

Building APIs in Government for Social Good

Tyrone Grandison PhD

www.tyronegrandison.org @tyrgr

My Time In Government

Deputy Chief Data Officer,
US Department of Commerce (2015-16)

White House Presidential Innovation Fellow,
Department of Labor & US Census Bureau (2014-15)





○ THE PLAN

○ The Reality

○ APIs: Sweat and Toil, CitySDK, MIDAAS

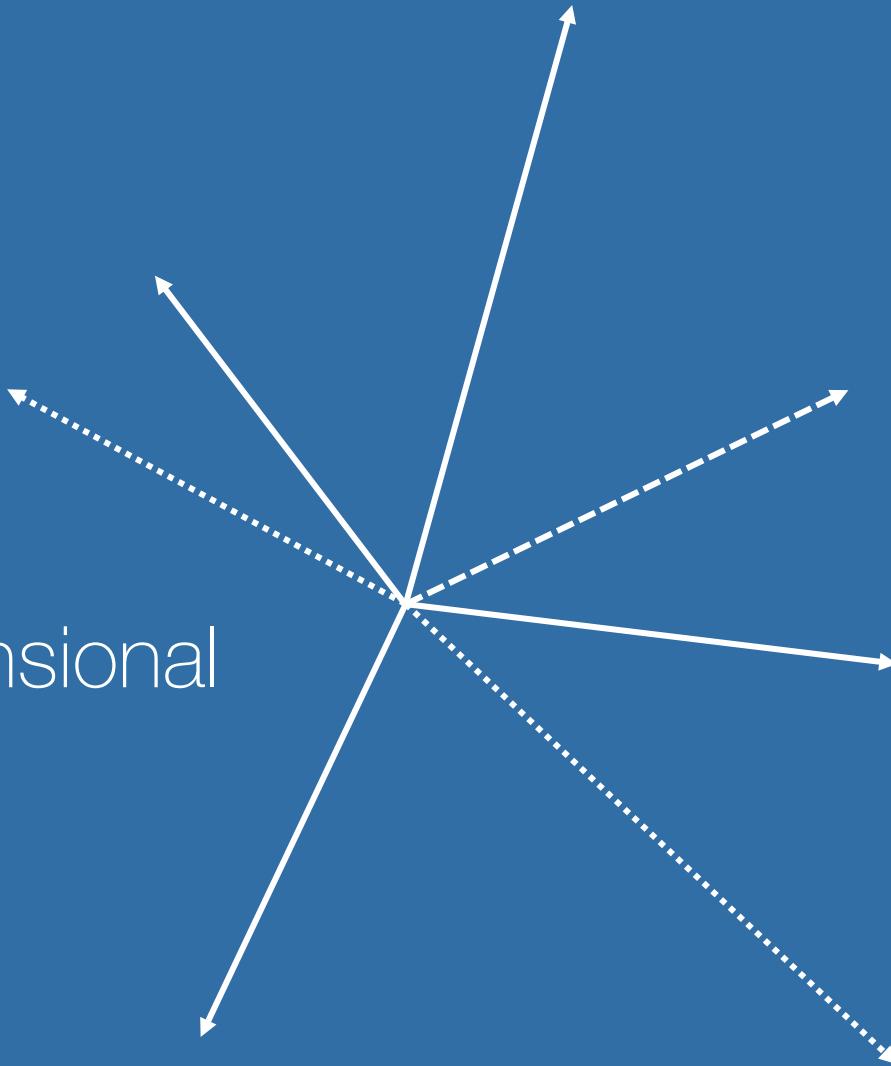
○ Lessons

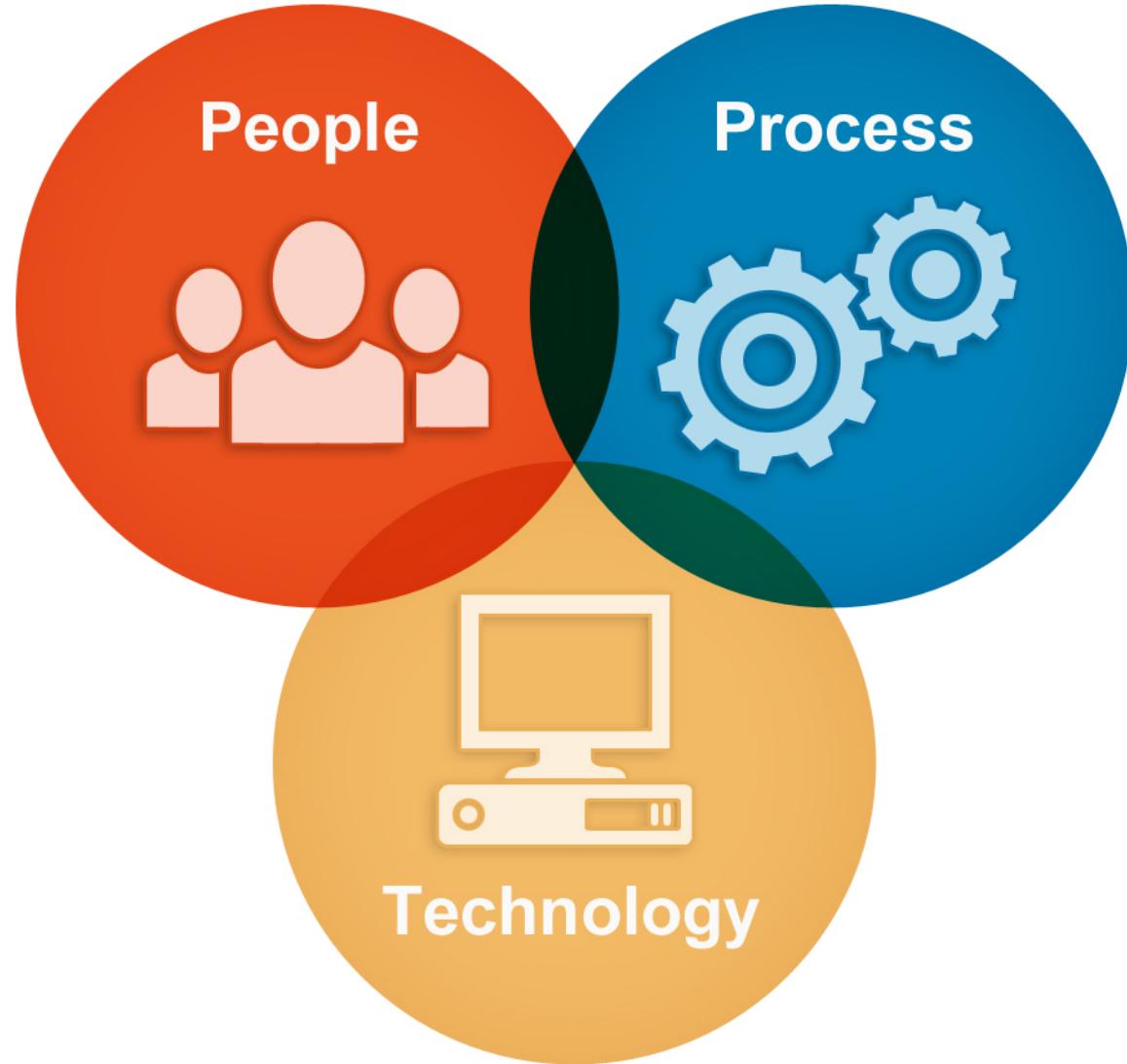
government takes
on the hardest,
inelastic problems

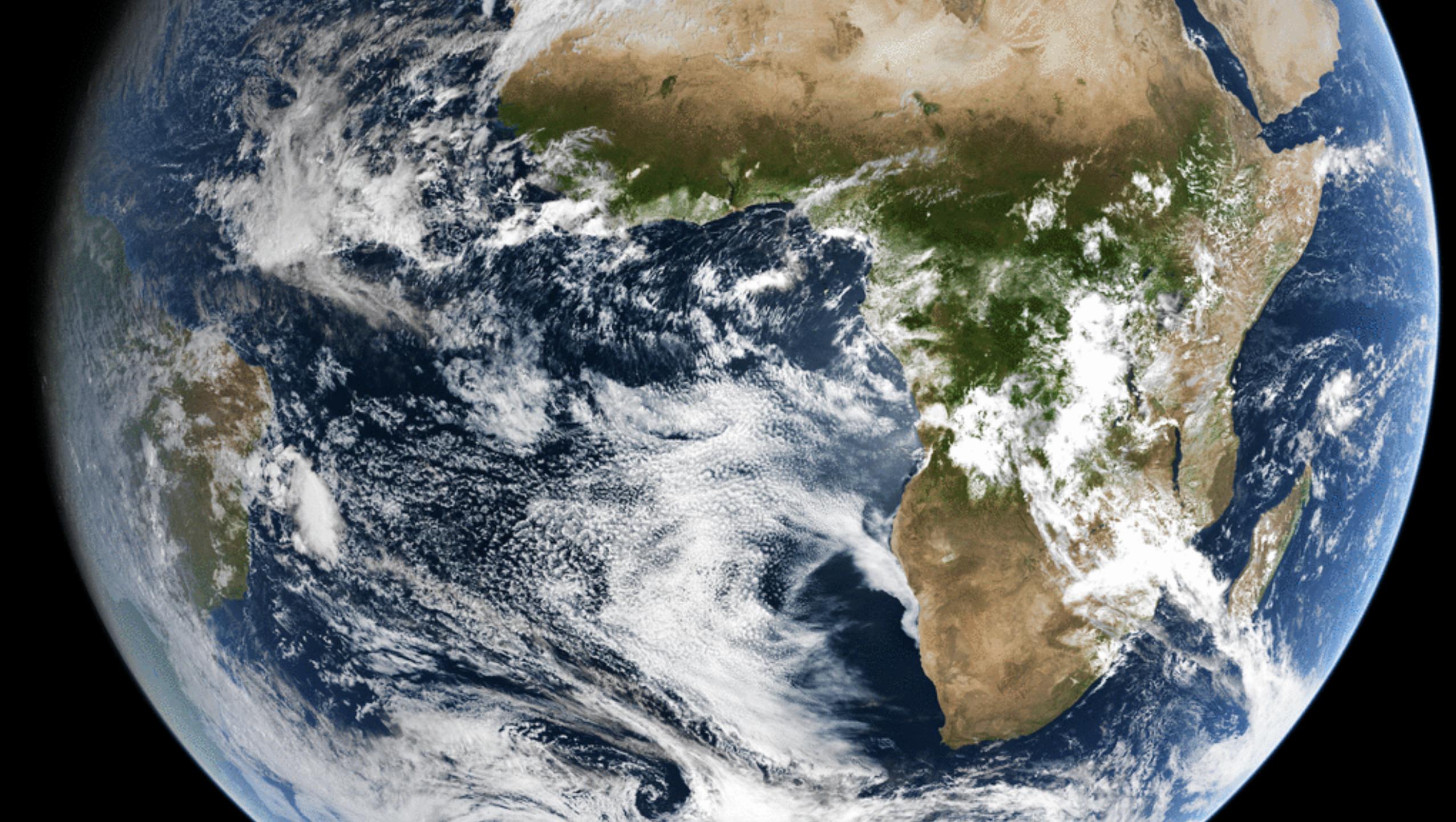
optimum



n-dimensional









BRIEFING ROOM

ISSUES

THE ADMINISTRATION

PARTICIPATE

1600 PENN

Home • Transparency and Open Government

Transparency and Open Government

Memorandum for the Heads of Executive Departments and Agencies

SUBJECT: Transparency and Open Government

My Administration is committed to creating an unprecedented level of openness in Government. We will work together to ensure the public trust and establish a system of transparency, public participation, and collaboration. Openness will strengthen our democracy and promote efficiency and effectiveness in Government.

Government should be transparent. Transparency promotes accountability and provides information for citizens about what their Government is doing. Information maintained by the Federal Government is a national asset. My Administration will take appropriate action, consistent with law and policy, to disclose information rapidly in forms that the public can readily find and use. Executive departments and agencies should harness new technologies to put information about their operations and decisions online and readily available to the public. Executive departments and agencies should also solicit public feedback to identify information of greatest use to the public.

Government should be participatory. Public engagement enhances the Government's effectiveness and improves the quality of its decisions. Knowledge is widely dispersed in society, and public officials benefit from having access to that dispersed

Executive Order M-13-13

 <https://www.data.gov>

 DATA.GOV

DATA TOPICS ▾ IMPACT APPLICATIONS DEVELOPERS CONTACT

The home of the U.S. Government's open data

Here you will find data, tools, and resources to conduct research, develop web and mobile applications, design data visualizations, and [more](#).

GET STARTED
SEARCH OVER [186,304 DATASETS](#)

Federal Student Loan Program Data



BROWSE TOPICS

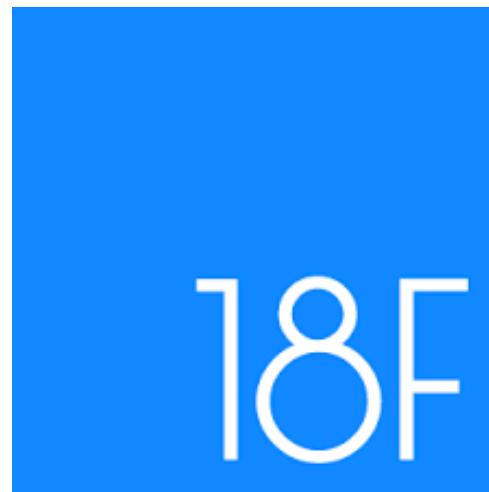
| | | | | | | |
|--|---|--|---|---|--|---|
|  Agriculture |  Business |  Climate |  Consumer |  Ecosystems |  Education |  Energy |
|--|---|--|---|---|--|---|

Federal Source Code Policy

[Discuss](#) | [Edit](#) | [View PDF of Policy](#)[Introduction](#)[1 - Objectives](#)[2 - Scope and Applicability](#)[3 - Three-Step Software Solutions Analysis](#)[4 - Government-Wide Code Reuse](#)[5 - Open Source Software](#)[6 - Exceptions to Government Code Reuse](#)[7 - Implementation](#)[Appendix A - Definitions](#)[Discuss](#)[Edit this page](#)**M-16-21****MEMORANDUM FOR THE HEADS OF DEPARTMENTS AND AGENCIES****FROM:****Tony Scott
United States Chief Information Officer****Anne E. Rung****United States Chief Acquisition Officer****SUBJECT:****Federal Source Code Policy: Achieving Efficiency, Transparency, and Innovation through Reusable and Open Source Software**

The U.S. Government is committed to improving the way Federal agencies buy, build, and deliver information technology (IT) and software solutions to better support cost efficiency, mission effectiveness, and the consumer experience with Government programs. Each year, the Federal Government spends more than \$6 billion on software through more than 42,000 transactions.¹ A significant proportion of software used by the Government is comprised of either preexisting Federal solutions or commercial solutions. These solutions include proprietary, open source, and mixed source² code and often do not require additional custom code development.

When Federal agencies are unable to identify an existing Federal or commercial software solution that satisfies



COMMERCE DATA SERVICE

A photograph of the United States Capitol building in Washington, D.C., showing the dome and the surrounding grounds. The sky is clear and blue.

○ THE PLAN

- The Reality

- APIs: Sweat and Toil, CitySDK, MIDAAS

- Lessons

Sweat and Toil

<http://developer.dol.gov/others/sweat-and-toil>

<https://github.com/USDepartmentofLabor/Child-Labor>

<https://github.com/USDepartmentofLabor/Child-Labor-Android>



Building the API – Step 1



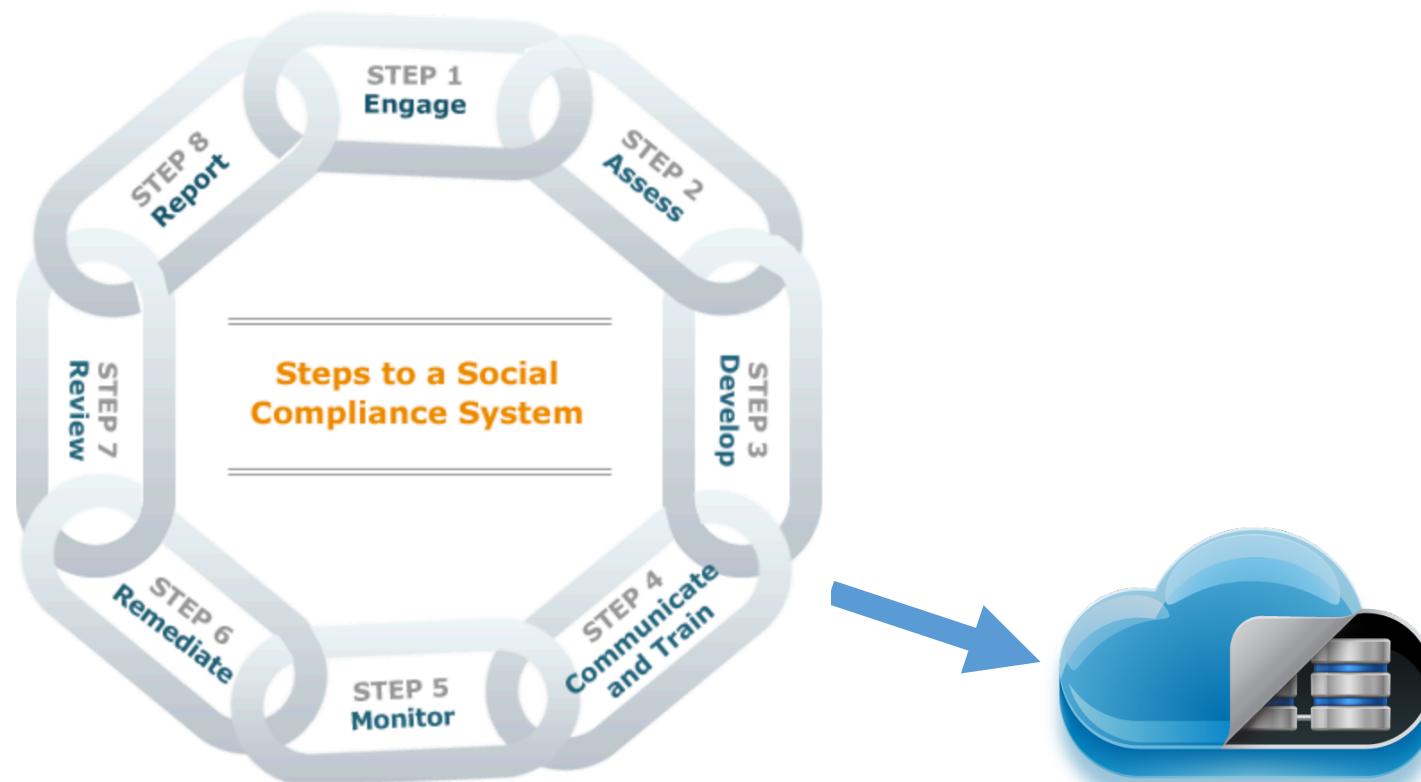
Coalition and Urgency

Building the API – Step 2



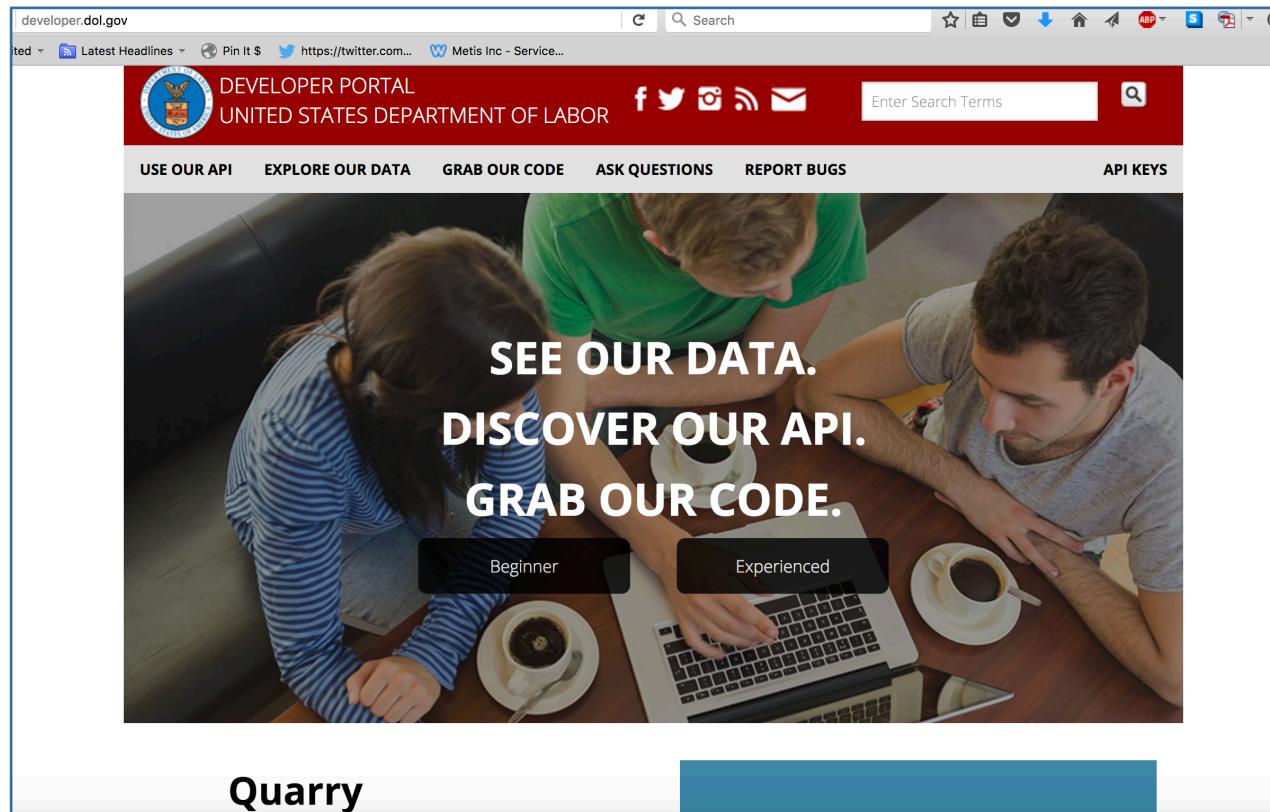
Understand The Data Production Process

Building the API – Step 3



Find Natural Intervention Point

Building the API – Step 4



Learn Technology Constraints

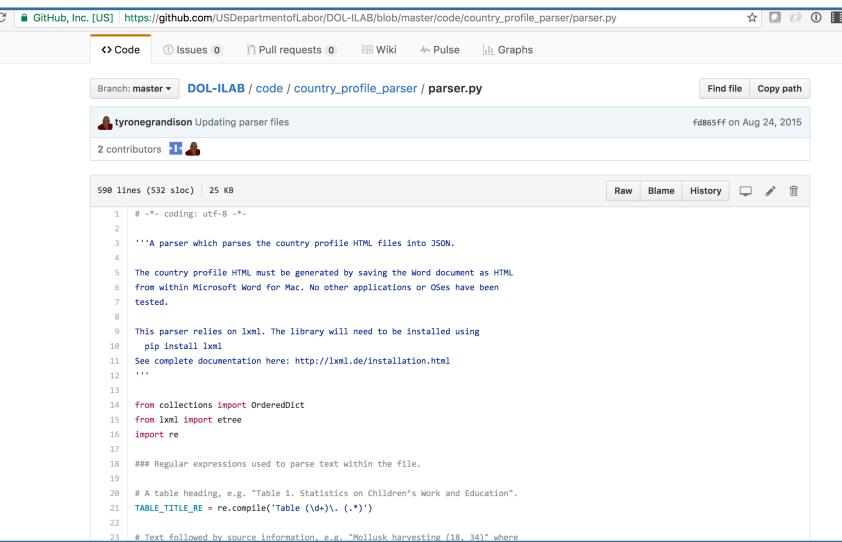
Building the API – Step 5



Understand The API User Groups

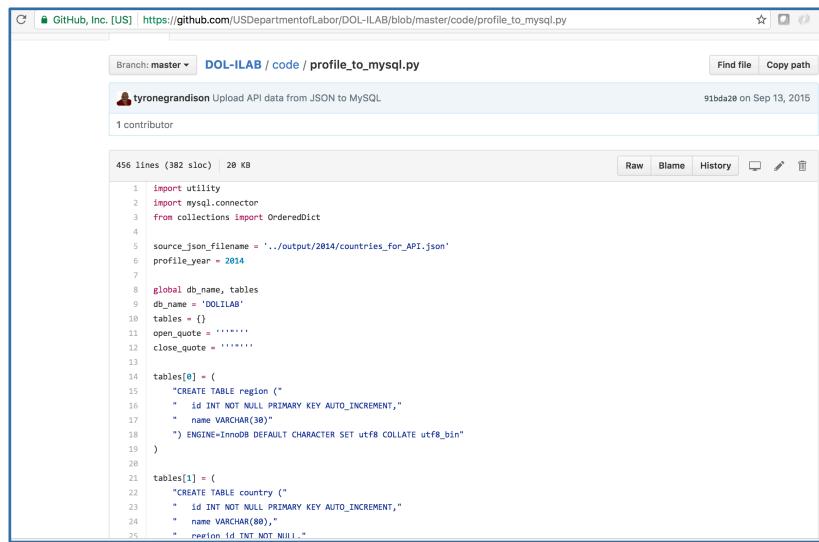
Building the API – Step 6

Python → MySQL, MSSQL → Quarry



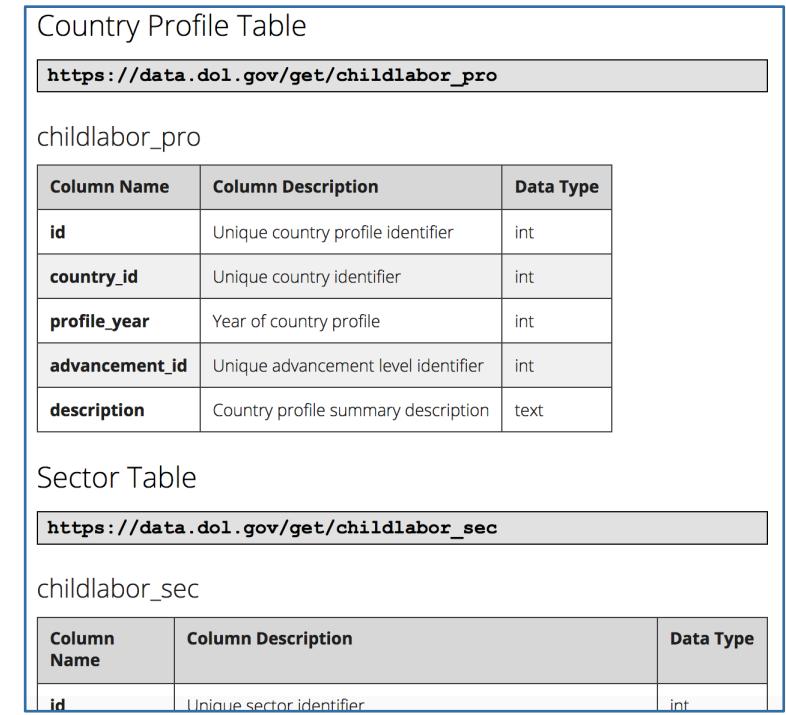
A screenshot of a GitHub repository page for 'DOL-ILAB'. The file 'parser.py' is displayed. The code is a Python script that parses country profile HTML files into JSON. It uses the 'xml' library to handle XML files and 're' for regular expressions. The script saves the Word document as HTML and then processes it. It includes comments explaining the steps and dependencies.

```
598 lines (532 sloc) | 25 KB
# -*- coding: utf-8 -*-
...
'''A parser which parses the country profile HTML files into JSON.
...
The country profile HTML must be generated by saving the Word document as HTML
from within Microsoft Word for Mac. No other applications or OSes have been
tested.
...
This parser relies on lxml. The library will need to be installed using
pip install lxml
See complete documentation here: http://lxml.de/installation.html
...
...
from collections import OrderedDict
from lxml import etree
import re
...
## Regular expressions used to parse text within the file.
...
# A table heading, e.g. "Table 1. Statistics on Children's Work and Education".
TABLE_TITLE_RE = re.compile('Table (\d+)\. (.*)')
...
# Text followed by source information, e.g. "Mollusk harvesting (18, 34)" where
```



A screenshot of a GitHub repository page for 'DOL-ILAB'. The file 'profile_to_mysql.py' is displayed. This script uploads API data from JSON to MySQL. It defines global variables for the database name ('DOLILAB'), tables ('region' and 'country'), and their respective column definitions. It also specifies the character set and collation for the tables.

```
456 lines (382 sloc) | 20 KB
import utility
import mysql.connector
from collections import OrderedDict
...
source_json_filename = '../output/2014/countries_for_API.json'
profile_year = 2014
...
global db_name, tables
db_name = 'DOLILAB'
tables = {}
open_quote = '*****'
close_quote = '*****'
...
tables[0] = (
    "CREATE TABLE region (" +
    "  id INT NOT NULL PRIMARY KEY AUTO_INCREMENT," +
    "  name VARCHAR(90)" +
    ") ENGINE=InnoDB DEFAULT CHARACTER SET utf8 COLLATE utf8_bin"
)
...
tables[1] = (
    "CREATE TABLE country (" +
    "  id INT NOT NULL PRIMARY KEY AUTO_INCREMENT," +
    "  name VARCHAR(90)," +
    "  region_id INT NOT NULL"
)
```



The image shows the Quarry API documentation for the 'childlabor_pro' endpoint. It includes a table for the 'Country Profile Table' and another for the 'Sector Table'.

Country Profile Table

https://data.dol.gov/get/childlabor_pro

| Column Name | Column Description | Data Type |
|-----------------------|-------------------------------------|-----------|
| id | Unique country profile identifier | int |
| country_id | Unique country identifier | int |
| profile_year | Year of country profile | int |
| advancement_id | Unique advancement level identifier | int |
| description | Country profile summary description | text |

Sector Table

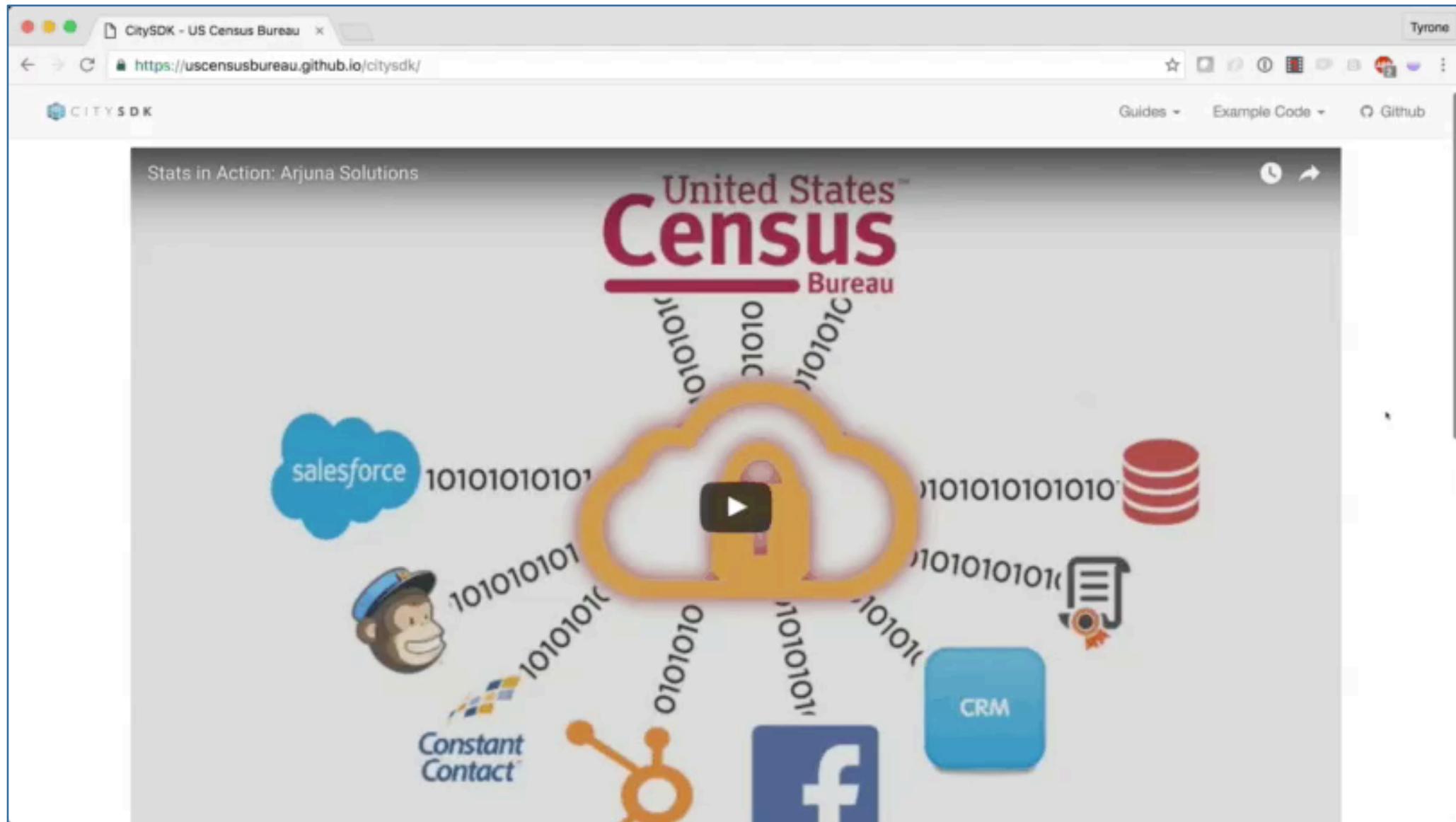
https://data.dol.gov/get/childlabor_sec

| Column Name | Column Description | Data Type |
|-------------|--------------------------|-----------|
| id | Unique sector identifier | int |

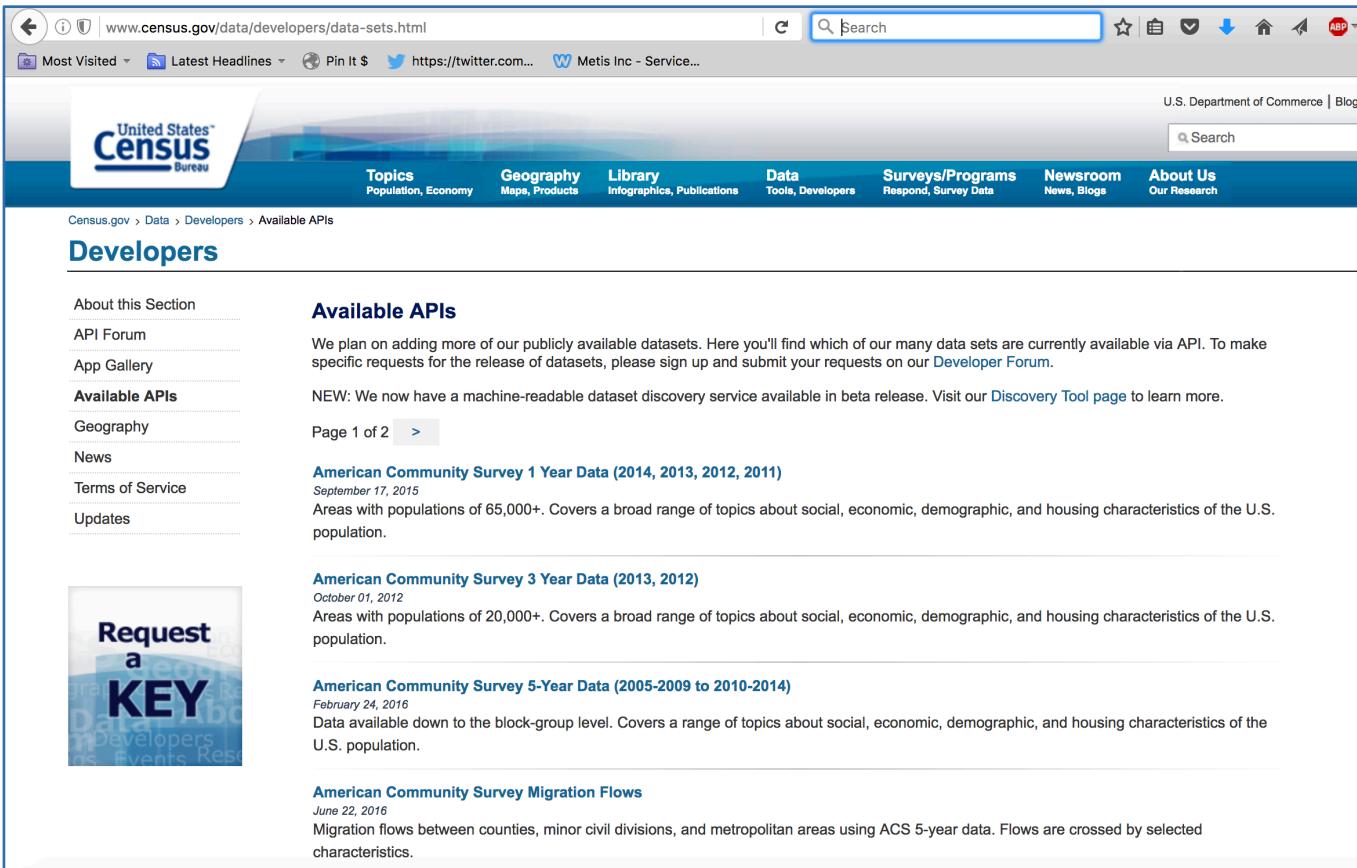
Census CitySDK

<https://uscensusbureau.github.io/citysdk/>

<https://github.com/uscensusbureau/citysdk>



Building the API – Step 1



The screenshot shows a web browser displaying the Census.gov Developers page for Available APIs. The page has a blue header with the Census Bureau logo and navigation links like Topics, Geography, Library, Data, Surveys/Programs, Newsroom, and About Us. The main content area is titled "Available APIs" and includes a sidebar with links for About this Section, API Forum, App Gallery, Available APIs (which is selected), Geography, News, Terms of Service, and Updates. Below the sidebar is a button labeled "Request a KEY". The main content lists four datasets:

- American Community Survey 1 Year Data (2014, 2013, 2012, 2011)**
September 17, 2015
Areas with populations of 65,000+. Covers a broad range of topics about social, economic, demographic, and housing characteristics of the U.S. population.
- American Community Survey 3 Year Data (2013, 2012)**
October 01, 2012
Areas with populations of 20,000+. Covers a broad range of topics about social, economic, demographic, and housing characteristics of the U.S. population.
- American Community Survey 5-Year Data (2005-2009 to 2010-2014)**
February 24, 2016
Data available down to the block-group level. Covers a range of topics about social, economic, demographic, and housing characteristics of the U.S. population.
- American Community Survey Migration Flows**
June 22, 2016
Migration flows between counties, minor civil divisions, and metropolitan areas using ACS 5-year data. Flows are crossed by selected characteristics.

Validate The Demand

Building the API – Step 2

The screenshot shows an Airtable database interface with a dark theme. The title bar reads "Airtable" and "API User Research". The main view displays a table with columns: ID, Interview, Interface, and Quote. The table contains 17 rows of data, each with a timestamp and a brief quote from a user. The left sidebar shows a navigation menu with sections like "Engagements", "People", "Interviews", and "Quotes", with "Interviews" currently selected. A search bar and filter options are at the top of the main table area. On the right side, there are two "Copy base" buttons and a preview window for a mobile device.

| ID | Interview | Interface | Quote |
|----|-----------------|-----------|---|
| 1 | 2/8/2015 17:00 | API | Mostly, I feel that finding the structure of the data is a pain to process. I work with map data all the time, but sorting out the structure of the data is a big pain this |
| 2 | 2/8/2015 17:00 | API | Nokia HERE Maps has CENSUS layers that can facilitate the analyses of census data. |
| 3 | 2/8/2015 17:00 | API | I would have liked (the API response) to fit the construct of common off-the-shelf front-ends for maps. I would have liked (the API response) to fit the construct of |
| 4 | 2/8/2015 17:10 | API | We spent more than 10 hours converting lat / longs from our point data into Census block-groups... |
| 5 | 2/8/2015 17:20 | API | ...Dealing with the Census shapes was the most complicated part... We wanted to compare the data over time, but had to go through a bunch of work just to be |
| 6 | 2/8/2015 17:30 | API | We wanted to filter our query by a (Census) geographic boundary, but the amount of trouble we would have had to go through kept us from doing it that way...We |
| 7 | 2/8/2015 17:10 | API | Across a lot of APIs, you give the url and get parameters with a '?' ... which works if you're doing one specific thing, but I think that's an older paradigm... If you w |
| 8 | 2/8/2015 17:20 | API | I would like to see all the available open data based on a lat/long query (e.g., the Census has x here and crime data is available here) maybe at a certain geograph |
| 9 | 2/8/2015 17:30 | API | HUD had a nice exploration tool that allows you to play with the API, make a call and get a response, which was helpful... however, it would have been nice if (for |
| 10 | 2/8/2015 17:40 | API | First (there is) a 12-digit code that doesn't map to geographic coordinates.. |
| 11 | 2/8/2015 17:40 | API | Then the block-group is not a nice square shape, there are a ton of coordinates. It could be a ring shape... overlapping if your using the outer boundary instead o |
| 12 | 2/8/2015 17:40 | API | The HUD API was well documented, for the Census I would have liked to see all the variables listed and their descriptions like it is for HUD ... list all the variables |
| 13 | 5/19/2015 10:10 | SDK | Move examples to github.io pages |
| 14 | 5/19/2015 10:30 | SDK | Enforce a consistent naming convention |
| 15 | 5/19/2015 10:30 | SDK | More aliases ACS aliases to correct (arcGis) |
| 16 | 5/19/2015 10:10 | SDK | List of features (consider making a list of links with teasers, but each having its own page with examples, tutorial, etc.) .IO Pages |
| 17 | 5/19/2015 10:20 | SDK | Variable search engine (still think the Search functionality would be cool here), maybe pull down the Census Reporter functionality |

Understand The User

Building the API – Step 3

Geography

Geography is at the heart of all Census Bureau data. All of our data sets pertain to very specific geographies that embody very specific geographic concepts. Although we try to make using our API as intuitive as possible, the greater your understanding of these concepts, the more powerful you become as a developer using these data.

Every query must include a geography, and this API supports both FIPS and GNIS geography codes.

To query the API your request URL query string must include a `for` argument which defines the geography level and FIPS code(s).

For example:

`http://api.census.gov/data/2010/sf1?key=...&get=P0010001&for=state:06`

will find the total population for the state of California.

The "for" argument may include additional FIPS codes, separated by a comma.

For example:

`http://api.census.gov/data/2010/sf1?key=...&get=P0010001&for=state:06,24`

will find the total population for the states of California and Maryland.

In some cases you must also include an `in` argument to fully qualify the geography.

Understand Technology Constraints

Building The API – Step 4

2. Client-side Javascript library

For client-side only apps, a Javascript library is provided that is expanded by modules which call on various APIs. Each module essentially adds a new data source for scripting to consume. The core library (`citysdk.js`) includes some basic utility functions that return coordinates for state capitals, etc.

2.1 Add the Library

Add the script libraries to the page. You will need the CitySDK core and then any modules you wish to use. This example will use the Census Module.

```
<script src="https://cdn.rawgit.com/uscensusbureau/citysdk/Release1.1/js/citysdk.js"></script>
<script src="https://cdn.rawgit.com/uscensusbureau/citysdk/Release1.1/js/citysdk.census.js"></script>
```

Note: jQuery is also a required.

2.2 Enable the Library(s)

Make sure you have a Census API key, you may request one [here](#).

```
<script>
    var sdk = new CitySDK(); //Create the CitySDK Instance
    census = sdk.modules.census; //Create an instance of the module
    census.enable("API-KEY"); //Enable the module with the api key
</script>
```

2.3 Create the Request

Define Location

The location can be defined by coordinates (latitude/longitude), state, zipcode, or address. Generally coordinate points are optimal and

MVP



JS

beta



node.js

1. Node API

Once you have acquired an API key, it needs to be included in the [basic auth header](#) of every request.

```
Authorization: Basic <your_api_key>
```

1.1 Example with cURL

```
curl --user yourApiKey: http://citysdk.commerce.gov
```

Note: leave the `password` field empty.

1.2 Endpoints

Base URL: <http://citysdk.commerce.gov>

| Path | Method | Request Data |
|------|--------|--------------|
|------|--------|--------------|

| | | |
|---|------|---------------------|
| / | POST | JSON Request object |
|---|------|---------------------|

| | | |
|--------------------|-----|--|
| /alias-to-variable | GET | Comma separated aliases. Example: ? aliases=income,population |
|--------------------|-----|--|

| | | |
|--------------------|-----|--|
| /variable-to-alias | GET | Comma separated variables. Example: ? variables=P0010001,P0110022 |
|--------------------|-----|--|

Description

Evaluates request and returns either GeoJSON data or GeoJSON with Census data

Returns a map of alias -> variable for the given parameter values

Returns a map of variable -> alias for the given parameter values

1.3 The Request Object

The request object you send using the POST method is just JSON.

| Property | Type | Supported Values | Description |
|----------|------|------------------|-------------|
|----------|------|------------------|-------------|

MIDAAS

<https://midaas.commerce.gov>

<https://github.com/CommerceDataService/midaas-api>

MIDAAS | Making Income Data Accessible

An official website of the United States Government

This site is currently in alpha. Learn More.

MIDAAS Alpha
Commerce Data Service

Topics Explore Developers

Income Across AMERICA

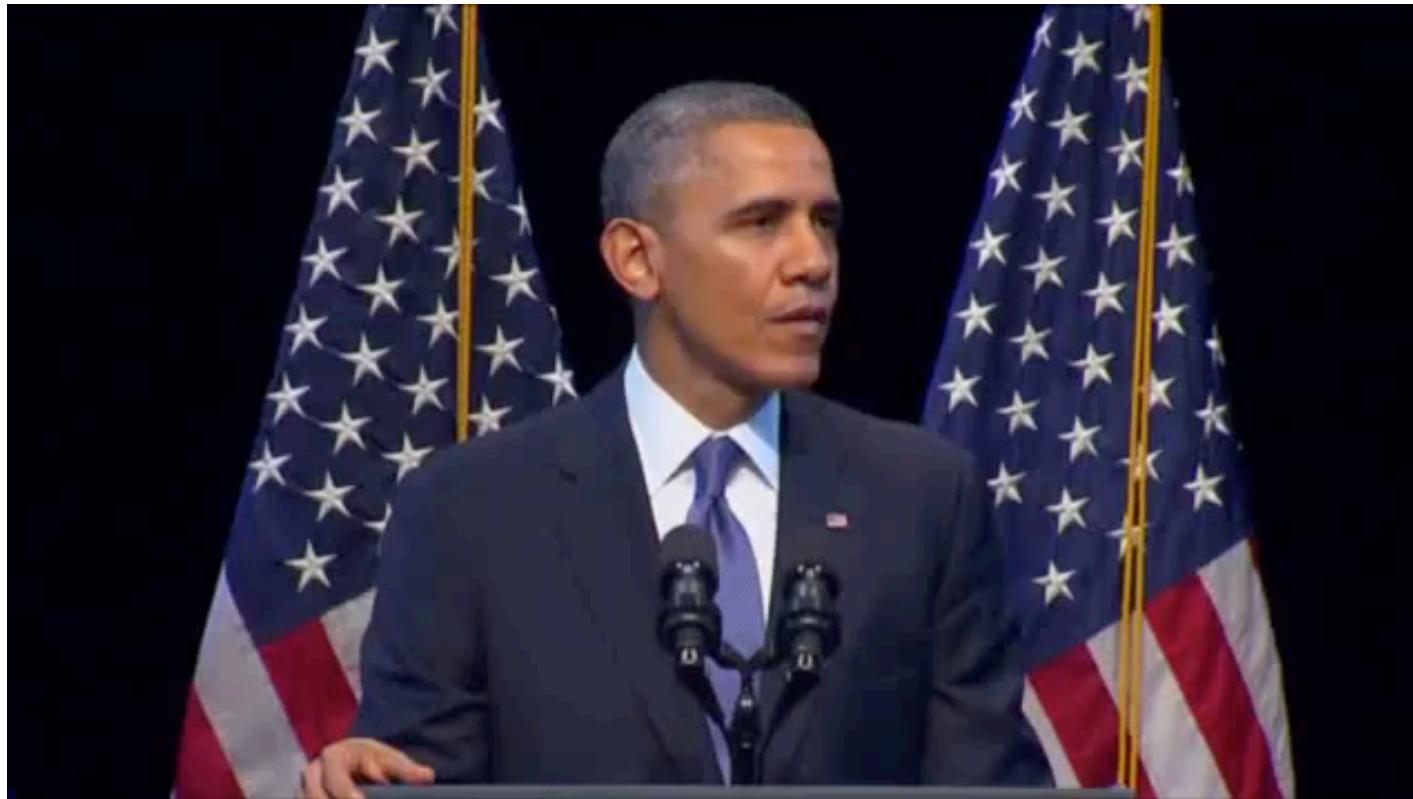
Explore how income changes across geographies and demographics.

Browse Topics or Explore Data

View and Build Powerful Visualizations

Income distribution in the United States is a complex issue that everyone deserves the right to

Building the API – Step 1



Validate The Need

Building the API – Step 2



Kim Stevenson
Intel
Council Chair (2015-2016)



Dan Castro
Center for Data Innovation
Council Chair (2015-2016)



Brian Schimpf
Palantir
Global Director of Engineering



Steve Adler
IBM
Chief Information Strategist



Allen Blue
LinkedIn
Co-Founder + VP



Joy Bonaguro
City of San Francisco
Chief Data Officer



Katy Börner
Indiana University
Professor of Information Sciences



Danah Boyd
Microsoft
Principal Researcher + Privacy Expert



Jack Dangermond
ESRI
CEO



Christopher DiBona
Google
Director of Open Source Engineering



Bill Gail
Global Weather Corporation
CTO



Stan Humphries
Zillow
Chief Economist



Heather Joseph
SPARC
Director



Vadim Kutsyy
PayPal
Head of Data Strategy and Stewardship



Kevin Merritt
Socrata
COO



CJ Moses
Amazon Web Services
General Manager



Jennifer Pahlka
Code for America
Founder



Colin Parris
General Electric
CVP Software Research



Karin Remington
Arjuna Solutions
CTO

Understand Initial User Stories

Building the API – Step 3

The screenshot shows the United States Census Bureau website with a blue header bar. The header includes the Census logo, a search bar, and links to U.S. Department of Commerce, Blogs, Index A-Z, Glossary, and FAQs. Below the header, the main navigation menu has categories: Topics (Population, Economy), Geography (Maps, Products), Library (Infographics, Publications), Data (Tools, Developers), Surveys/Programs (Respond, Survey Data), Newsroom (News, Blogs), and About Us (Our Research). The current page is 'Income' under the 'Topics' category. The page content includes a sidebar with links to 'About this Topic', 'Data', 'Guidance for Data Users', 'Library', and 'News & Updates'. The main content area features a paragraph about income surveys, a 'Read More' link, and three icons: 'News and Updates' (newspaper icon), 'Visualizations' (globe icon), and 'Data Tables' (grid icon). To the right is a large image of various US coins and bills. At the bottom left, there's a 'Latest' section with tabs for 'News' (selected), 'Publications', 'Data', and 'Working Papers'. A news item is shown: 'Income, Poverty and Health Insurance Coverage in the U.S.: 2015' from September 13, 2016. It states: 'Real median household income increased by 5.2 percent between 2014 and 2015 while the official poverty rate decreased 1.2 percentage points.' On the right side, there's a small sidebar with a timestamp 'Sep 13, 2016 22:59 UTC (+7)', a 'United States' button, a 'World' button, a 'U.S. Population' section showing the number '324,473,463', and a 'COMPONENTS OF POPULATION CHANGE' section.

Validate with Domain Experts

Building the API – Step 4



AWS Granted Authority to Operate for Department of Commerce and NOAA

June 14, 2016 | Chad Woolf | Compliance | ATOs | Authority to operate | Department of Commerce | Federal Cloud Computing Strategy | NOAA

AWS already has [a number of federal agencies](#) onboarded to the cloud, including the Department of Energy, The Department of the Interior, and NASA. Today we are pleased to announce the addition of two more ATOs (authority to operate) for the [Department of Commerce](#) (DOC) and the [National Oceanic and Atmospheric Administration](#) (NOAA). Specifically, the DOC will be utilizing AWS for their [Commerce Data Service](#), and NOAA will be leveraging the cloud for their “Big Data Project.” According to NOAA, the goal of the Big Data Project is to “create a sustainable, market-driven ecosystem that lowers the cost barrier to data publication. This project will create a new economic space for growth and job creation while providing the public far greater access to the data created with its tax dollars.”

Understand Technical Constraints

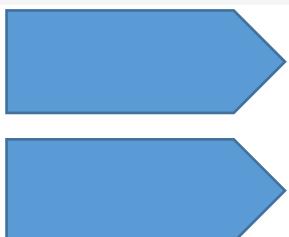
Building the API – Step 5

[GET] /quantiles?[year=?][state=?][race=?][sex=?][ageg [GET] /income/distribution?[year=?][state=?][race=?][sex=?][agegroup=?][compare=?]
returns the income amounts for each quantile

```
curl 'https://api.commerce.gov/midaas/quantiles?state=CA'
{
  overall: {
    5%: 400,
    10%: 4000,
    20%: 10000,
    30%: 16000,
    40%: 23000,
    50%: 30000,
    60%: 40000,
    70%: 52000,
    80%: 71000,
    90%: 100000,
    95%: 143200,
    99%: 455000
  }
}
```

```
curl 'https://api.commerce.gov/midaas/distribution?state=CA&race=white&agegroup=25-34&sex=male&api_key=
{your_api_key}'
{
  '$20.00k-$30.00k': 0.1261146303655346,
  '$120.00k-$130.00k': 0.018000879457209275,
  '$450.00k-$460.00k': 0.005737466859976182,
  '$160.00k-$170.00k': 0.004153510772184425,
  '$180.00k-$190.00k': 0.002701356626896694,
  '$110.00k-$120.00k': 0.013195357305735017,
  '$220.00k-$230.00k': 0.0006076886021645845,
  '$130.00k-$140.00k': 0.011160008724588761,
  '$370.00k-$380.00k': 0.0001702927752706897,
  '$240.00k-$250.00k': 0.0004910497149928792,
  '$380.00k-$390.00k': 0.00006298499907272085,
  '$300.00k-$310.00k': 0.00014229944234948042,
  '$500.00k-$510.00k': 0.0003919066608969297,
  '$540.00k-$550.00k': 0.00009564388748079833,
  '$30.00k-$40.00k': 0.12264112430556122,
  '$10.00k-$20.00k': 0.11912912741282118,
  '$0.00-$10.00k': 0.1194382204638262,
  '$60.00k-$70.00k': 0.07218664088169667,
  '$80.00k-$90.00k': 0.041483786611488695,
  '$70.00k-$80.00k': 0.05156255285199575,
  '$50.00k-$60.00k': 0.0887400317491051,
  '$90.00k-$100.00k': 0.03705034251009218,
  '$320.00k-$330.00k': 0.0015373005329230756,
```

MVP
beta



Redshift, Lambda, API Gateway
Postgres, S3, EC2

A photograph of the United States Capitol building in Washington, D.C., showing the dome and the surrounding grounds. The sky is clear and blue.

○ THE PLAN

- The Reality

- APIs: Sweat and Toil, CitySDK, MIDAAS

- Lessons

Summary

People
Scope
Constraints
Validate
Show The Thing

So Far

Sweat and Toil

- Monthly Data Users > Web Traffic
- Three tools built using this data.

CENSUS CitySDK

- Over 10 civic solutions built using CitySDK
- Positive User Feedback

So Far

Sweat and Toil

- 2016 Department of Labor's Innovation Award.

CENSUS CitySDK

- 2016 Department of Commerce Gold medal
- 2016 Best Data API Award, API:World
- 2015 FedScoop Innovation Of The Year

The background image shows the United States Capitol building, a large white neoclassical structure with a prominent central dome and a surrounding portico of Corinthian columns. In the foreground, there is a well-maintained green lawn with a curved red flower bed. A set of wide stone steps leads up to the main entrance of the building. The sky is clear and blue.

Thank you

tyronewagrandison@gmail.com