON-STAT](#).

XML

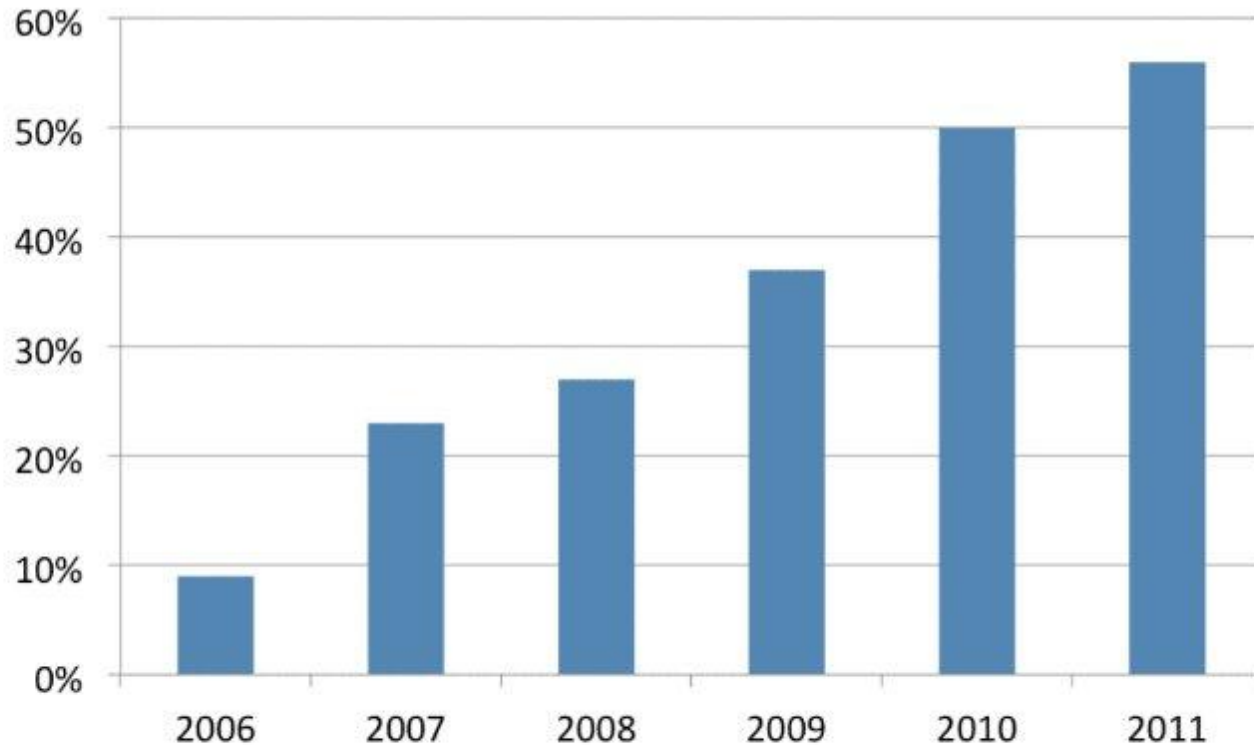


Percentage of APIs with XML support

Based on directory of 3,200 web APIs listed at ProgrammableWeb, May 2011

XML losing popularity but still a “must”

JSON



Percentage of new APIs with JSON support

Based on directory of 3,200 web APIs listed at ProgrammableWeb, May 2011

JSON on the rise – also a “must”

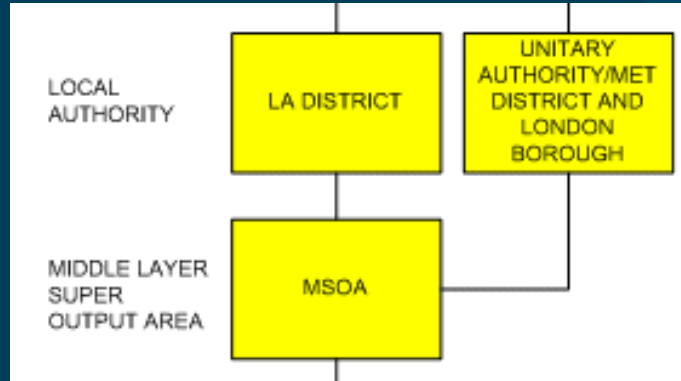
Data Context

- The datastore underneath the API is divided into four sections, each one of these is selectable via the query string `?context=<section name>`
- The four contexts are Census, Socio-Economic, Economic and Social.
- It is possible for items with the same name to exist in more than one context.
- In the Enhanced Beta version of the API, most data is under the context of Census, plus a few Economic ones.

Filter Terms

- For datasets, you can specify a slice using dimension item values. `?dm/area=E06000047, E05008002 &dm/sex=M` (only give me data for males in Durham and Sedgefield).
- For geographic hierarchies, you can slice them using level numbers and / or parent codes. For example `?geog=2011WARDH&parent=E12000004&level=3`

Geographic Hierarchies



- The ONS OpenAPI uses an entity called a collection.
- Each dataset in the system belongs to a collection.
- Each dataset in a collection differs only by the geographic hierarchy the data is for.
- You can discover which hierarchies are available for a collection
- Then specify the required hierarchy using `?geog=<hierarchy name>`

Paging Parameters

- Most applications, such as data explorers will want the data served to them in chunks – they will not want to parse huge responses.
- The API allows you to specify the number of observations in each chunk e.g. `?noobs=1000`.
- You can then page from chunk to chunk by changing the start observation number e.g. `?startobs=1001`

Discovery and Delivery



- All resources within the API can be discovered. There are two types of discovery:
 1. **Lists** – e.g. give me a list of dimensions within this dataset;
 2. **Nodes** – e.g. describe this node (entity), including the URIs of related nodes (such as parents and children).



- Deliveries are usable things such as data cells, classifications, metadata, binaries etc.

4. Discovery

- Self-discovery (HATEOAS)
- Entities
- Lists
- Geographic Queries
- New Features
- Exercise

HATEOAS

- HATEOAS, is an abbreviation for Hypermedia As The Engine Of Application State. The principle is that the client program dynamically discovers the URIs for all possible state transitions (where do I go next?)
- This is sometimes called self-discovery or programmatic discovery.
- Developer discovery (read the manual) is the most common approach but being able to discover related nodes programmatically is still a useful feature.
- HATEOAS is partially implemented in the Beta API. Many discovery responses contain `linkedNodes` which allow navigation of the API's entity model (see next slide).

Entities

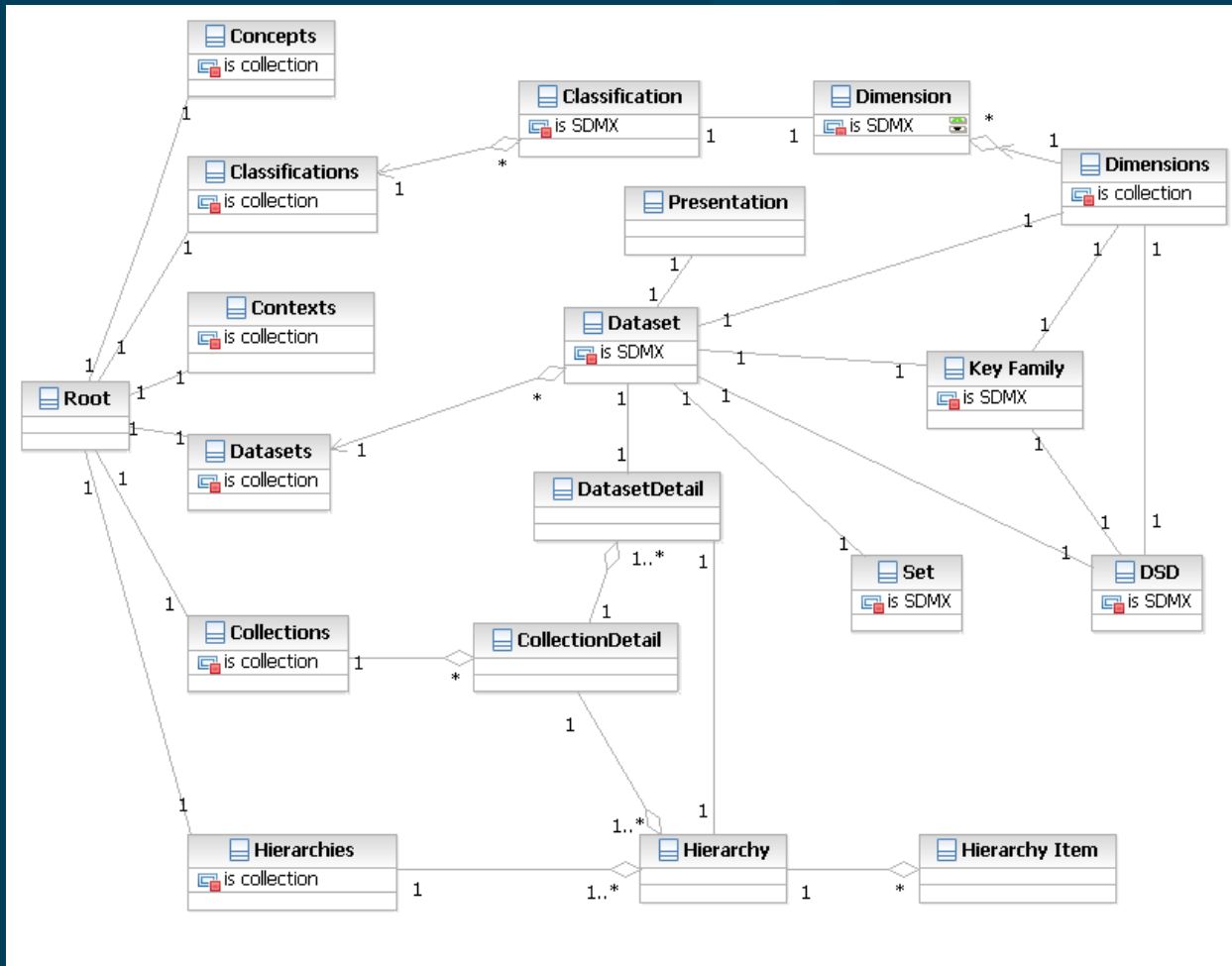


Diagram showing the entities in the ONS OpenAPI

Lists

- Some of the entities in the diagram are collection nodes (ending in s).
- For example the url classifications gives you a list of classifications
- There is also a collection of collections called collections
- The ONS discovery XSD describes the structure of all the possible list items

Geographic Queries



- Many client user journeys involve clicking on maps and selecting areas. The Beta API does not have advanced geographic querying functions, but sometimes other services can be used (e.g. NeSS Data Exchange does postcode to OA lookups, and ONS has just launched its [Open Geography Service](#)).

New In Enhanced Beta



- Function to detect new and updated datasets by date.
- Area-first queries can now be done (query hierarchies without selecting a dataset) including searching for area names.

Discovery Exercise

1. In a web browser, view the API's root node
2. Use the response to discover how to get a list of contexts.
3. Now get a list of collections for context=Census
4. For one of these collections, view the details
5. Using these details, pick a geography for which a dataset exists
6. View the details of one of the datasets
7. Get a list of dimensions for this dataset

Discovery Exercise - Answers

1. Just the base domain url, plus your key, e.g.
[data/?apikey=12345](#)
2. [contexts.xml](#)
3. [collections.xml?context=Census](#)
4. [collectiondetails/QS105EW.xml?context=Census](#)
5. 2011WARDH is valid
6. [datasetdetails/QS105EW.xml?context=Census&geog=2011WARDH](#)
7. [dataset/QS105EW/dimensions.xml?context=Census&geog=2011WARDH](#)

5. Delivery

- Metadata
 - Details
 - Reference
 - Structural
- Data
 - Slicing
 - Paging
 - Downloads
- Exercise

Metadata - Details

```
- <collectionDetail>
  <id>QS207WA</id>
  - <names>
    <name xml:lang="en">Welsh language skills (detailed)</name>
    <name xml:lang="cy">Sgiliau iaith Gymraeg (manwl)</name>
  </names>
  - <defurl representation="xml">
    <href>collection/QS207WA/def</href>
  </defurl>
  - <url representation="xml">
    <href>collection/QS207WA</href>
  </url>
```

- Details can be requested for collections, datasets and other entities.
- General information about the item, for example number of cells in a dataset.

Metadata – Reference

[illegible]

- Reference metadata is held for collections, datasets classifications, dimensions and other entities.
- The URL is the same as the one which delivers the entity, with the path to resource prefixed by /metadata

Metadata – Structural

```
- <structure:CodeList id="URESINDPUK11_URES_POP" agencyID="ONS" version="1.0" isFinal="true">
  <structure:Name xml:lang="en">Usual resident (Population)</structure:Name>
  <structure:Name xml:lang="cy">C_URESINDPUK11</structure:Name>
- <structure:Code value="1" urn="" parentCode="">
  <structure:Description xml:lang="en">Usual resident</structure:Description>
  <structure:Description xml:lang="cy">Preswlydd arferol</structure:Description>
- <structure:Annotations>
  - <common:Annotation>
    <common:AnnotationType>DisplayOrder</common:AnnotationType>
    <common:AnnotationText xml:lang="en">1</common:AnnotationText>
  </common:Annotation>
```

- This is the structure of a datasets, known in SDMX terms as the DSD (data structure definition)
- The entire DSD can be returned in a single response or parts of it e.g. the “Key Family” can be returned individually.

Data - Slicing

```
- <generic:Series>
  - <generic:SeriesKey>
    <generic:Value concept="Location" value="K04000001" />
    <generic:Value concept="C_145" value="1" />
  </generic:SeriesKey>
  - <generic:Obs>
    <generic:Time>Time</generic:Time>
    <generic:ObsValue value="3917833" />
  </generic:Obs>
</generic:Series>
```

- Each dataset has a number of dimensions, and each of these has dimension items.
- To subset the data, name some or all of the dimensions and the items to include e.g.
?dm/sex=M&dm/shoesize=8,9
- Unlisted dimensions are either returned in full if ?total=false, or just the total item if ?total=true

Data - Paging

```
- <generic:Series>
  - <generic:SeriesKey>
    <generic:Value concept="Location" value="K04000001" />
    <generic:Value concept="C_145" value="1" />
  </generic:SeriesKey>
  - <generic:Obs>
    <generic:Time>Time</generic:Time>
    <generic:ObsValue value="3917833" />
  </generic:Obs>
</generic:Series>
```

- The API has a maximum number of cells internal setting (ca. 50000) for immediate responses.
- The parameter ?noobs can be used to limit the number of observations and ?startobs to move from one “page” to another.

Data - Downloads

```
<errorMessage>Your download is being generated. It may not be available immediately.</errorMessage>
- <documents>
- <document>
  <href>http://www.ons.gov.uk/ons/datasets/slice/xml/XML_QS301EW2011WARDH_150638.zip</href>
</document>
</documents>
</dataPackage>
```

- Full datasets are available as pre-canned downloads from the Data Explorer
- Large slices can be requested as zip file using the /dwn representation. In this case the URI of the dynamically generated zip will be returned.
- The download cell limit is higher than the direct one, but if exceeded the full dataset will be offered.

Data - Other

- The API is bilingual, most textual content is available in English and Welsh
- Some datasets are segmented, each segment containing a different combination of cross-tabulated variables. E.g. Segment 1 = Area x Age x Sex and Segment 2 = Area x Age x Religion
- Presentation information (axis and display order) is held for each dimension in a dataset, but is only a “serving suggestion”.



New In Enhanced Beta



- Output is now sorted by the recommended display order by default.
- JSON-Stat implemented. This format is excellent for lightweight applications

Delivery Exercise

1. Display the details of the dataset you discovered earlier
2. Display the dataset itself (remove the word details)
3. Display the reference metadata for that dataset (add the word metadata between data and dataset)
4. Display the structural metadata (add the word dsd)
5. Display the cells only (replace dsd with set)
6. Choose a subset of items for one of the dimensions and apply this to the cells using the ?dm parameter
7. Limit the number of observations using ?noobs
8. Get a download url for the slice using the .xml/dwn representation

Delivery Exercise - Answers

1. [datasetdetails/QS105EW.xml?context=Census&geog=2011WARDH](#)
2. [dataset/QS105EW.xml?context=Census&geog=2011WARDH](#)
3. [metadata/dataset/QS105EW.xml?context=Census&geog=2011WARDH](#)
4. [dataset/dsd/QS105EW.xml?context=Census&geog=2011WARDH](#)
5. [dataset/QS105EW/set.xml?context=Census&geog=2011WARDH](#)
6. [dataset/QS105EW.set.xml?context=Census&geog=2011WARDH&dm/2011WARDH=E05008002](#)
7. [dataset/QS105EW/set.xml?context=Census&geog=2011WARDH&noobs=100](#)
8. [dataset/QS105EW/dwn.xml?context=Census&geog=2011WARDH&dm/2011WARDH=E05008002](#)

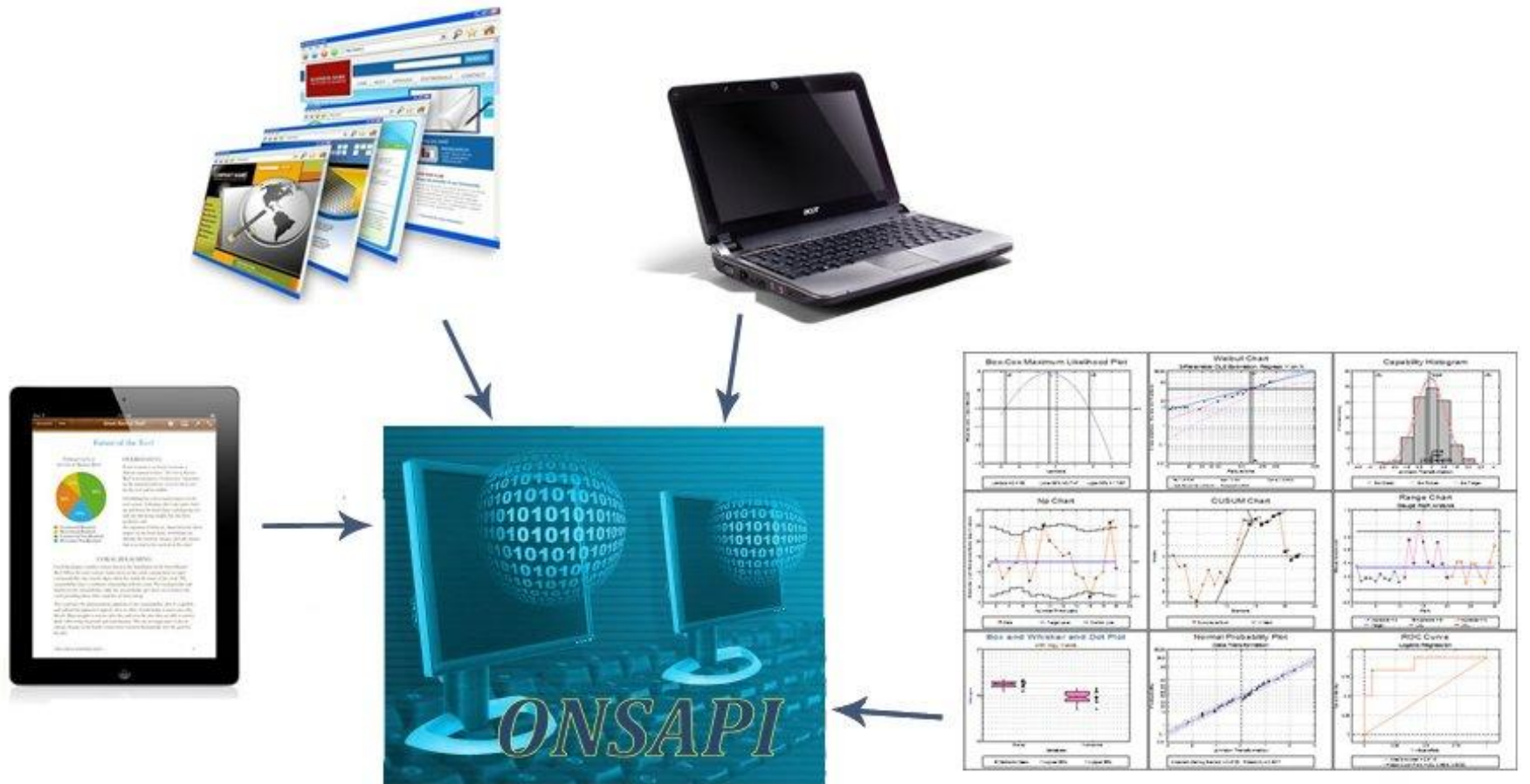
6. Response Codes

- 500 - internal server error (unexpected)
- 407 - proxy authentication required
- 406 - not acceptable (usually an incorrect accept header)
- 404 - not found (could be misspelling or dataset id not found on database)
- 400 – Bad Request (usually a syntax error)
- 202 – Accepted, not complete (download being generated)
- 200 - OK (only returned for successful requests, never used for any type of error condition as this can cause ambiguity for the client application)

7. Client Applications

- Types of Application
 - Hunters and Gatherers
- Programming Languages
- Demo Clients
 - AJAX
 - Android
 - Excel
- Using The API Indirectly
 - Data Explorer
 - Profile Generator

Types of Application



Many types of client applications are possible

Hunters and Gatherers



- A Hunter application slays its data and immediately consumes it.



- A Gatherer application squirrels away the data for later use (possible transformation, and/or input to other data stores).

Programming Languages



PHP



Ruby



Python



.NET



JS



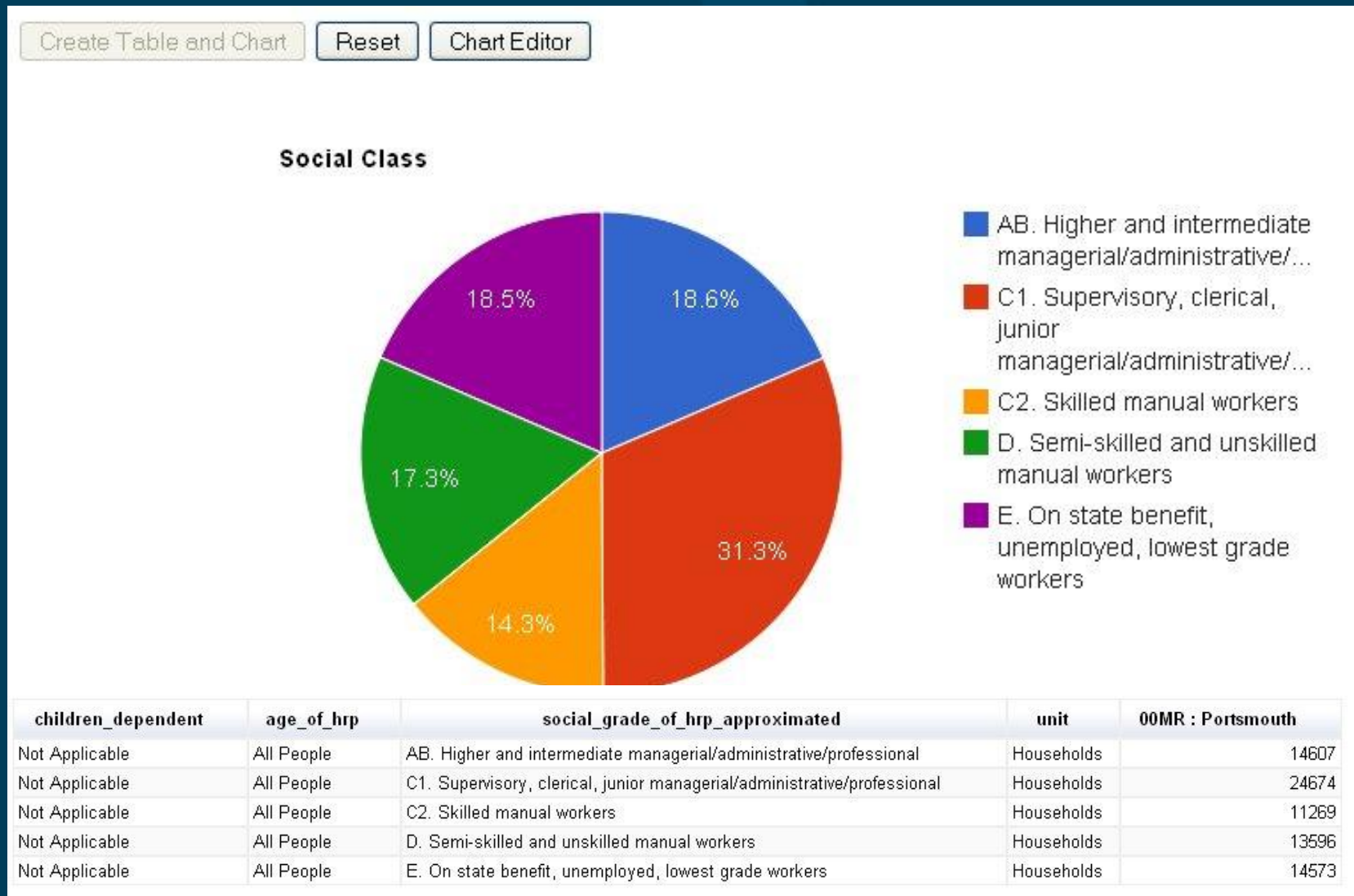
Java



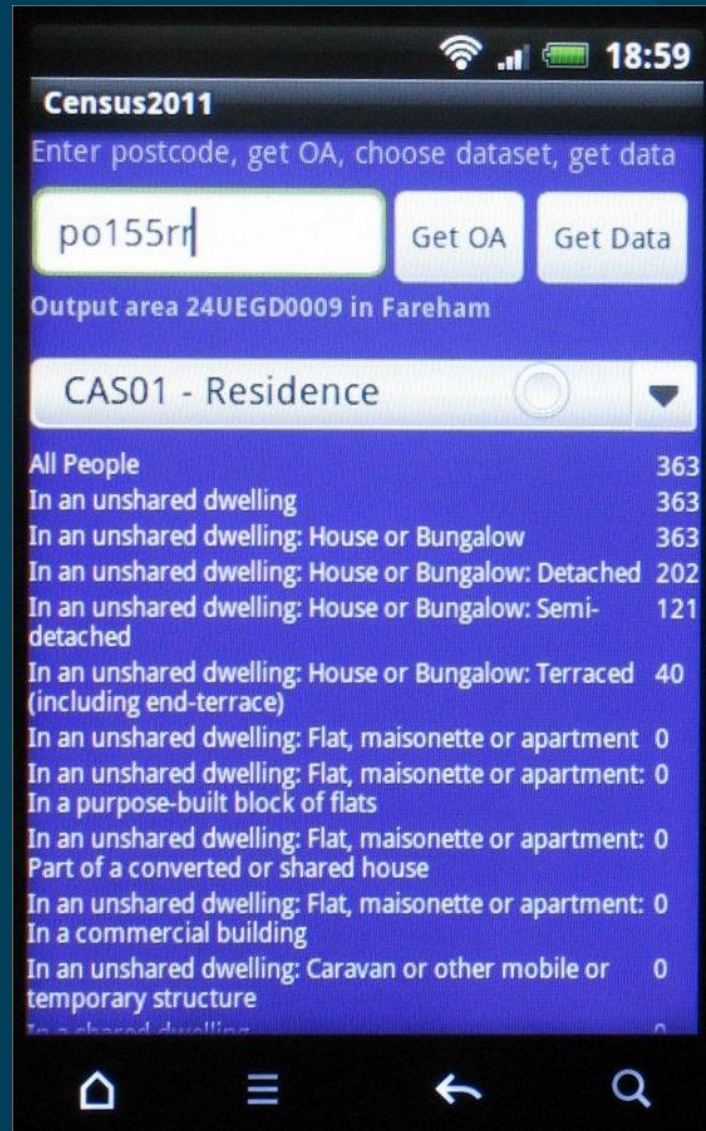
Flash

- The languages above are Programmable web's most popular development platforms.
- There are many others, including statistical tools such as SAS and R.
- The API Service pages currently host examples and demos in AJAX, HTML5, D3, Flash, Java, Excel, Visual Basic and Android. Some screen shots follow...

Clients: AJAX - Google Visualizer



Clients: Android



Clients: Excel

Census 2011 Dashboard

For illustrative purposes only - using WDA Test API

Postcode PO155RR

Data OK

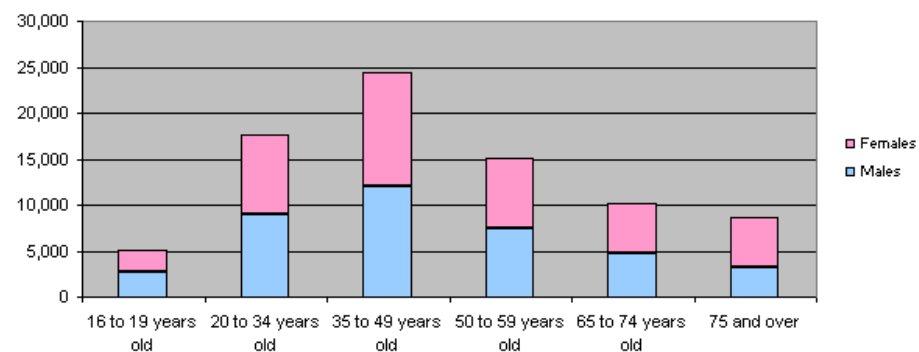
Geographic Level

LA

Fareham

Age by Sex (CAS 01, Persons)

	Male	Female	Total	Percent
16 to 19 years old	2,654	2,397	5,051	4.7%
20 to 34 years old	8,899	8,804	17,703	16.4%
35 to 49 years old	12,026	12,317	24,343	22.5%
50 to 59 years old	7,400	7,761	15,161	14.0%
65 to 74 years old	4,818	5,362	10,180	9.4%
75 and over	3,260	5,359	8,619	8.0%
All Ages	52,882	55,095	107,977	100.0%



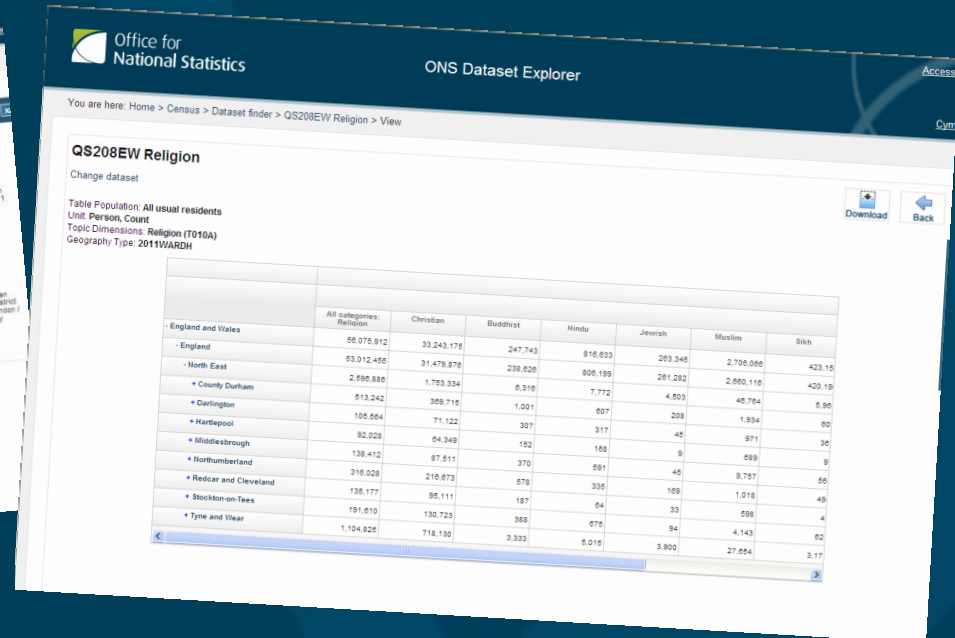
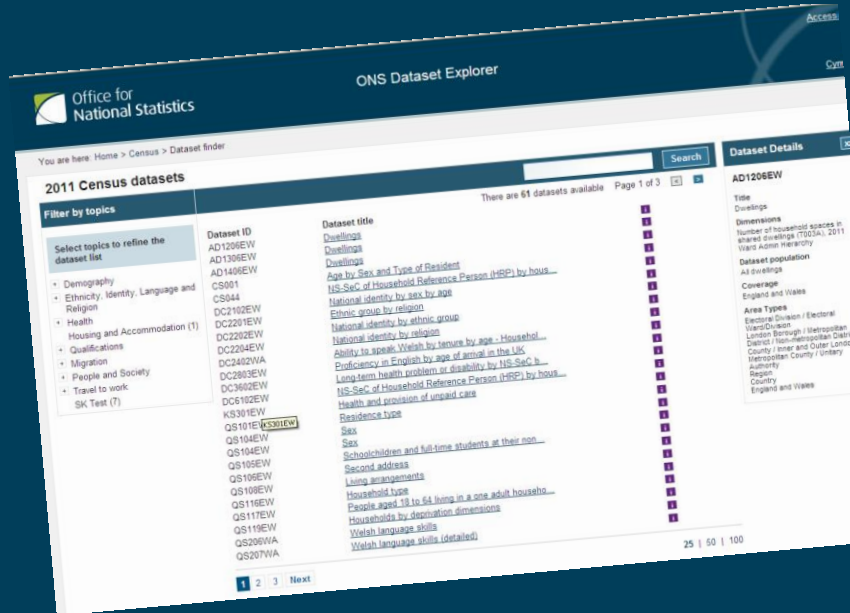
Age by Economic Activity (CAS 43, Persons)

	Economically Active						Economically Inactive					
	Employed: Part Time	Employed: Full Time	Self Employed	Unemployed	Full-Time Student	All Active	Retired	Student	Home or Family Care	Sick or Disabled	Other	All Inactive
16 to 19 years old	237	1,751	39	165	1,696	3,634	0	1,294	39	16	67	1,417
20 to 34 years old	2,038	11,349	1,166	479	354	15,386	9	316	1,396	193	403	2,317
35 to 49 years old	4,541	14,110	2,543	398	74	21,666	47	104	1,621	538	367	2,677
50 to 59 years old	2,629	7,362	1,751	245	10	11,997	927	22	1,131	732	352	3,164
65 to 74 years old	618	1,256	469	57	13	2,413	2,715	3	163	358	113	3,352
All Ages	10,395	35,723	6,226	1,362	2,156	55,862	12,534	1,753	4,446	2,181	1,426	22,340

Using the API Indirectly

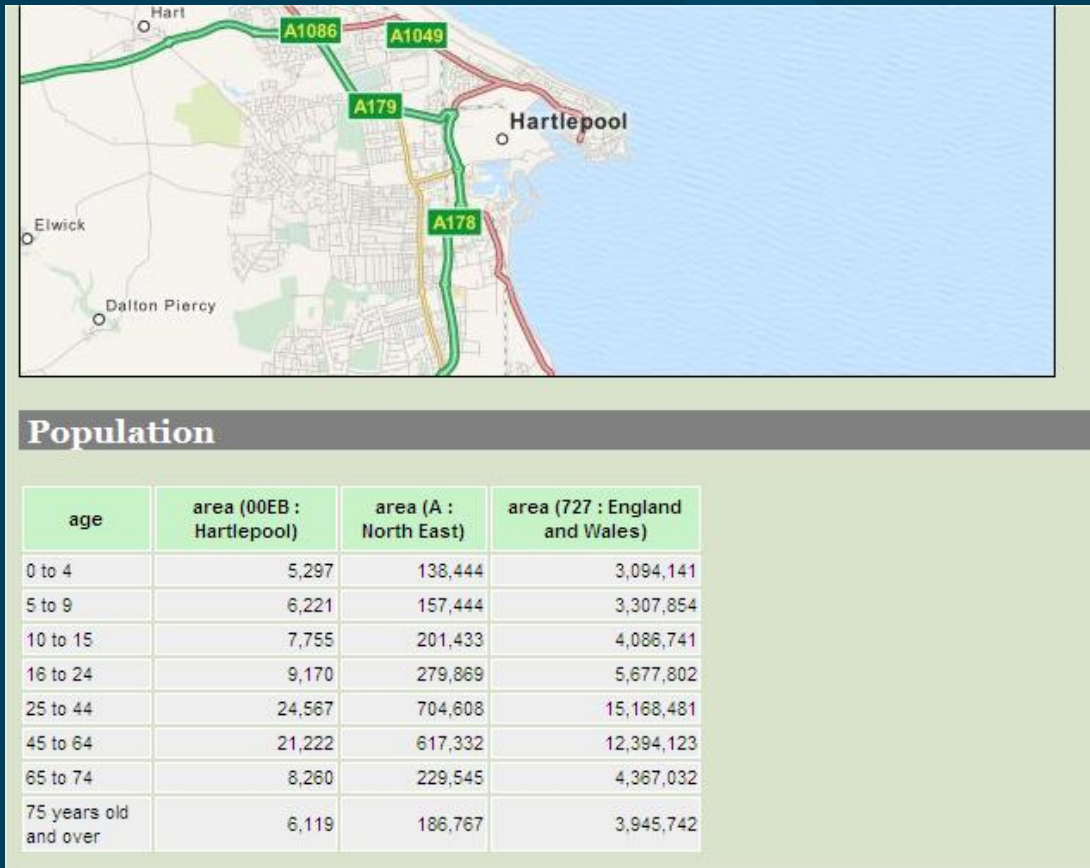
- Don't like programming?
- Use a tool or service that does the API calls for you!
- External Examples: SASPAC, Instant Atlas, LG Inform.
- Internal Example: ONS Data Explorer.
- Internal Example: Profile Generator – a wizard that “writes” a web page for a dynamic area profile.

Data Explorer



- Uses the API to allow end users to select, view, visualise and download data

Profile Generator



- Allows LAs and others to create their own profile HTML page to take away.

8. Examples

- The How To Guides page on the API Service portal takes you through common tasks such as navigating a geographic hierarchy.
- The portal also contains some fully functional example applications.
- Each demo comes with a one-page description page and contains extensive comments in the code.
- They are designed to be simple and concentrate on showing how to use the API with different development tools and platforms.

Web: AJAX



- AJAX stands for Asynchronous Javascript and XML.
- AJAX applications update parts of web pages as required rather than recreating the whole page when a user clicks a button.
- AJAX applications often use a function called XMLHttpRequest to call web services.
- Similar applications can be written using JSON instead of XML, sometimes using JQuery.
- An area picker, a google charts demo, and a mapping demo are supplied, all using JSON.

Desktop: Visual Basic



- Web services can also be called from desktop applications.
- The demo supplied is a Visual Studio 2010 project (compatible with older versions).
- Unlike web applications, desktop clients have full access to the PCs filestore, so this one uses saved XML responses from the API to navigate an area tree part way (live API required to go further)

Other: XSL Transformation



- The Beta API does not have an HTML output for humans (though this is on the wish list).
- One of the examples is an automated XSL runner which will allow you to browse SDMX-ML responses from the API as HTML .
- Unfortunately the XSL to create an an HTML data table from generic SDMX is not a simple. The presence of multiple segments complicates matters but a sample stylesheet is supplied.