Product Requirements for Data Dwarf

Prepared by	***Mercury Squad***
Sean Spearman	seanmspeaman@gmail.com
Cody Brown	codybrwn551@aol.com
Ray Smets	rayjsmets@gmail.com
Aimee Galang	aimeegalang@gmail.com
Tim Shen	timshen1116@gmail.com

Github: https://github.com/rsmets/hgcapstone

Revisions

Version	*Primary Author(s)*	*Description of the Version*	*Date Completed*
1.0	Sean Spearman, Cody Brown, Ray Smets, Aimee Galang, Tim Shen	Initial Version	01/29/2015
1.1	Sean Spearman, Cody Brown, Ray Smets, Aimee Galang, Tim Shen	Test cases added, models and architecture updated, user cases appended	02/26/2015

0: Table of Contents

1: Introduction

1.1 Document Purpose

The purpose of this document is to specify the requirements for the web application *Data Dwarf*. This document is intended to be used as a reference for the development team as they develop the initial prototype of the application.

1.2 Product Scope

Given the amount of structured and unstructured data available today, there is a huge amount of potential knowledge waiting to be analyzed. *Data Dwarf* offers a fast solution to improve the analysis of data by allowing a user to dynamically compare their own data set to a vast set of data stored in our server, removing the need for users to mine, analyze, and visualize arbitrary, but potentially insightful, data. *Data Dwarf* intends to offer a user multiple sets of data that have the strongest correlation to the user's input data. It will additionally allow users to select these data sets that have already been gathered and stored on a server for custom comparison to their input data.

The user will then have the option to choose from among multiple styles of graphs or figures, so that they can find a relevant way of displaying their information. The user will be able to select given correlated data sets to be displayed on their selected graph as well.

1.3 Intended Audience and Document Overview

Data Dwarf is an open source project. However, those who will likely find the most use out of it will be those who are studying or doing research on correlations between different events. For example, if one were studying the change in the amount of ghirardelli chocolate eaten in relation to amount of rainfall in the San Francisco area *Data Dwarf* is there for you. One should use their imagination as any data set supplied *Data Dwarf* can show changes over the same period of time and make a correlation judgement call.

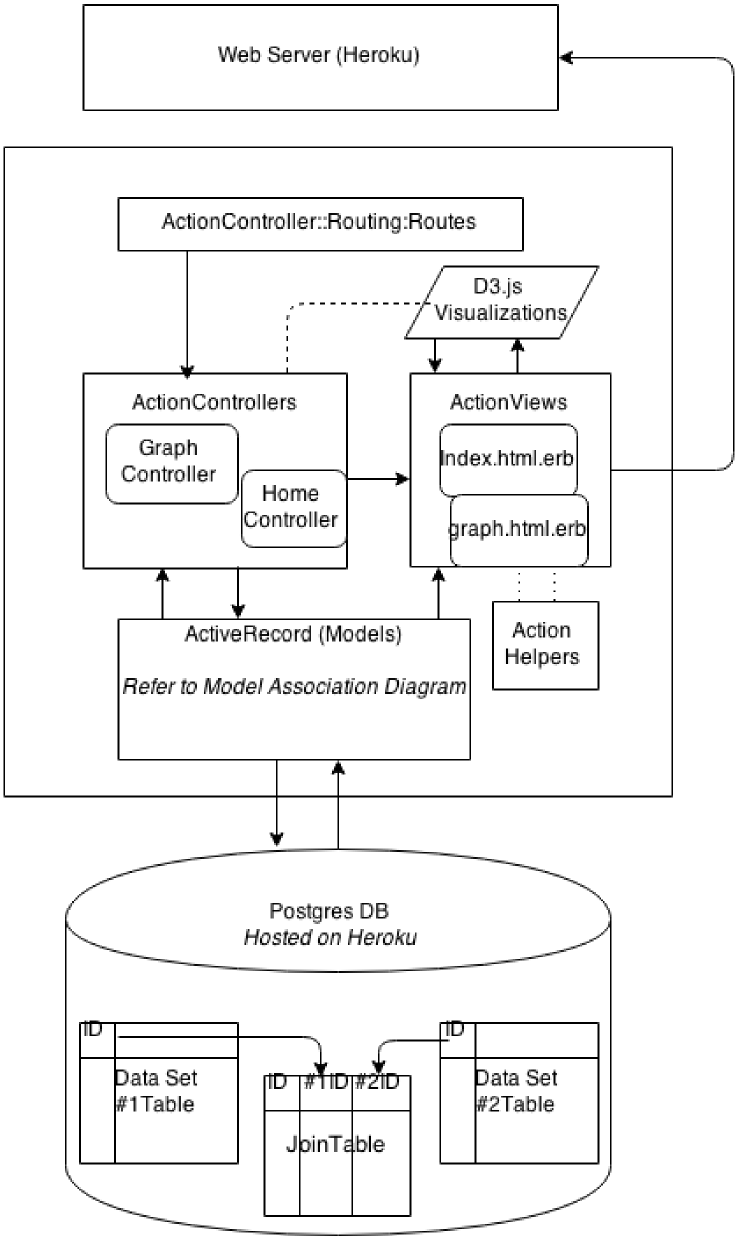
2: Glossary of Terms

API	Application program interface (API) is a set of routines, protocols, and tools for building software applications
Bootstrap	Bootstrap is a front-end framework that is compatible and scales with the latest versions of all major browsers
Correlation	The mutual relation of mathematical or statistical variables which tend to occur together
D3.js	D3.js is a JavaScript library that can use HTML, SVG, and CSS to create graphical representation of document-based data
Heroku	Cloud computing services that provides a platform to develop, run and manage Web applications.
HTTP	The Hypertext Transfer Protocol (HTTP) is an application protocol and the foundation of data communication for the World Wide Web
MVC (Rails)	MVC (Model, View, Controller) is the architectural framework Rails uses for implementing user interfaces
Object-relational Database Management System (ORDBMS)	ORDBMS have an object-oriented database model so that objects, classes, and inheritance are supported in database schemas and query language.
Open Source Project	Development model promoting universal access to a product’s design and its subsequent improvements by anyone
PostgreSQL	PostgreSQL is an open-source object-relational database management system (ORDBMS)
R + OpenCPU	HTTP API for data analysis
Ruby on Rails (Rails)	Rails is a web application development framework written in the Ruby language.
Ruby	Ruby is an object-oriented scripting language

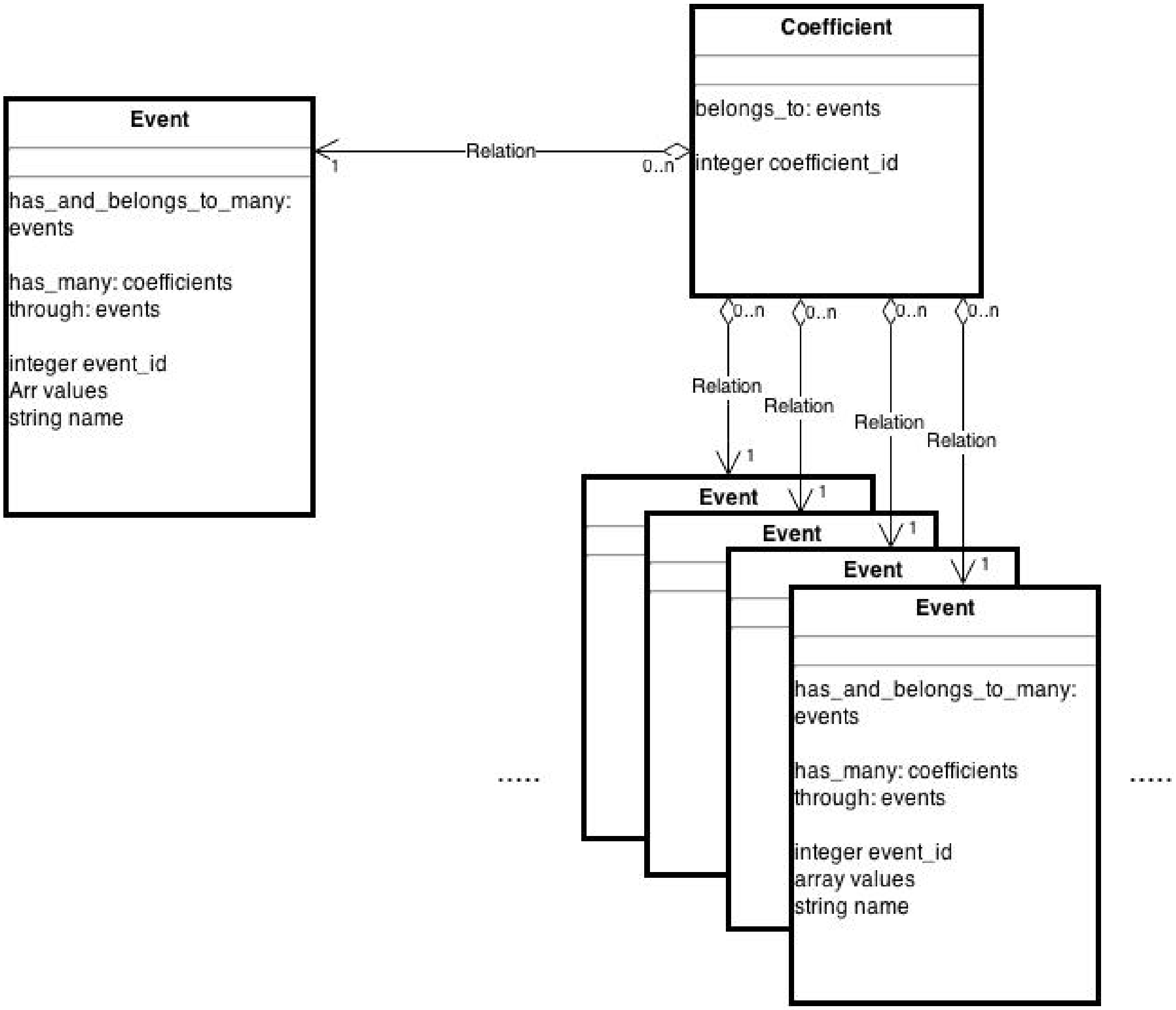
SODA Consumer API	The Socrata Open Data API allows a developer to programmatically access a wealth of open data resources from governments, non-profit organizations, and etc. from around the world
User Interface (UI)	Interface that allows users to interact with an application through visual indicators

3: System Architecture

3.1 Rails Architecture



3.2 Polymorphic Model Association Architecture



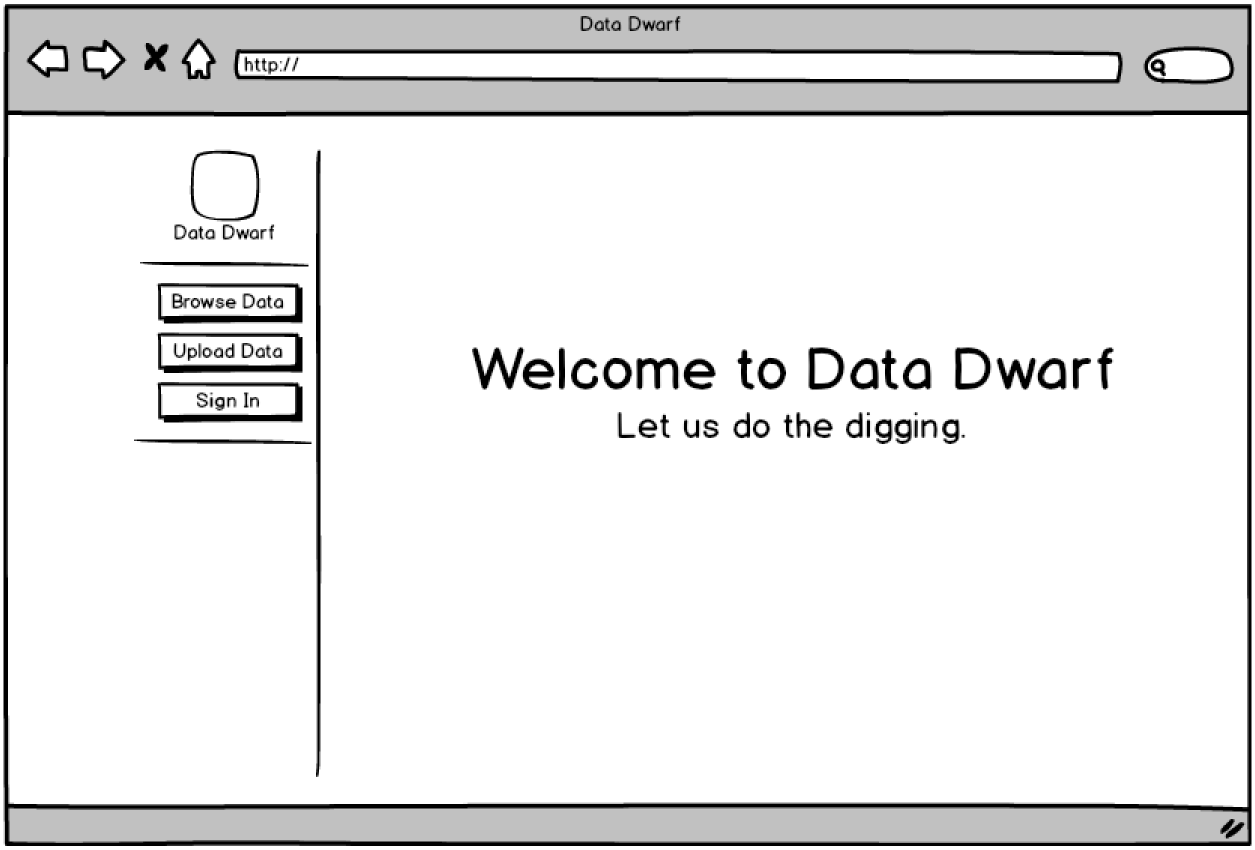
4: Correlation Algorithm

The correlation algorithm works comparing two data sets point-by-point. The more often the distance between such points is the same, the more correlated the two data sets are.

In explicit detail, let's say we have two data sets called D1 and D2. We subtract each point in D1 minus the corresponding points in D2. This gives us a bunch of differences. We want the difference that occurs the most often, the mode. We subtract the mode from all the previously calculated differences. This is, in effect, is the same as overlaying D2 onto D1, and then calculating how much each point in D2 differs from the corresponding point in D1. We find how much each point in D2 differs from those in D1 by calculating the square distance of each point in D2 from its corresponding point in D1. Then we find the average of all these square distances, and finally we take the square root of that average. This gets us our correlation value for any two data sets.

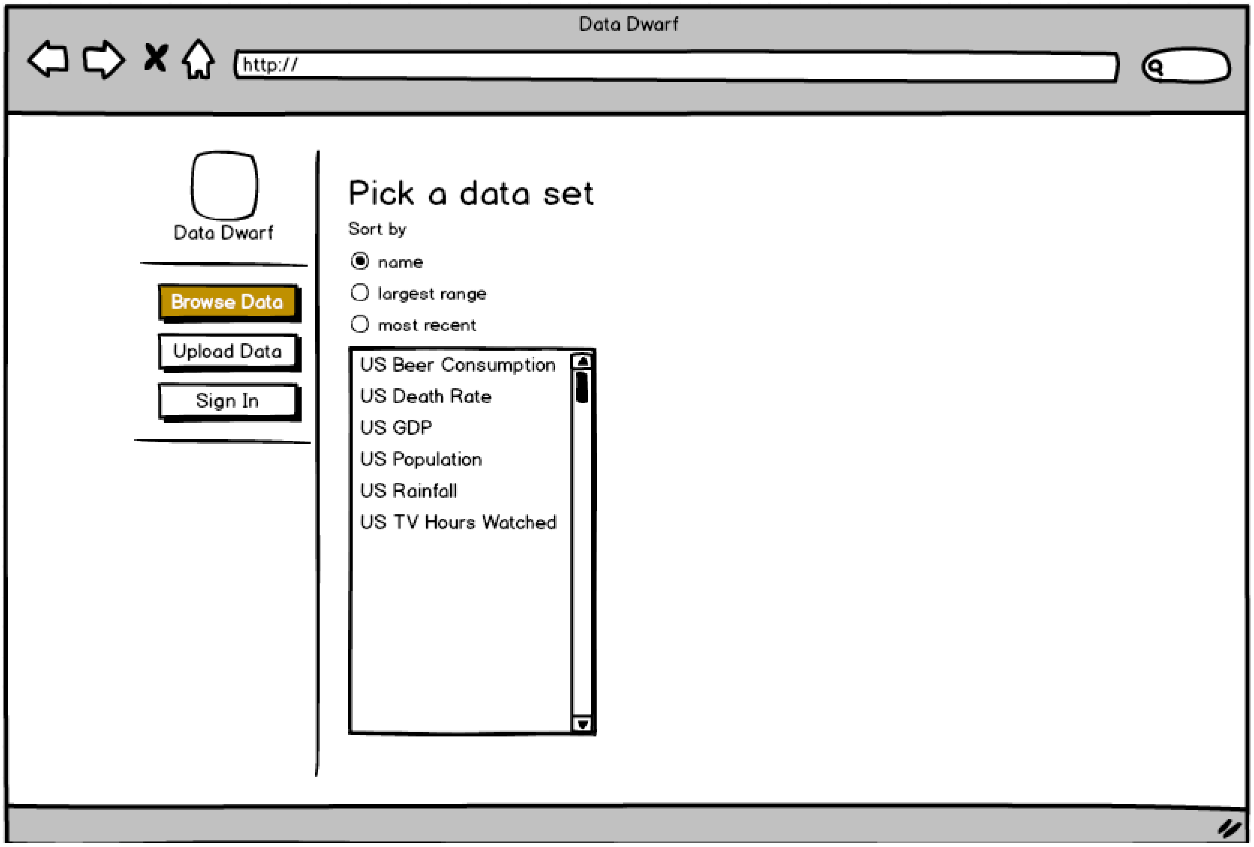
One worry about finding the correlation between any two data sets is the idea of trending. This problem occurs when two uncorrelated data sets are both given a slope or trend. Many correlation algorithms bluntly see that both data sets have a common trend and, because of that, rate them as being correlated, even though they aren't. A simple way to account for this problem is to detrend the data sets. One simple way to do this is to traverse through each data set, and subtract from each point the value of the previous point. This is how our algorithm will account for any such trending.

5: User Interface Mockups



Welcome to Data Dwarf

Let us do the digging.



Pick a data set

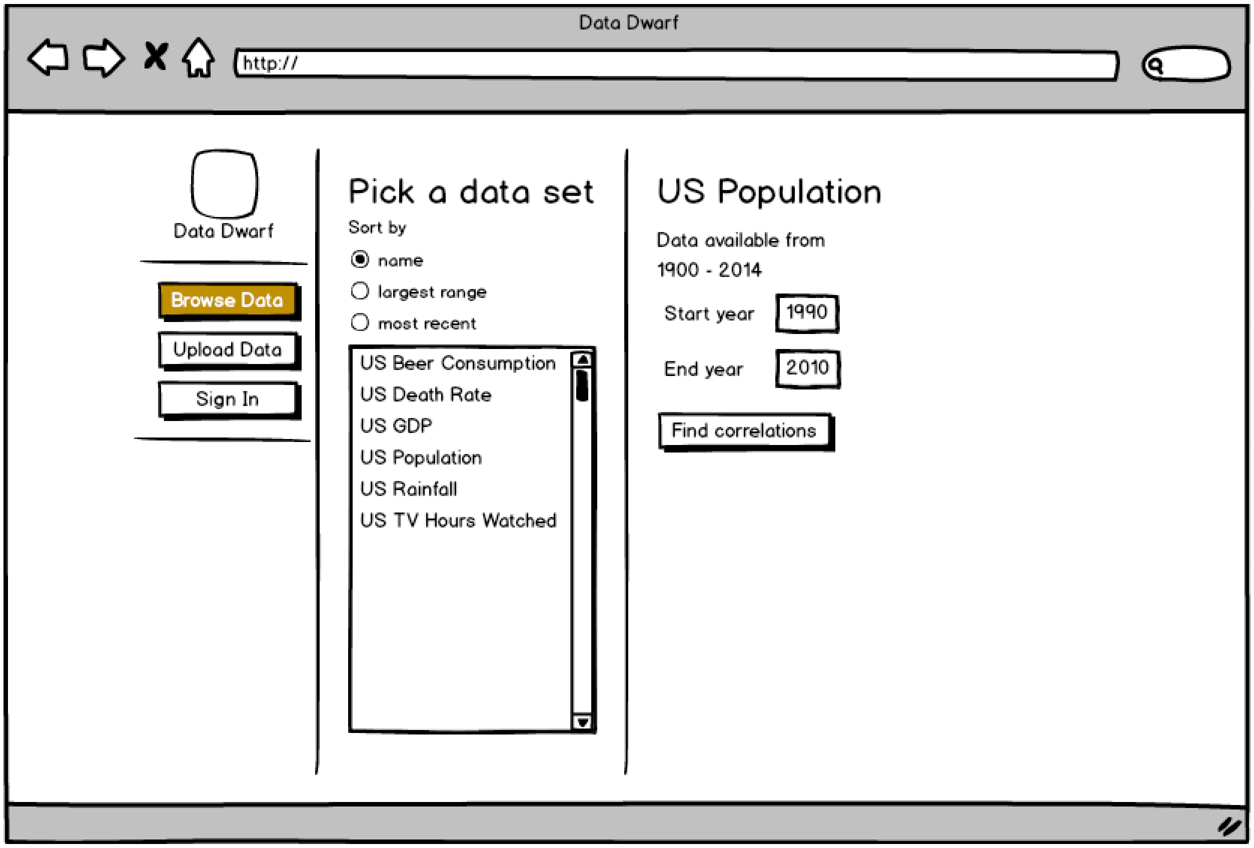
Sort by

● name

○ largest range

○ most recent

- US Beer Consumption
- US Death Rate
- US GDP
- US Population
- US Rainfall
- US TV Hours Watched



Pick a data set

Sort by

● name

○ largest range

○ most recent

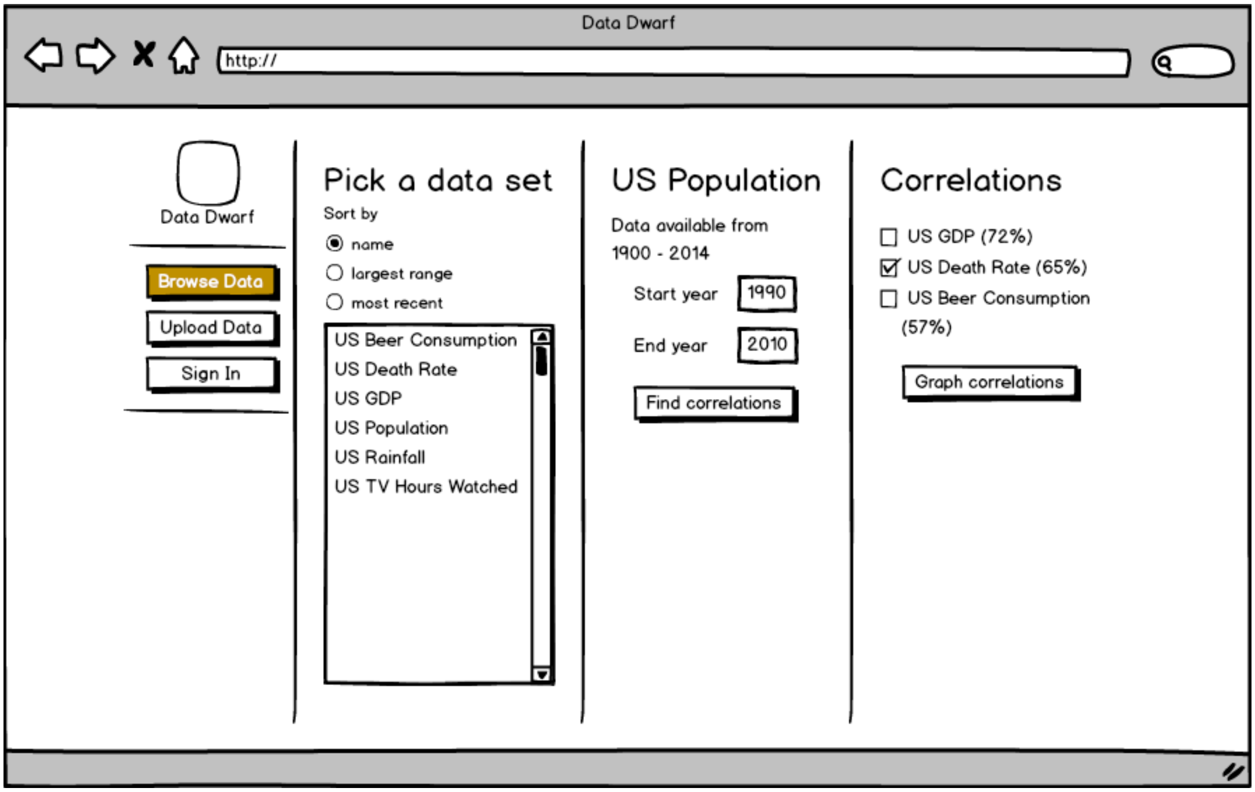
- US Beer Consumption
- US Death Rate
- US GDP
- US Population
- US Rainfall
- US TV Hours Watched

US Population

Data available from
1900 - 2014

Start year 1990

End year 2010



Pick a data set

Sort by

● name
○ largest range
○ most recent

- US Beer Consumption
- US Death Rate
- US GDP
- US Population
- US Rainfall
- US TV Hours Watched

US Population

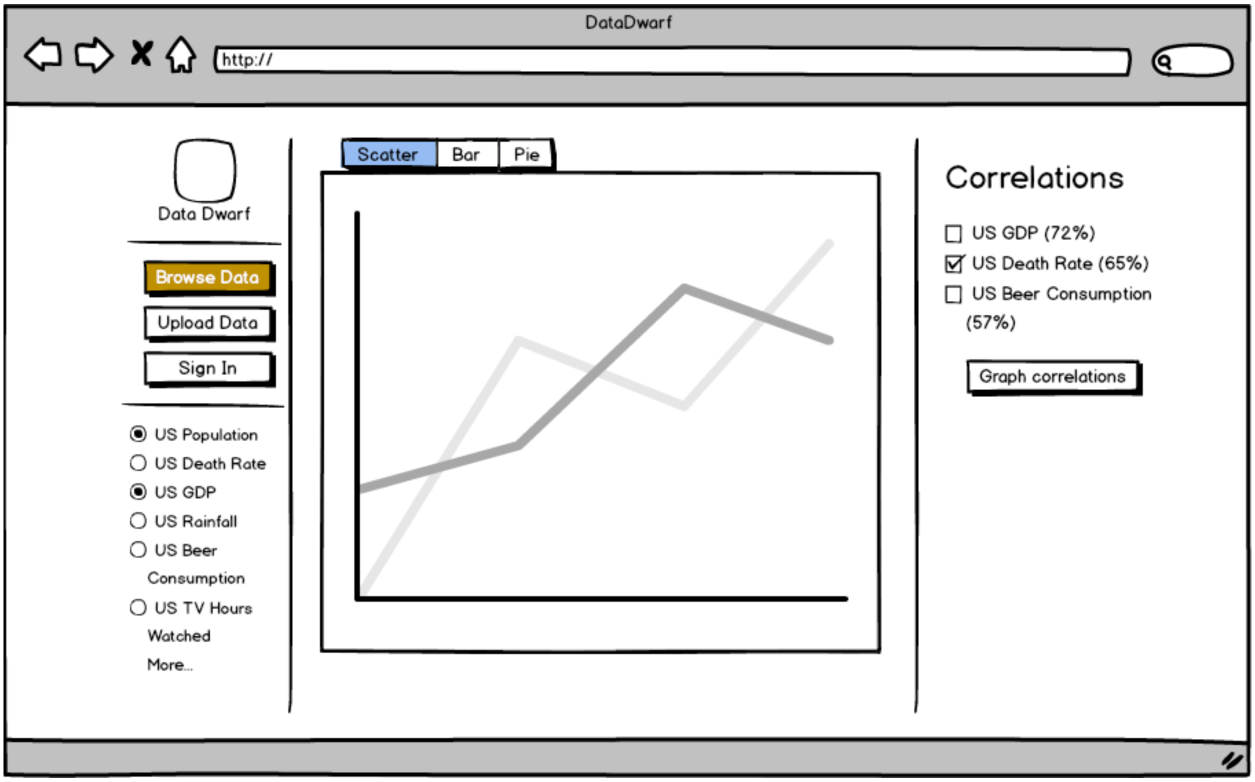
Data available from 1900 - 2014

Start year 1990

End year 2010

Correlations

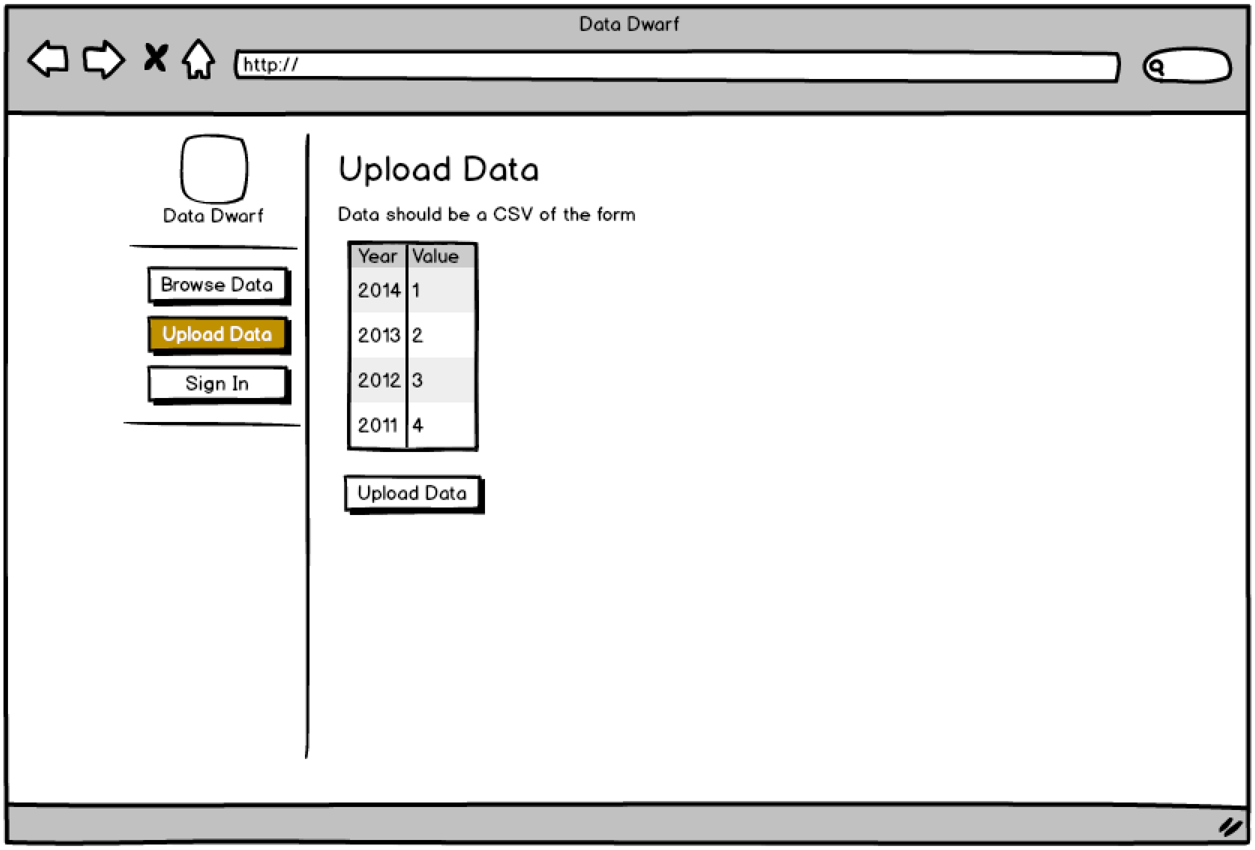
□ US GDP (72%)
☑ US Death Rate (65%)
□ US Beer Consumption (57%)



● US Population
○ US Death Rate
● US GDP
○ US Rainfall
○ US Beer Consumption
○ US TV Hours Watched
More...

Correlations

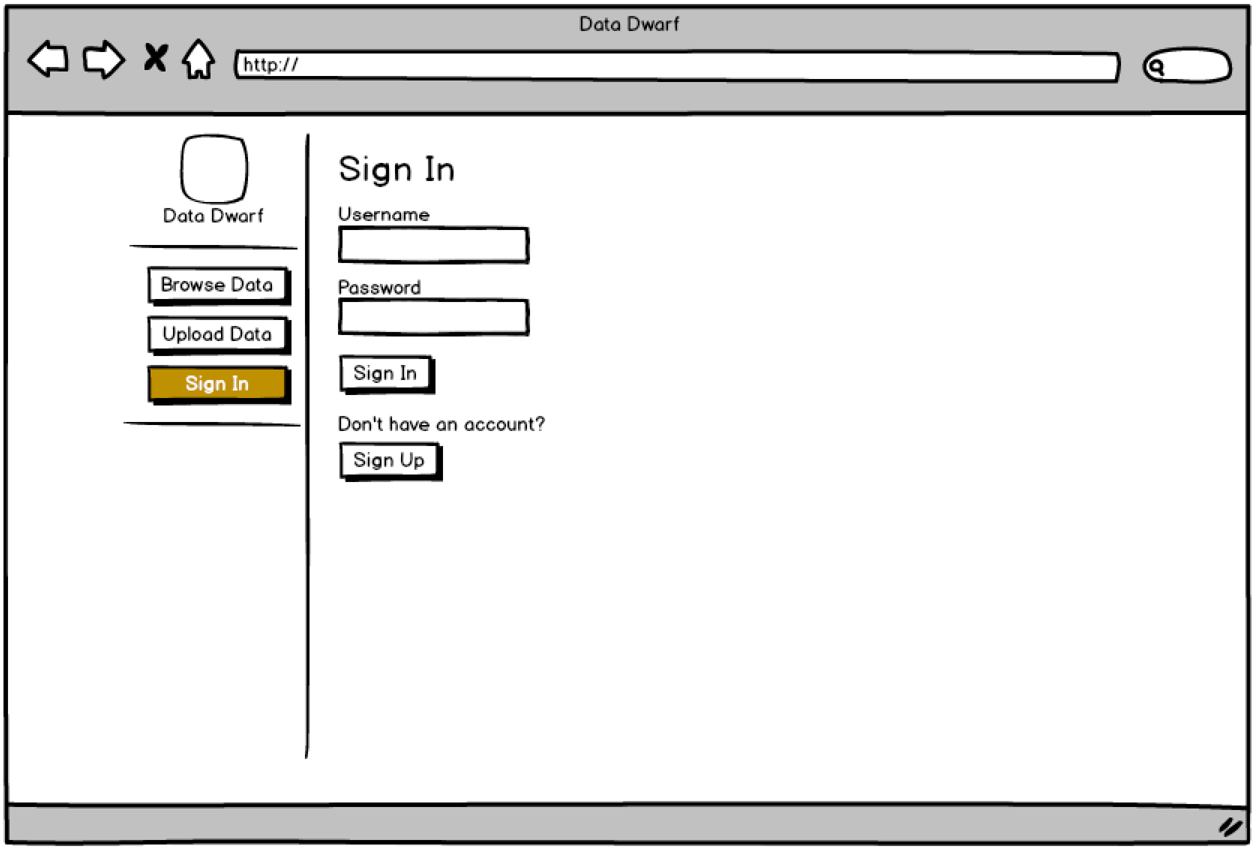
□ US GDP (72%)
☑ US Death Rate (65%)
□ US Beer Consumption (57%)



Upload Data

Data should be a CSV of the form

Year	Value
2014	1
2013	2
2012	3
2011	4

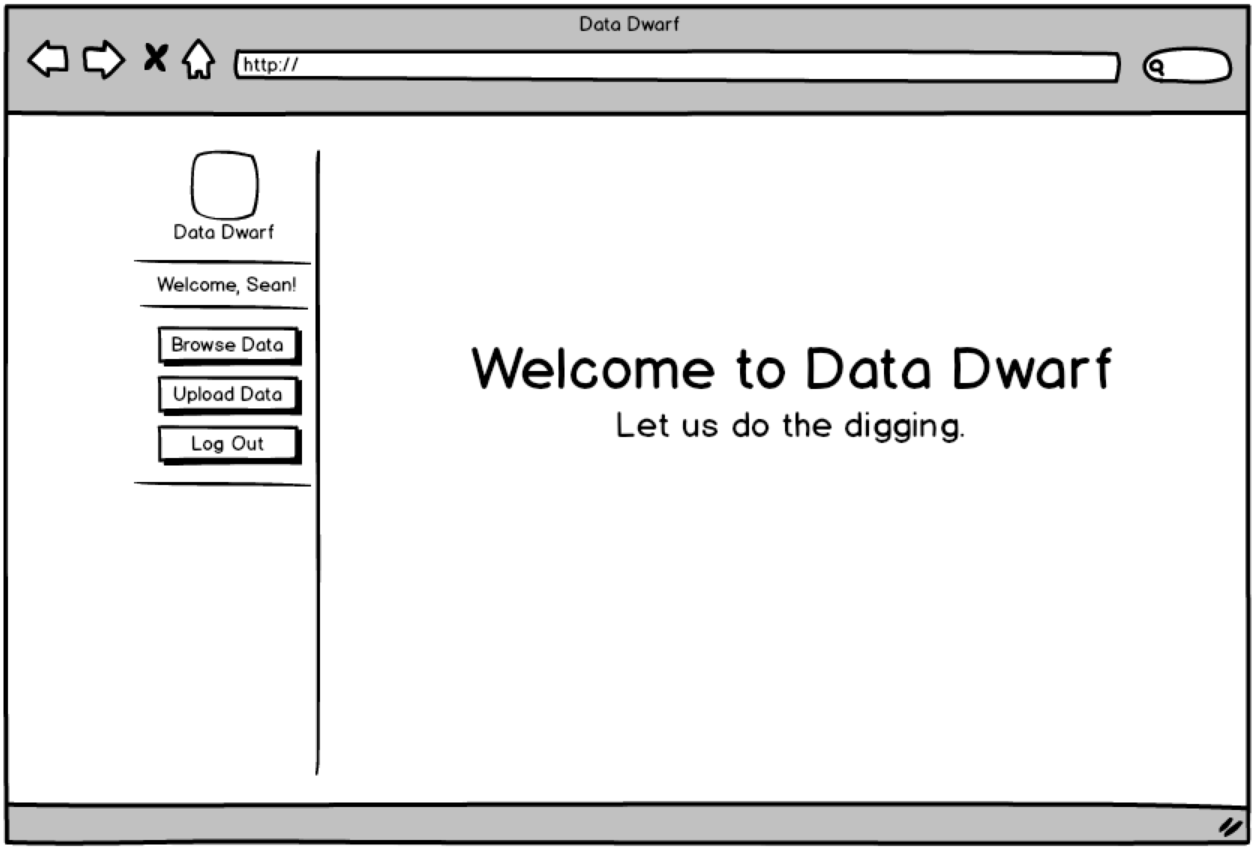


Sign In

Username

Password

Don't have an account?



Welcome, Sean!

Welcome to Data Dwarf

Let us do the digging.

6: Requirements

6.1 User Cases

Use Case 1: Generate Charts

Actors: User

Precondition: User is at the front page of the website and none of the data sets are selected

Flow of Events:

1. User clicks and selects a data set on the left toolbar
2. Charts with the data are generated for the user to select

Use Case 2: User Uploads Data

Actors: User

Precondition: User is at the front page and there is an empty column to put data in

Flow of Events:

Basic Path

1. User copies and pastes data from a spreadsheet (or manually enters in data)
2. User titles their data set
3. User hits “save”
4. Data set saved and added to the list of other data sets from which the user can now select

Alternative Path

1. User’s list of data points is too long for column, but user tries to paste it into the empty column anyways
2. Column expands to fit all data
3. Scroll bar appears to view all data entered

Use Case 3: User chooses length of each selected data set

Actors: User

Precondition: Each data set is customizable/editable

Flow of Events:

1. User selects first data set
2. Data set is added to graph
3. “Edit” button appears next to selected data set
4. User selects “Edit”
5. new menu appears
6. one of the options in the “Edit” menu allows the user to enter in the first and last points on the x-axis for that particular data set
7. User exits “Edit” menu

8. the selected data set scales appropriately on the graph

Use Case 4: User adjusts multiple selected data sets at a time
Actors: User
Precondition: Each data set has a check box next to it that the User can check and uncheck somewhere at the top there's an "Edit" menu which allows data set customization.
Flow of Events:

1. User selects multiple data sets
2. User clicks "graph" button
3. selected data sets all scale to fit on the same graph
4. the only check boxes that can be selected are those of the selected data sets
5. User clicks on (adds check to) one or more check boxes
6. User clicks on the "Edit" button somewhere at the top of the screen
7. "Edit" menu appears
8. User changes of the "Edit" menu options
9. User exits "Edit" menu
10. changes get applied to all checked data sets and the graph updates accordingly

Use Case 5: Query for data sets
Actors: User
Precondition: a text window is available to type in possible names of data sets not all data sets are shown on front page of website
Flow of Events:

1. User selects the text window
2. User begins typing the name of a possible data set
3. new window appears just below the text window and it has relevant available titles of data sets
4. User continues typing and the relevant titles window adjusts accordingly (kind of like when you type something into Google)
5. User finishes typing and clicks "enter"
6. all possible matches appear in a temporary box on the web page (it may be scrollable)
7. User selects one of the new data sets
8. New data set gets added to the main list of data sets on the web page

6.2 User Stories and Corresponding Acceptance Tests

Each numbered item in the following list is a user story with bullets detailing the tests required to ensure it is complete.

1. As a user, I can access the website from a browser
   - ❏ Open the website from its hostname on major browsers: Chrome, Internet Explorer, Firefox, and Safari
   - ❏ Verify that the hostname is resolved and the website data is loaded into each browser
2. As a user, I can find third-party data sets with high correlations to my own data set/s over time
   - ❏ Test from the front-end by uploading a CSV data file from the DataDwarf website
   - ❏ On the server side, verify calculated coefficients fall within a specified threshold ratio to the rest of the population of data coefficients
   - ❏ A loose verification of correlations can be confirmed by visually graphing and comparing each third-party data set with the input data set
3. As a user, I can access information on how third-party data sets are correlated to my own data set/s over the variable time
   - ❏ Make sure that correlations are only calculated over a change in time
   - ❏ Data mined from the Socrata Open Data API needs to be properly queried with keywords regarding time such as “Year”,”year”,”date”,”week”, etc.
4. As a user, I can upload my own files of data sets
   - ❏ Verify that the input CSV data file is received by the server
5. As a user, I can visualize my own data sets
   - ❏ After selecting the third-party datasets to be visualized with the input dataset along with the chart type, verify that the selected chart is generated with appropriate graphing of data points
6. As a user, I can change what data sets are shown on the charts
   - ❏ After initially generating a selected chart, a link should be available to go back to the available data sets. From then, the process can be repeated and third-party datasets can be selected to be visualized with the input dataset along with a different chart type.
   - ❏ The above test case is highly dependant on the implementation of User Story 6
7. As a user, I can change what type of chart the data is displayed on
   - ❏ After initially generating a selected chart, a link should be available to go back to the available data sets. From then, the process can be repeated and third-party datasets can be selected to be visualized with the input dataset along with a different chart type.
   - ❏ The above test case is highly dependant on the implementation of User Story 6
8. As a user, I can find the source of a third-party data set for download

    - ❏ After receiving third-party data sets with high correlations, I can follow the link given and easily identify and download the dataset
    - ❏ Verify the data stored in the database is identical to the dataset available at the link given

9. As a user, I can upload my own files entering my own data into column fields
    - ❏ After entering the data into the column fields, verify the integrity of the input data received by the server
10. As a user, I can filter selected data sets
    - ❏ Verify after selecting a filter, data set outputs unrelated to the filter are discarded from the data sets initially shown
11. As a user, I can search through available data sets
    - ❏ Verify keyword search returns data sets in the pool produced as well as relate to the searched keyword
12. As a user, I can download third-party data sets directly from the website
    - ❏ After selecting and requesting a dataset for download, verify the integrity of the data in comparison the the CSV file available on the linked source
13. As a user, I can use previously uploaded data if I have registered and logged in
    - ❏ After logging into the same account, verify that any user settings and data on the account has been preserved over multiple logins and changes
14. As a user, I can log into the site
    - ❏ While on the credential page, verify that the web server accepts input credentials
15. As a user, I can log out of the site
    - ❏ After signaling logout, verify that any user settings and data are not accessible in any way
16. As a user, I can register with the site
    - ❏ After registering an account, verify that the account can be logged into via the credentials entered during registration
17. As a user, I can select the scope of the visualization charts
    - ❏ Verify visualizations are constrained to the selected scope of the dependant variable by comparing the graphed data points with the constrained raw numerical data points of the data set
18. As a user, I can change the scope of data sets
    - ❏ Verify the datasets downloaded from the website is within the range of the scope requested
    - ❏ This test case is high dependant on User Case 12
19. As a user, I can customize the aesthetics of the visualization charts
    - ❏ Verify any prompts to customize a charts produce visually intended results

20. As a user, I can receive recommendations on the type of chart to visualize my data on
    - ❏ As a user, verify the charts recommended make intuitive sense for the data being visualized
    - ❏ For example, numerical continuous data does well visualized with a scatterplot and percentage data does well visualized with a pie graph
21. As a user, I can specify the span of time to correlate my input dataset over third-party datasets
    - ❏ Any dynamic correlation calculations required after changing the scope need to be performed only on data that encompasses the selected scope of time
22. As a user, I can download multiple third-party data sets directly from the website in a compressed format
    - ❏ Verify that a compressed file after extraction contains the data sets requested
    - ❏ Verify the compressed file downloaded can be extracted universally on major operating systems
23. As a user, I can upload multiple input data sets and receive third-party datasets that are highly correlated with all of my input data sets
    - ❏ Verify that we consider all of the input datasets by averaging their attributes and using the results with the standard correlation procedure with a singular input data set
24. As a user, I can find third-party data sets with high correlations to my own data set/s over variables in addition to time
    - ❏ In order to fulfill this user story, the database will need to be populated with additional data dependent on variables other than time
    - ❏ User input would be required to specify the dependant variable of their input data set
25. As a user, I can access the website from a phone or tablet
    - ❏ In order to fulfill this user story, additional applications will need to be created to be native to Android and iOS

Appendix A: Technologies Employed

- D3.js
- Bootstrap
- Heroku
- PostgreSQL
- Ruby on Rails

- SODA Consumer API