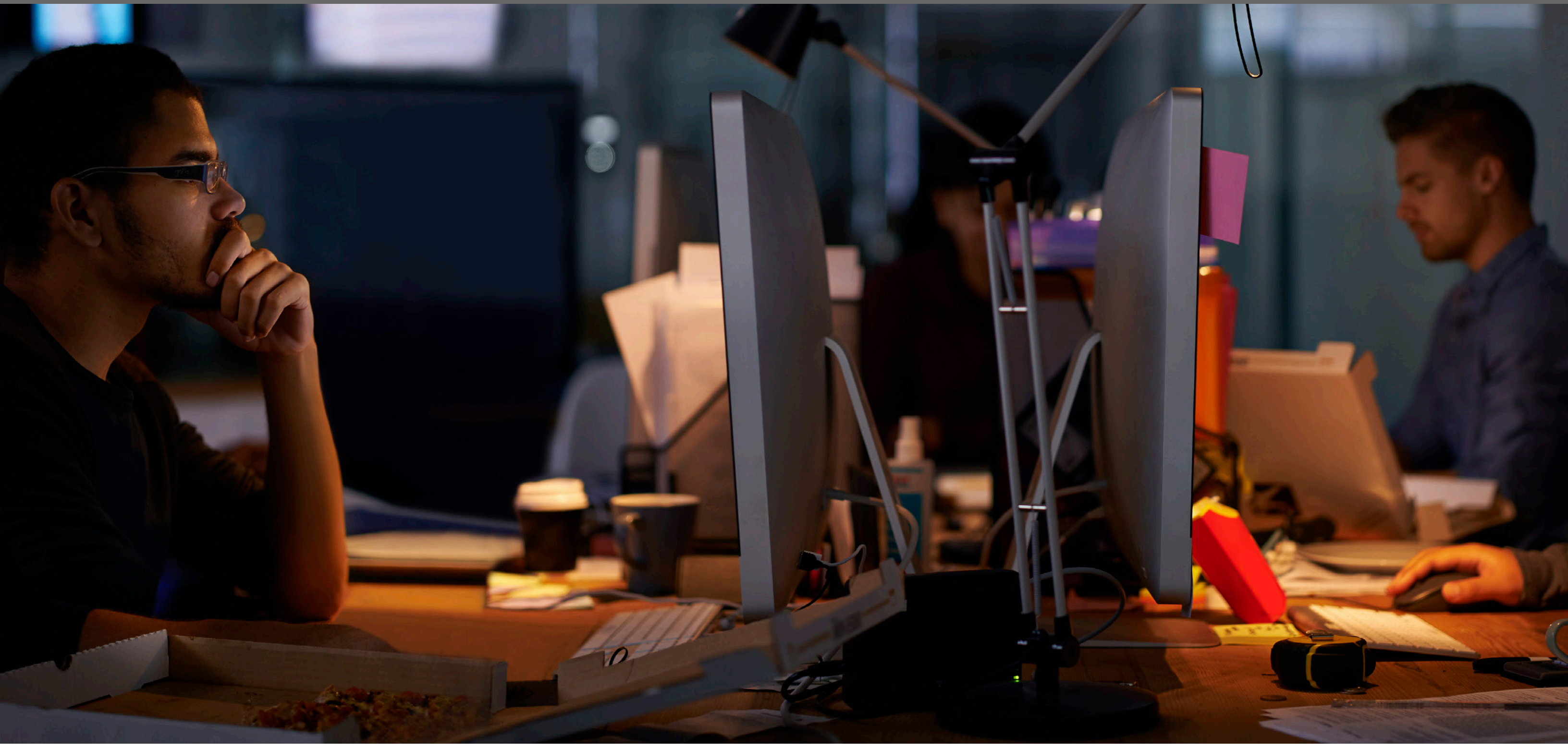
Socrata
The Data Experience Company™

WHITE PAPER

Getting Started with Open Data with ASP.NET MVC



An eBook by: Brian McKeiver, BizStream

Twitter: **@mcbeev**
Email: **bmckeiver@bizstream.com**
Blog: **http://www.mcbeev.com**

Table of Contents

About the Author – Brian McKeiver

As a senior solution architect and co-owner of **BizStream**, a web and custom software development company in Allendale, Michigan, I have been programming professionally for 14 years. I lead a team of software engineers who focus on Microsoft development platforms and technologies.

My passion for software development has not changed since day one of my first job. I enjoy helping customers across many different industries solve real-world business problems with technology. I have helped many small, medium, and large companies get past the pain of repetitive data entry, automate manual processes, integrate disparate systems, and make it so their websites and applications are lightning fast, easy to update, and a breeze to maintain.

Specialties include: ASP.NET 2.0, 3.5, 4.0, 4.5, Web Forms, MVC, C#, LINQ, Entity Framework, JavaScript, AJAX, jQuery, Kentico CMS, and SharePoint.

When not working, which is not often, I enjoy hanging out with my wife, chasing around my three children, and vigorously rooting for the Michigan State Spartans and Detroit Lions.

Introduction

This series of walkthroughs and tutorials is designed to illustrate creating a web application that works with government-based open data that is most easily accessed via tools powered by Socrata, a company specializing in open data. The core of this exercise is to walk through what it takes to setup a sample application, to connect to a remote dataset, and finally to consume that dataset inside of a web application. Tools such as Visual Studio, ASP.NET MVC, and C# will be used to create the application.

At the completion of the eBook you will be able to programmatically access government-based datasets easily inside of a Microsoft technology platform-based application. Doing so will open up a wealth of open data resources from governments, non-profits, and NGOs around the world.

When I was first asked to write this eBook, I was truly honored and humbled to do so. I will do my best to give you all of the knowledge needed to understand the concepts of creating a Microsoft ASP.NET application using the Socrata Open Data API.

The full source code of this eBook can be found at: **https://github.com/mcbeev/SocrataSodaNetSample**

A live demo of the sample application can be viewed at: **http://sodanetsample.azurewebsites.net**

Intended Audience

The intended audience of this eBook is experienced ASP.NET developers who are new to using an open data dataset and API. A developer interested in this tutorial should have the following skills:

- Familiar with an object oriented programming (OOP) language
- Familiar with ASP.NET web forms or MVC development methodologies
- Familiar with web development concepts (HTML, CSS, JavaScript)
- Familiar with JSON representation of data
- Familiar with REST API design pattern and related HTTP verbs
- Familiar with Microsoft Visual Studio IDE

Setup / Required Tools

Completing this exercise requires the following toolsets installed on a local workstation with internet connectivity:

- Microsoft Visual Studio 2013 Express or higher edition
- ASP.NET MVC 5
- ASP.NET Framework 4.5 / C#
- NuGet Package Manager

If your setup does not contain any of the above requirements please visit **http://www.visualstudio.com/** and **http://www.asp.net/mvc** to get started.

What is Open Data?

Wikipedia defines open data as the idea that certain data should be freely available to everyone to use and republish as they wish, without restrictions from copyright, patents, or other mechanisms of control. The goals of the open data movement are similar to those of other "open" movements, such as open source, open hardware, open content, and open access.

The philosophy behind open data has been long established, but the term "open data" itself is recent, gaining popularity with the rise of the internet and World Wide Web and, especially, with the launch of open data government initiatives such as Data.gov and Data.gov.uk.

To me the term "open data" is new, but the concept is something that I have worked with for more than 10 years. I have worked with local area government agencies here in the state of Michigan since 2002, helping them track data in a niche sector.

My company designed, implemented, and currently runs an application that provides juvenile facility management for these court-based agencies. The application is used in a dozen counties in Michigan and elsewhere in the United States. It is a central database for the different cases, the activity that happens on a daily basis inside of a facility, and assists in reporting different sets of data to the state. In essence the application has provided data from the county level to the state so that the state can analyze how well the facilities run and the results of the juvenile programs. We would be unable to do that without the data that the application collects.

There are many other documented examples of how open data is changing the way we think about trends, analytics, and measuring results. It is just one Google search away.

Who is Socrata?

Socrata is a privately-held cloud software company headquartered in Seattle's historic Pioneer Square, with offices in Washington, D.C. and London. The company was founded in 2007 with a goal of creating a cloud platform that enables public sector organizations to easily manage and share data. This platform makes the data accessible to anyone with a web connection, while also giving them the power to visualize and analyze the data with ease.

Socrata continues to build on its original concept, offering products that enable researchers, public policy experts, entrepreneurs, and busy citizens alike to better understand and make use of public sector services. It also continues to expand, developing and releasing new services that address key challenges faced by many organizations, such as performance measurement and financial accountability.

For more information you can visit their website at **http://www.socrata.com**.

What is Socrata Open Data API™ (SODA)?

SODA provides an open, standards-based, RESTful application programming interface to access government datasets. This interface can be called from many different client libraries on many different platforms including Google Android, Apple iOS, Java, PHP, and Microsoft .NET.

Because of the RESTful API design each open data API endpoint can be manipulated with HTTP methods of GET, POST, PUT, and DELETE. It is assumed for this eBook that you have knowledge and experience with using RESTful verbs.

An endpoint in a SODA API is simply a unique URL that represents an object or collection of objects. Every Socrata dataset, and even every data record, has its own endpoint. The endpoint is what you'll point your HTTP client at to interact with data resources.

http://data.cityofchicago.org/resource/uupf-x98q

The hostname of data.city.ofchicago.org is the main location of the hosted dataset. Each city, state, or government agency has its own. From there, all resources are accessed through a common base path of /resource/ along with their dataset identifier.

This paradigm holds true for every dataset in every SODA API. All datasets have a unique 4x4 identifier—eight alphanumeric characters split into 2 four-character phrases by a dash. For example, uupf-x98q is the identifier for business licenses. The 4x4 identifier can then be appended to the /resource/ endpoint to construct the API endpoint.

A SODA endpoint works with basic datatypes of String, Numeric, Boolean, Timestamp, and Location. Each endpoint also has the ability to filter, sort, group, and perform full text searching on its data. The endpoint itself is normally structured in the following format:

Read full details here: **http://www.socrata.com/products/open-data-api/**

SODA Developers Site

The SODA Developers site is the primary resource for developers looking to learn about Socrata and SODA. More background and documentation is found at the SODA Developer Site **http://dev.socrata.com/**.

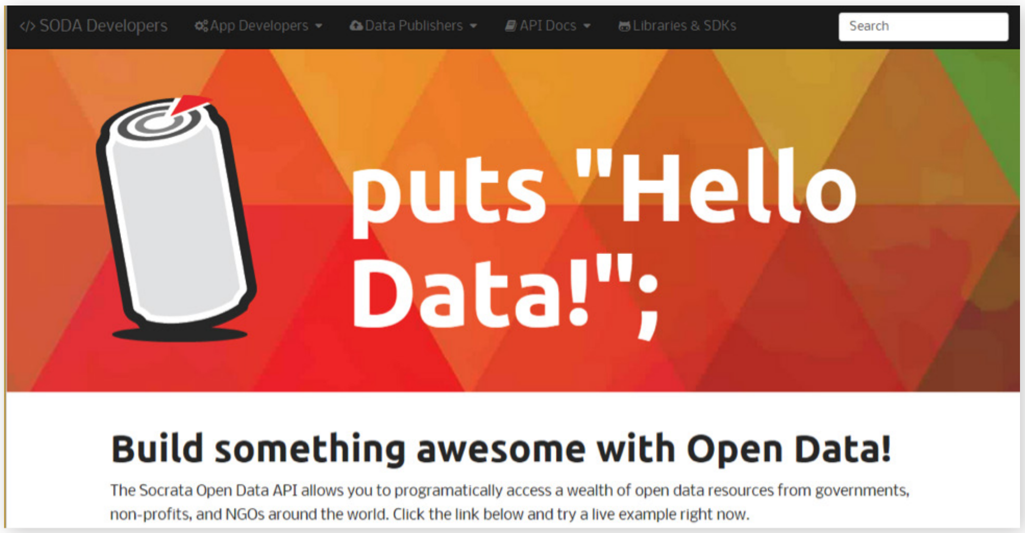
Included in the developer site is also a good Getting Started guide from Socrata that new developers should start with, **http://dev.socrata.com/consumers/getting-started.html**.

The developer site provides links to the different client libraries that are available to work with, documentation on how to query a SODA dataset, and where to go for help and support.

How to Choose Your Applications Open Data

There are many different locations that host open data. Socrata hosts more than 100 different data catalogs for governments, non-profits, and NGOs around the world, so finding an open data catalog to work with is easy:

- Check to see if your local government or state already has an open data site. Check your city or state website, or even Google "open data" and your government's name. You'll find something pretty quickly.



Build something awesome with Open Data!

The Socrata Open Data API allows you to programatically access a wealth of open data resources from governments, non-profits, and NGOs around the world. Click the link below and try a live example right now.

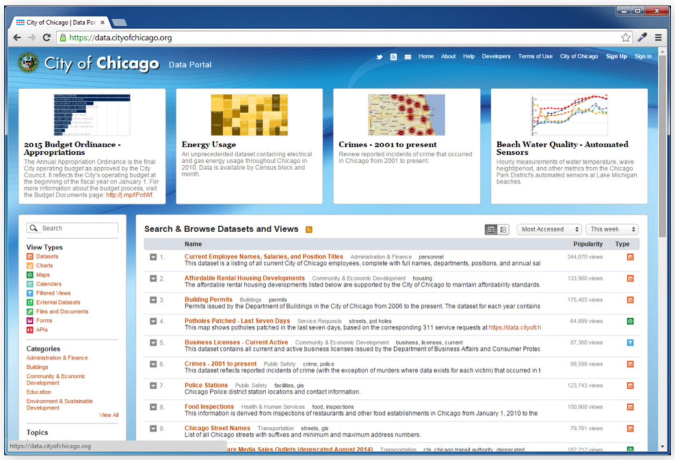
Online: www.socrata.com | Phone: 206 340 8008 | Twitter: @socrata

- Review the online listing of Socrata-powered open data sites. There's probably one near you.
- Check to see if there's a community group in your area with its own catalog hosted on **http://communities.socrata.com**.
- Check out **Data.gov**, a site for U.S. government data.

Once you're on your chosen open data site, scroll down to the data catalog and use the search box and browse filters to find datasets that interest you.

If your data site has custom API Foundry-managed APIs, you can use the "API" filter on the left-hand side to show only those custom APIs. But if your dataset doesn't have the red API icon, don't fret—every dataset is accessible via the SODA API.

For this eBook I chose to use the City of Chicago's data portal. Chicago's data portal, **https://data.cityofchicago.org/**, is powered by Socrata, which makes it easy to use, and I really enjoy deep dish pizza. What can I say?


City of Chicago | Data Por
https://data.cityofchicago.org
City of Chicago Data Portal
Home About Help Developers Terms of Use City of Chicago Sign Up Sign In

2015 Budget Ordinance - Appropriations
The Annual Appropriation Ordinance is the final City operating budget as approved by the City Council. It reflects the City's operating budget at the beginning of the fiscal year on January 1. For more information about the budget process, visit the Budget Documents page: http://j.mp/lPotWf.

Energy Usage
An unprecedented dataset containing electrical and gas energy usage throughout Chicago in 2010. Data is available by Census block and month.

Crimes - 2001 to present
Review reported incidents of crime that occurred in Chicago from 2001 to present.

Beach Water Quality - Automated Sensors
Hourly measurements of water temperature, wave height/period, and other metrics from the Chicago Park District's automated sensors at Lake Michigan beaches.

Search

View Types: Datasets, Charts, Maps, Calendars, Filtered Views, External Datasets, Files and Documents, Forms, APIs

Categories: Administration & Finance, Buildings, Community & Economic Development, Education, Environment & Sustainable Development, View All

Topics

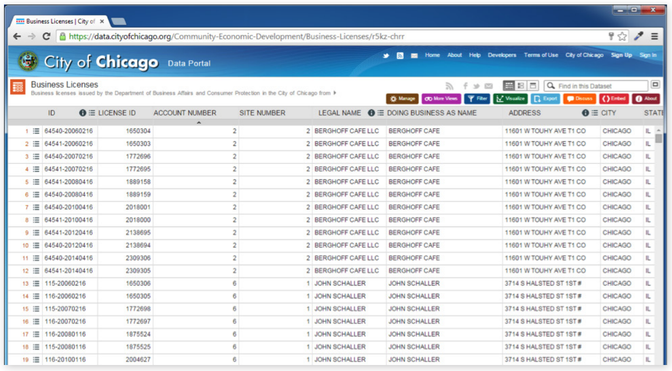
Search & Browse Datasets and Views — Most Accessed — This week

	Name	Popularity	Type
1.	Current Employee Names, Salaries, and Position Titles — Administration & Finance — personnel. This dataset is a listing of all current City of Chicago employees, complete with full names, departments, positions, and annual sal	344,870 views	
2.	Affordable Rental Housing Developments — Community & Economic Development — housing. The affordable rental housing developments listed below are supported by the City of Chicago to maintain affordability standards.	133,960 views	
3.	Building Permits — Buildings — permits. Permits issued by the Department of Buildings in the City of Chicago from 2006 to the present. The dataset for each year contains	175,403 views	
4.	Potholes Patched - Last Seven Days — Service Requests — streets, pot holes. This map shows potholes patched in the last seven days, based on the corresponding 311 service requests at https://data.cityofch	64,899 views	
5.	Business Licenses - Current Active — Community & Economic Development — business, licenses, current. This dataset contains all current and active business licenses issued by the Department of Business Affairs and Consumer Protec	87,360 views	
6.	Crimes - 2001 to present — Public Safety — crime, police. This dataset reflects reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in t	98,598 views	
7.	Police Stations — Public Safety — facilities, gis. Chicago Police district station locations and contact information.	123,743 views	
8.	Food Inspections — Health & Human Services — food, inspections. This information is derived from inspections of restaurants and other food establishments in Chicago from January 1, 2010 to the	100,868 views	
9.	Chicago Street Names — Transportation — streets, gis. List of all Chicago streets with suffixes and minimum and maximum address numbers.	79,761 views	

There are many types of datasets in an open data portal. The most common ones are listed on the left hand side of any data portal. In my experience the **Dataset** type is the most commonly used to integrate applications with, followed closely by the **Map** type.

For the remaining purposes of this eBook I chose the business licenses dataset. This dataset lists the business licenses issued by the Department of Business Affairs and Consumer Protection in the City of Chicago.

This dataset gave me the idea for creating an application to help me find a list of CrossFit® gyms in Chicago. Yes, I know most developers are not physical fitness fans. But, most of the developers I work with love CrossFit.


Business Licenses | City of
https://data.cityofchicago.org/Community-Economic-Development/Business-Licenses/r5kz-chrr
City of Chicago Data Portal
Home About Help Developers Terms of Use City of Chicago Sign Up Sign In
Find in this Dataset
Manage More Views Filter Visualize Export Discuss Embed About

Business Licenses
Business licenses issued by the Department of Business Affairs and Consumer Protection in the City of Chicago from

	ID	LICENSE ID	ACCOUNT NUMBER	SITE NUMBER	LEGAL NAME	DOING BUSINESS AS NAME	ADDRESS	CITY	STATE
1	64540-20060216	1650304	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
2	64541-20060216	1650303	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
3	64540-20070216	1772696	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
4	64541-20070216	1772695	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
5	64541-20080416	1889158	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
6	64540-20080416	1889159	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
7	64540-20100416	2018001	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
8	64541-20100416	2018000	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
9	64541-20120416	2138695	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
10	64540-20120416	2138694	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
11	64540-20140416	2309306	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
12	64541-20140416	2309305	2	2	BERGHOFF CAFE LLC	BERGHOFF CAFE	11601 W TOUHY AVE T1 CO	CHICAGO	IL
13	115-20060216	1650306	6	1	JOHN SCHALLER	JOHN SCHALLER	3714 S HALSTED ST 1ST #	CHICAGO	IL
14	116-20060216	1650305	6	1	JOHN SCHALLER	JOHN SCHALLER	3714 S HALSTED ST 1ST #	CHICAGO	IL
15	115-20070216	1772698	6	1	JOHN SCHALLER	JOHN SCHALLER	3714 S HALSTED ST 1ST #	CHICAGO	IL
16	116-20070216	1772697	6	1	JOHN SCHALLER	JOHN SCHALLER	3714 S HALSTED ST 1ST #	CHICAGO	IL
17	116-20080116	1875524	6	1	JOHN SCHALLER	JOHN SCHALLER	3714 S HALSTED ST 1ST #	CHICAGO	IL
18	115-20080116	1875525	6	1	JOHN SCHALLER	JOHN SCHALLER	3714 S HALSTED ST 1ST #	CHICAGO	IL
19	116-20100116	2004627	6	1	JOHN SCHALLER	JOHN SCHALLER	3714 S HALSTED ST 1ST #	CHICAGO	IL

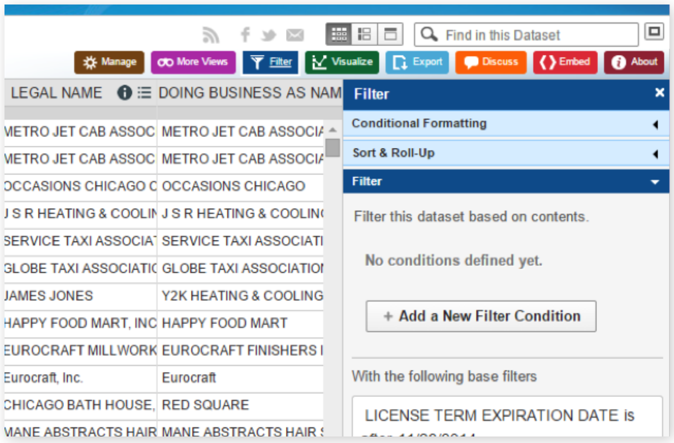
Accessing Dataset via SODA API

The **Socrata Open Data API (SODA)** provides programmatic access to this dataset, including the ability to **filter**, **query**, and **aggregate** data. Each open data portal allows you to also create your own saved filter or query against a dataset, as long as you register an account and login.

Determining API Endpoint

I created a simple filter, using the data portal at the City of Chicago's site, of the business licenses dataset to look for any legal name or DBA of CrossFit or gym.

To do that, I navigated to the endpoint I wanted to work with in a browser. From there, I simply clicked the Filter button towards the top left of the web page. Doing so opened up the filter sidebar showing the existing filters that were already added to the business license dataset.


Find in this Dataset
Manage More Views Filter Visualize Export Discuss Embed About
Filter
Conditional Formatting
Sort & Roll-Up
Filter
+ Add a New Filter Condition

LEGAL NAME	DOING BUSINESS AS NAM
METRO JET CAB ASSOC	METRO JET CAB ASSOCIA
METRO JET CAB ASSOC	METRO JET CAB ASSOCIA
OCCASIONS CHICAGO C	OCCASIONS CHICAGO
J S R HEATING & COOLIN	J S R HEATING & COOLIN
SERVICE TAXI ASSOCIAT	SERVICE TAXI ASSOCIATI
GLOBE TAXI ASSOCIATIC	GLOBE TAXI ASSOCIATION
JAMES JONES	Y2K HEATING & COOLING
HAPPY FOOD MART, INC	HAPPY FOOD MART
EUROCRAFT MILLWORK	EUROCRAFT FINISHERS I
Eurocraft, Inc.	Eurocraft
CHICAGO BATH HOUSE,	RED SQUARE
MANE ABSTRACTS HAIR	MANE ABSTRACTS HAIR S

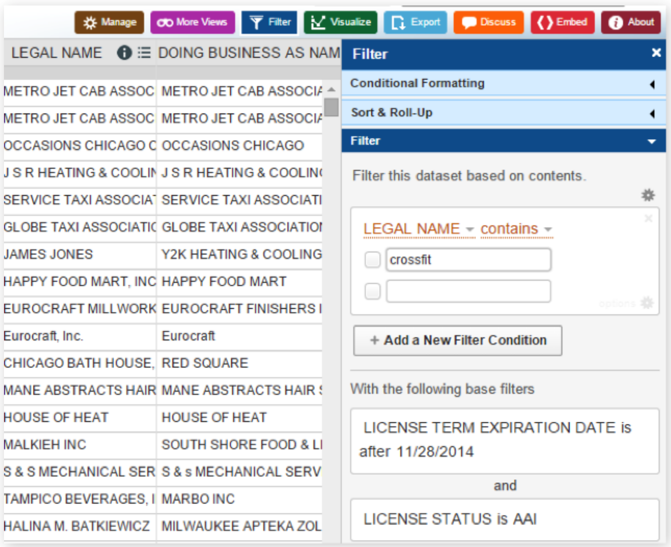
Filter this dataset based on contents.

No conditions defined yet.

With the following base filters

LICENSE TERM EXPIRATION DATE is

I then clicked the **Add a New Filter Condition** button to add a condition to filter on **Legal Name** that **contains** the value of "**CrossFit**". Now that the filter is correctly setup to show me all active businesses in the city of Chicago that had the word CrossFit in them, I can save that view of the data to get a unique 4x4 resource and name the view accordingly.



LEGAL NAME	DOING BUSINESS AS NAME
METRO JET CAB ASSOC	METRO JET CAB ASSOCIA
METRO JET CAB ASSOC	METRO JET CAB ASSOCIA
OCCASIONS CHICAGO C	OCCASIONS CHICAGO
J S R HEATING & COOLIN	J S R HEATING & COOLIN
SERVICE TAXI ASSOCIA	SERVICE TAXI ASSOCIATI
GLOBE TAXI ASSOCIATIO	GLOBE TAXI ASSOCIATIO
JAMES JONES	Y2K HEATING & COOLING
HAPPY FOOD MART, INC	HAPPY FOOD MART
EUROCRAFT MILLWORK	EUROCRAFT FINISHERS I
Eurocraft, Inc.	Eurocraft
CHICAGO BATH HOUSE,	RED SQUARE
MANE ABSTRACTS HAIR	MANE ABSTRACTS HAIR S
HOUSE OF HEAT	HOUSE OF HEAT
MALKIEH INC	SOUTH SHORE FOOD & LI
S & S MECHANICAL SER	S & S MECHANICAL SERV
TAMPICO BEVERAGES, I	MARBO INC
HALINA M. BATKIEWICZ	MILWAUKEE APTEKA ZOL

Filter

Conditional Formatting

Sort & Roll-Up

Filter

Filter this dataset based on contents.

LEGAL NAME contains

crossfit

+ Add a New Filter Condition

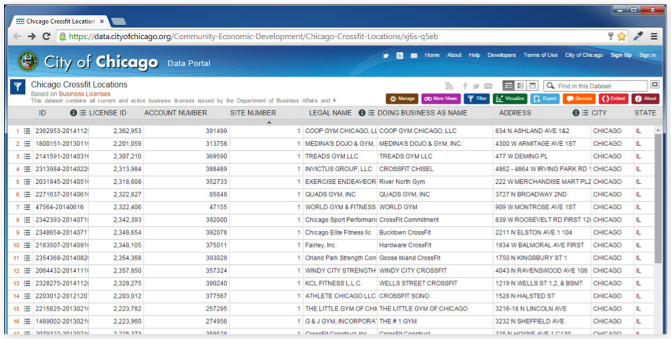
With the following base filters

LICENSE TERM EXPIRATION DATE is after 11/28/2014

and

LICENSE STATUS is AAI

The only requirement to save the view is to be logged in with your Socrata account. View the finished endpoint at: https://data.cityofchicago.org/resource/xj6s-q5eb.


Chicago Crossfit Location
https://data.cityofchicago.org/Community-Economic-Development/Chicago-Crossfit-Locations/xj6s-q5eb
City of Chicago Data Portal
Home
About
Help
Developers
Terms of Use
City of Chicago
Sign Up
Sign In
Find in this Dataset
Manage
More Views
Filter
Visualize
Export
Discuss
Embed
About

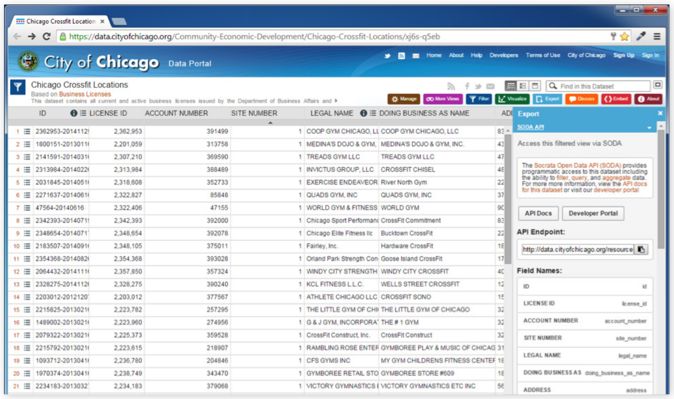
Chicago Crossfit Locations

Based on Business Licenses

This dataset contains all current and active business licenses issued by the Department of Business Affairs and

	ID	LICENSE ID	ACCOUNT NUMBER	SITE NUMBER	LEGAL NAME	DOING BUSINESS AS NAME	ADDRESS	CITY	STATE
1	2362953-2014112	2,362,953	391499	1	COOP GYM CHICAGO, LL	COOP GYM CHICAGO, LLC	834 N ASHLAND AVE 1&2	CHICAGO	IL
2	1800151-2013011	2,201,059	313758	1	MEDINA'S DOJO & GYM,	MEDINA'S DOJO & GYM, INC.	4300 W ARMITAGE AVE 1ST	CHICAGO	IL
3	2141591-2014031	2,307,210	369590	1	TREADS GYM LLC	TREADS GYM LLC	477 W DEMING PL	CHICAGO	IL
4	2313984-2014022	2,313,984	388489	1	INVICTUS GROUP, LLC	CROSSFIT CHISEL	4862 - 4864 W IRVING PARK RD 1	CHICAGO	IL
5	2031845-2014051	2,318,608	352733	1	EXERCISE ENDEAVEOR	River North Gym	222 W MERCHANDISE MART PLZ	CHICAGO	IL
6	2271637-2014061	2,322,827	85848	1	QUADS GYM, INC	QUADS GYM, INC	3727 N BROADWAY 2ND	CHICAGO	IL
7	47564-20140616	2,322,406	47155	1	WORLD GYM & FITNESS	WORLD GYM	909 W MONTROSE AVE 1ST	CHICAGO	IL
8	2342393-2014071	2,342,393	392000	1	Chicago Sport Performanc	CrossFit Commitment	839 W ROOSEVELT RD FIRST 12	CHICAGO	IL
9	2348654-2014071	2,348,654	392078	1	Chicago Elite Fitness llc	Bucktown CrossFit	2211 N ELSTON AVE 1 104	CHICAGO	IL
10	2183507-2014091	2,348,105	375011	1	Fairley, Inc.	Hardware CrossFit	1834 W BALMORAL AVE FIRST	CHICAGO	IL
11	2354368-2014082	2,354,368	393028	1	Orland Park Strength Con	Goose Island CrossFit	1750 N KINGSBURY ST 1	CHICAGO	IL
12	2064432-2014111	2,357,850	357324	1	WINDY CITY STRENGTH	WINDY CITY CROSSFIT	4043 N RAVENSWOOD AVE 106	CHICAGO	IL
13	2328275-2014112	2,328,275	390240	1	KCL FITNESS L.L.C.	WELLS STREET CROSSFIT	1219 N WELLS ST 1,2, & BSMT	CHICAGO	IL
14	2203012-2012120	2,203,012	377567	1	ATHLETE CHICAGO LLC	CROSSFIT SONO	1528 N HALSTED ST	CHICAGO	IL
15	2215825-2013021	2,223,782	257295	1	THE LITTLE GYM OF CHI	THE LITTLE GYM OF CHICAGO	3216-18 N LINCOLN AVE	CHICAGO	IL
16	1489002-2013021	2,223,960	274956	1	G & J GYM, INCORPORA	THE # 1 GYM	3232 N SHEFFIELD AVE	CHICAGO	IL

With the newly created resource endpoint, I can easily find the API endpoint that represents it by clicking on the **Export** button and looking at the **API Endpoint** field that the data portal makes available for me.


Chicago Crossfit Location
https://data.cityofchicago.org/Community-Economic-Development/Chicago-Crossfit-Locations/xj6s-q5eb
City of Chicago Data Portal
Home
About
Help
Developers
Terms of Use
City of Chicago
Sign Up
Sign In
Find in this Dataset
Manage
More Views
Filter
Visualize
Export
Discuss
Embed
About

Chicago Crossfit Locations

Based on Business Licenses

This dataset contains all current and active business licenses issued by the Department of Business Affairs and

	ID	LICENSE ID	ACCOUNT NUMBER	SITE NUMBER	LEGAL NAME	DOING BUSINESS AS NAME	ADI
1	2362953-2014112	2,362,953	391499	1	COOP GYM CHICAGO, LL	COOP GYM CHICAGO, LLC	83
2	1800151-2013011	2,201,059	313758	1	MEDINA'S DOJO & GYM,	MEDINA'S DOJO & GYM, INC.	43
3	2141591-2014031	2,307,210	369590	1	TREADS GYM LLC	TREADS GYM LLC	47
4	2313984-2014022	2,313,984	388489	1	INVICTUS GROUP, LLC	CROSSFIT CHISEL	48
5	2031845-2014051	2,318,608	352733	1	EXERCISE ENDEAVEOR	River North Gym	22
6	2271637-2014061	2,322,827	85848	1	QUADS GYM, INC	QUADS GYM, INC	37
7	47564-20140616	2,322,406	47155	1	WORLD GYM & FITNESS	WORLD GYM	90
8	2342393-2014071	2,342,393	392000	1	Chicago Sport Performanc	CrossFit Commitment	83
9	2348654-2014071	2,348,654	392078	1	Chicago Elite Fitness llc	Bucktown CrossFit	22
10	2183507-2014091	2,348,105	375011	1	Fairley, Inc.	Hardware CrossFit	18
11	2354368-2014082	2,354,368	393028	1	Orland Park Strength Con	Goose Island CrossFit	17
12	2064432-2014111	2,357,850	357324	1	WINDY CITY STRENGTH	WINDY CITY CROSSFIT	40
13	2328275-2014112	2,328,275	390240	1	KCL FITNESS L.L.C.	WELLS STREET CROSSFIT	12
14	2203012-2012120	2,203,012	377567	1	ATHLETE CHICAGO LLC	CROSSFIT SONO	15
15	2215825-2013021	2,223,782	257295	1	THE LITTLE GYM OF CHI	THE LITTLE GYM OF CHICAGO	32
16	1489002-2013021	2,223,960	274956	1	G & J GYM, INCORPORA	THE # 1 GYM	32
17	2079322-2013021	2,225,373	359528	1	CrossFit Construct, Inc.	CrossFit Construct	32
18	2215792-2013021	2,223,615	218907	1	RAMBLING ROSE ENTER	GYMBOREE PLAY & MUSIC OF CHICAG	31
19	1093712-2013041	2,236,780	204846	1	CFS GYMS INC	MY GYM CHILDRENS FITNESS CENTER	18
20	1970374-2013041	2,238,749	343470	1	GYMBOREE RETAIL STO	GYMBOREE STORE #609	18
21	2234183-2013032	2,234,183	379068	1	VICTORY GYMNASTICS I	VICTORY GYMNASTICS ETC INC	56

Export

SODA API

Access this filtered view via SODA

The Socrata Open Data API (SODA) provides programmatic access to this dataset including the ability to filter, query, and aggregate data. For more more information, view the API docs for this dataset or visit our developer portal

API Docs | Developer Portal

API Endpoint:

http://data.cityofchicago.org/resource

Field Names:

ID	id
LICENSE ID	license_id
ACCOUNT NUMBER	account_number
SITE NUMBER	site_number
LEGAL NAME	legal_name
DOING BUSINESS AS	doing_business_as_name
ADDRESS	address

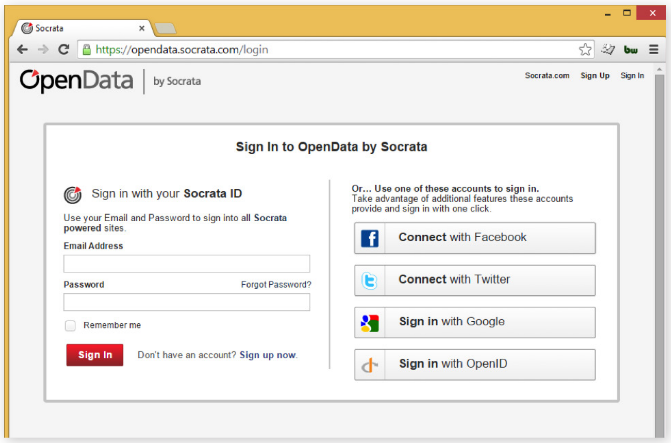
Socrata Setup

One of the first things you will need to do is to create your account on the Socrata Open Data platform.

Create Your Account

To create your account, navigate to the login page at https://opendata.socrata.com/login. This page will allow to register an account so that you can create a Socrata application and get the required app token.

From the login page click the **Sign up now** link towards the bottom of the form to create a Socrata account. Note you can also choose to use an existing social network account.



Sign In to OpenData by Socrata

Sign in with your Socrata ID

Use your Email and Password to sign into all Socrata powered sites.

Email Address

Password

Forgot Password?

Remember me

Sign In

Don't have an account? Sign up now.

Or... Use one of these accounts to sign in.
Take advantage of additional features these accounts provide and sign in with one click.

Connect with Facebook

Connect with Twitter

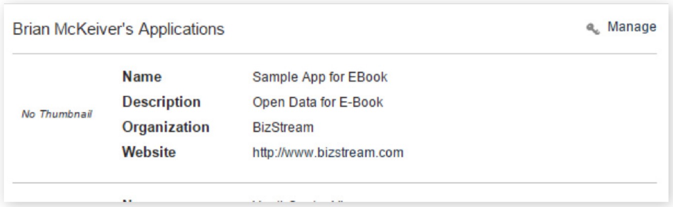
Sign in with Google

Sign in with OpenID

Fill in the required form fields and click **Create My Account**. Once your account is created you should land on the Profile page.

Register Application and App Token

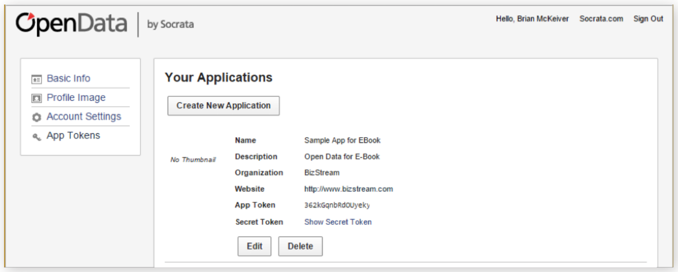
From the Profile page you can register an Application by clicking the **Manage** link on the title bar of the Applications widget.



Brian McKeiver's Applications — Manage

No Thumbnail	Name	Sample App for EBook
	Description	Open Data for E-Book
	Organization	BizStream
	Website	http://www.bizstream.com

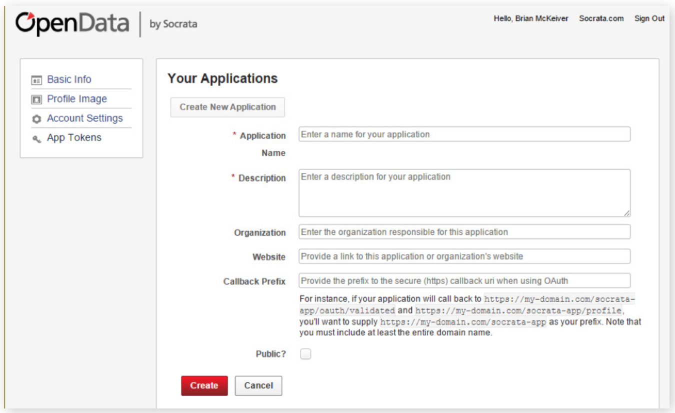
Once the **Manage** link is clicked, you are taken to the **Your Applications** page. From this page you can either create a new application or edit an existing one.



Your Applications

No Thumbnail	Name	Sample App for EBook
	Description	Open Data for E-Book
	Organization	BizStream
	Website	http://www.bizstream.com
	App Token	362kGqnbRdOUyeky
	Secret Token	Show Secret Token

To create your first application click on **Create New Application**.



Your Applications

* Application Name	Enter a name for your application
* Description	Enter a description for your application
Organization	Enter the organization responsible for this application
Website	Provide a link to this application or organization's website
Callback Prefix	Provide the prefix to the secure (https) callback uri when using OAuth
Public?	☐

For instance, if your application will call back to `https://my-domain.com/socrata-app/oauth/validated` and `https://my-domain.com/socrata-app/profile`, you'll want to supply `https://my-domain.com/socrata-app` as your prefix. Note that you must include at least the entire domain name.

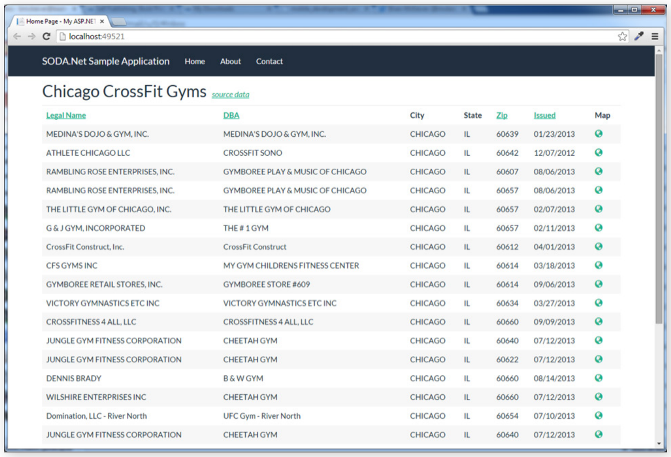
Fill in the required fields of Application Name and Description and click **Create**. Make sure that you have a unique application name or you will see an error message stating that the name is taken. All other fields are optional and don't impact the scope of this eBook.

After the Application is created, you can optionally upload a thumbnail for your Application. If you are planning on working with data in your ASP.NET application only, this thumbnail and the other fields are not used.

Click **App Tokens** in the left-hand side menu to return to Your Applications. Your new application will be listed and should have a freshly generated app token.

The Sample Application

Using this eBook, you will build a sample application using the SODA.NET open source library for ASP.NET. The sample application is a business directory listing of all of the CrossFit gyms in Chicago, IL. View the live version of the sample application can at: **http://sodanetsample.azurewebsites.net/**



Chicago CrossFit Gyms

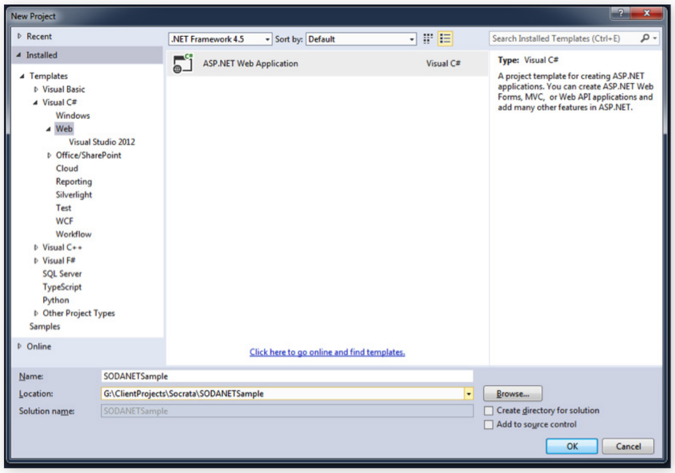
Legal Name	DBA	City	State	Zip	Issued	Map
MEDINA'S DOJO & GYM, INC.	MEDINA'S DOJO & GYM, INC.	CHICAGO	IL	60639	01/23/2013	●
ATHLETE CHICAGO LLC	CROSSFIT SONO	CHICAGO	IL	60642	12/07/2012	●
RAMBLING ROSE ENTERPRISES, INC.	GYMBOREE PLAY & MUSIC OF CHICAGO	CHICAGO	IL	60607	08/06/2013	●
RAMBLING ROSE ENTERPRISES, INC.	GYMBOREE PLAY & MUSIC OF CHICAGO	CHICAGO	IL	60657	08/06/2013	●
THE LITTLE GYM OF CHICAGO, INC.	THE LITTLE GYM OF CHICAGO	CHICAGO	IL	60657	02/07/2013	●
G & J GYM, INCORPORATED	THE # 1 GYM	CHICAGO	IL	60657	02/11/2013	●
CrossFit Construct, Inc.	CrossFit Construct	CHICAGO	IL	60612	04/01/2013	●
CFS GYMS INC	MY GYM CHILDRENS FITNESS CENTER	CHICAGO	IL	60614	03/18/2013	●
GYMBOREE RETAIL STORES, INC.	GYMBOREE STORE #609	CHICAGO	IL	60614	09/06/2013	●
VICTORY GYMNASTICS ETC INC	VICTORY GYMNASTICS ETC INC	CHICAGO	IL	60634	03/27/2013	●
CROSSFITNESS 4 ALL, LLC	CROSSFITNESS 4 ALL, LLC	CHICAGO	IL	60660	09/09/2013	●
JUNGLE GYM FITNESS CORPORATION	CHEETAH GYM	CHICAGO	IL	60640	07/12/2013	●
JUNGLE GYM FITNESS CORPORATION	CHEETAH GYM	CHICAGO	IL	60622	07/12/2013	●
DENNIS BRADY	B & W GYM	CHICAGO	IL	60660	08/14/2013	●
WILSHIRE ENTERPRISES INC	CHEETAH GYM	CHICAGO	IL	60660	07/12/2013	●
Domination, LLC - River North	UFC Gym - River North	CHICAGO	IL	60654	07/10/2013	●
JUNGLE GYM FITNESS CORPORATION	CHEETAH GYM	CHICAGO	IL	60640	07/12/2013	●

For scope purposes, let's focus on what it takes to read information from an open data API and present it on the web page. Inserting and updating data will not be discussed in the contents of this book, but know that it is possible to add and modify data via the API.

Create your Solution

Now that you have chosen your dataset and know a bit about accessing an endpoint, let's work on creating a sample application to put it all together. To do this, open up Visual Studio 2013. Create a new project by following these steps.

1. Create a new C# Web project named SODANETSample.



New Project

.NET Framework 4.5 · Sort by: Default · Search Installed Templates (Ctrl+E)

Recent / Installed / Templates / Visual Basic / Visual C# / Windows / Web / Visual Studio 2012 / Office/SharePoint / Cloud / Reporting / Silverlight / Test / WCF / Workflow / Visual C++ / Visual F# / SQL Server / TypeScript / Python / Other Project Types / Samples / Online

ASP.NET Web Application — Visual C#

Type: Visual C#

A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

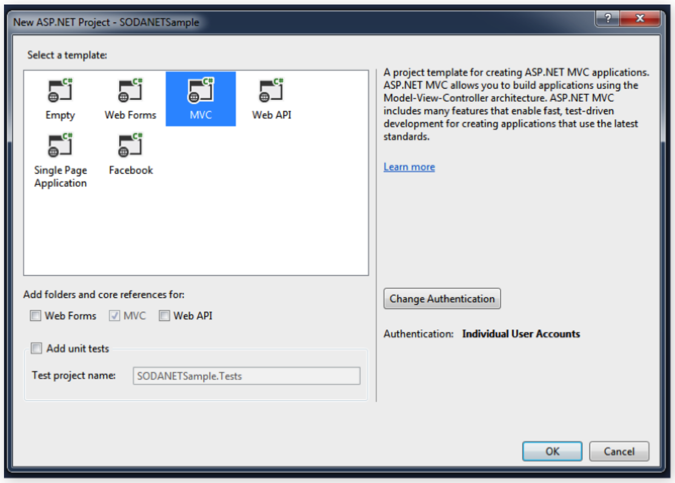
Click here to go online and find templates.

Name: SODANETSample
Location: G:\ClientProjects\Socrata\SODANETSample
Solution name: SODANETSample

☐ Create directory for solution
☐ Add to source control

OK · Cancel

2. In the New ASP.NET Project dialog box select the MVC template.



New ASP.NET Project - SODANETSample

Select a template: Empty · Web Forms · MVC · Web API · Single Page Application · Facebook

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Learn more

Add folders and core references for: ☐ Web Forms ☑ MVC ☐ Web API

☐ Add unit tests

Test project name: SODANETSample.Tests

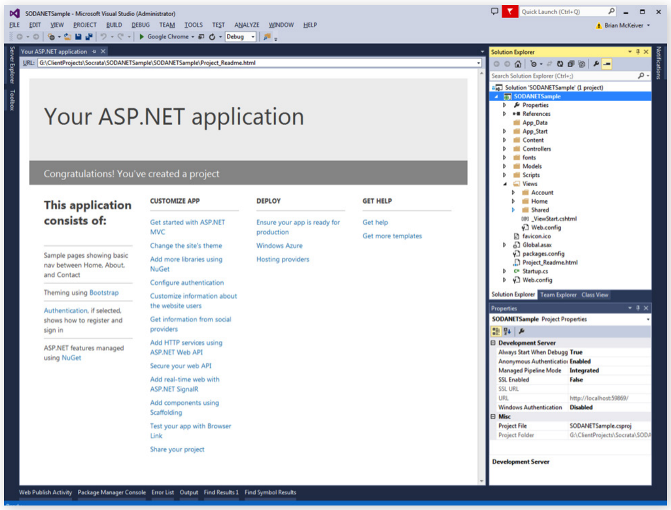
Change Authentication

Authentication: **Individual User Accounts**

OK · Cancel

3. Click OK to create the project.

After the project is created you should have a new fresh project that uses ASP.NET MVC 5. The Model-View-Controller architecture and convention is already setup for you. The project also contains just about everything you would need to create a modern web application, including references and assets to jQuery, Bootstrap, and Modernizr.



SODANETSample - Microsoft Visual Studio (Administrator)

URL: G:\ClientProjects\Socrata\SODANETSample\SODANETSample\Project_Readme.html

Your ASP.NET application

Congratulations! You've created a project

This application consists of:

Sample pages showing basic nav between Home, About, and Contact

Theming using Bootstrap

Authentication, if selected, shows how to register and sign in

ASP.NET features managed using NuGet

CUSTOMIZE APP: Get started with ASP.NET MVC · Change the site's theme · Add more libraries using NuGet · Configure authentication · Customize information about the website users · Get information from social providers · Add HTTP services using ASP.NET Web API · Secure your web API · Add real-time web with ASP.NET SignalR · Add components using Scaffolding · Test your app with Browser Link · Share your project

DEPLOY: Ensure your app is ready for production · Windows Azure · Hosting providers

GET HELP: Get help · Get more templates

Solution Explorer: Solution 'SODANETSample' (1 project) / SODANETSample / Properties / References / App_Data / App_Start / Content / Controllers / fonts / Models / Scripts / Views / Account / Home / Shared / _ViewStart.cshtml / Web.config / favicon.ico / Global.asax / packages.config / Project_Readme.html / Startup.cs / Web.config

Properties: SODANETSample Project Properties

Development Server	
Always Start When Debugg	True
Anonymous Authenticatio	Enabled
Managed Pipeline Mode	Integrated
SSL Enabled	False
SSL URL	
URL	http://localhost:59869/
Windows Authentication	Disabled
Misc	
Project File	SODANETSample.csproj
Project Folder	G:\ClientProjects\Socrata\SODA

At this point you can continue using the default project organization structure and theme for the application. Or you can modify the structure and theme to match your own coding standards and best practices.

I like to move the fonts, scripts, and other assets type folders in my MVC projects to the Content directory to clean it up a little. I also make an Helpers folder to hold utility classes and other miscellaneous class files. In the official demo of the sample project, I also choose to use a custom Bootstrap theme. You can easily find one at **http://bootswatch.com.**

Install SODA.NET

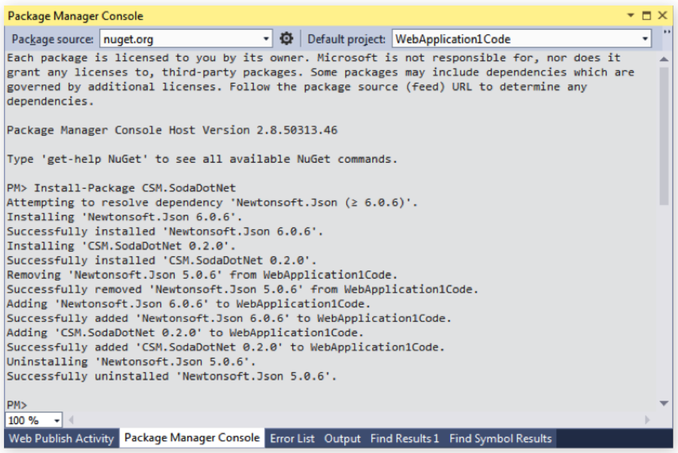
Kegan Maher from the City of Santa Monica created a client library for the Socrata Open Data API at **https://github.com/CityofSantaMonica/SODA.NET**. The rest of this eBook is centered on using this client library for ASP.NET because it does an excellent job of consuming an open data endpoint. Kudos to Kegan and the City of Santa Monica for releasing the full source.

The easiest way to acquire the library into your project is to use the NuGet package manager. Using NuGet you can easily install the dll and reference it for use in your project.

To install the library, open up Visual Studio to your solution. From the Tools menu, select **NuGet Package Manager**, then **Package Manager Console**.

In the **Package Manager Console** window, make sure the Package source is nuget.org and the default project is <Your Solution Name>, and then enter the following command:

Install-Package CSM.SodaDotNet



```
Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it
grant any licenses to, third-party packages. Some packages may include dependencies which are
governed by additional licenses. Follow the package source (feed) URL to determine any
dependencies.

Package Manager Console Host Version 2.8.50313.46

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package CSM.SodaDotNet
Attempting to resolve dependency 'Newtonsoft.Json (≥ 6.0.6)'.
Installing 'Newtonsoft.Json 6.0.6'.
Successfully installed 'Newtonsoft.Json 6.0.6'.
Installing 'CSM.SodaDotNet 0.2.0'.
Successfully installed 'CSM.SodaDotNet 0.2.0'.
Removing 'Newtonsoft.Json 5.0.6' from WebApplication1Code.
Successfully removed 'Newtonsoft.Json 5.0.6' from WebApplication1Code.
Adding 'Newtonsoft.Json 6.0.6' to WebApplication1Code.
Successfully added 'Newtonsoft.Json 6.0.6' to WebApplication1Code.
Adding 'CSM.SodaDotNet 0.2.0' to WebApplication1Code.
Successfully added 'CSM.SodaDotNet 0.2.0' to WebApplication1Code.
Uninstalling 'Newtonsoft.Json 5.0.6'.
Successfully uninstalled 'Newtonsoft.Json 5.0.6'.

PM>
```

The package manager should download and install SODA.NET, plus it's dependencies to the excellent Newtonsoft.Json library.

If you do not have access to NuGet, you can download the project files and/or full source from GitHub and manually reference the code to your project as well.

Getting to Know SodaClient

Once you have the library installed you have should now be able to reference the main SODA.NET classes. The most common namespace you will be using is SODA.

```
using SODA;

var client = new SodaClient();
```

These classes contain the most important object you will be working with, the **SodaClient**. The SodaClient object does the work of making the requests out to the Socrata Open Data API and interacting with the data portals, or endpoints, you choose to use. It handles making the HTTP transactions and moving the payloads to and from your application.

Before we dig into some code samples, let me introduce to you the way you will interact with the data that is available at an open data API endpoint.

Socrata Query Language (SoQL)

The Socrata Query Language (SoQL) is a simple, SQL-like query language specifically designed for ease in working with data on the web. The language is both powerful and easy to learn, and everything works via GET parameters. If you are familiar with SQL syntax you should have no issue working with SoQL. Just remember it cannot, of course, do everything that SQL can do.

The detailed documentation on SoQL is found at: **http://dev.socrata.com/docs/queries.html**

The advantage of using the SODA.NET client library and SodaClient object is that much of the raw SoQL is abstracted out for you as the developer. You can write fluent C# to select the data, order it, and group it.

Retrieving Datasets from API

To retrieve data from a SODA API endpoint you will want to instantiate a **SodaClient** object while passing in a valid resource URL and identifier (4x4 endpoint). Let's walk through how that is done. The full code to do this is found inside of SODAHelper.cs in the sample project that accompanies this e-book.

The Helper

The code snippet below shows the main concept of creating a client and returning data.

```
using PagedList;
using SODA;
using System.Linq;
using WebApplication.Models;

namespace WebApplication.Helpers
{

    public static class SODAHelper
    {
        #region Constants
        /// <summary>
        /// App Token from Socrata registered app - get yours at https://opendata.socrata.com/login
        /// </summary>
        private const string _AppToken = "362kGXXXXXXXXXXXXjkozax";

        /// <summary>
        /// HostName of "http://data.cityofchicago.org/resource/xj6s-q5eb.json";
        /// </summary>
        private const string _APIEndPointHost = "data.cityofchicago.org";

        /// <summary>
        /// Socrata 4x4 Identifier
        /// </summary>
        private const string _APIEndPoint4x4 = "xj6s-q5eb";

        #endregion

        #region Methods

        /// <summary>
        /// Gets sorted list of all Business Locations in the dataset by filter
        /// </summary>
        /// <param name="SearchQuery">Query to filter on</param>
        /// <param name="PageNumber">Current page number</param>
        /// <param name="PageSize">Number of items per page</param>
        /// <param name="OrderBy">Column name to sequence the list by</param>
        /// <param name="OrderByAscDesc">Sort direction</param>
        /// <returns>object PagedList</returns>
        public static PagedList<BusinessLocation> GetBusinessLocations(string SearchQuery, int PageNumber,
int PageSize, string OrderBy, bool OrderByAscDesc)
        {

         //Create client to talk to OpenDat API Endpoint
            var client = new SodaClient(_APIEndPointHost, _AppToken);

            //get a reference to the resource itself the result (a Resouce object) is a generic type
            //the type parameter represents the underlying rows of the resource
            var dataset = client.GetResource <PagedList<BusinessLocation>>(_APIEndPoint4x4);

            //Build the select list of columns for the SoQL call

            string[] columns = new[] { "legal_name", "doing_business_as_name", "date_issued", "city", "state",
"zip_code", "latitude", "longitude"  };
```

CONTINUED ON FOLLOWING PAGE...

CONTINUED FROM PREVIOUS PAGE...

```
            //Column alias must not collide with input column name, i.e. don’t alias ‘city’ as ‘city’
            string[] aliases = new[] { “LegalName”, “DBA”, “IssuedOn” };

            //using SoQL and a fluent query building syntax
            var soql = new SoqlQuery().Select(columns)
                .As(aliases)
                .Order((OrderByAscDesc) ? SoqlOrderDirection.ASC: SoqlOrderDirection.DESC,  new[] { OrderBy });
            if(!string.IsNullOrEmpty(SearchQuery))
            {
                soql = new SoqlQuery().FullTextSearch(SearchQuery);
            }

            var results = dataset.Query<BusinessLocation>(soql);

            //page’em cause there might be quite a few
            PagedList<BusinessLocation> pagedResults = new PagedList<BusinessLocation>(results.ToList(),
PageNumber, PageSize);


            return pagedResults;
        }

    }
}
```

There are a few interesting notes in the code snippet. First, after you create a **SodaClient**, you must make a reference to the resource itself (your API endpoint). This reference is what sets up the REST call for you. This is done in the **client.GetResource** line. Once you have a reference, you can choose to work with it directly, which means all columns and all rows by calling dataset.**GetRows()**, or as the code snippet shows you can choose to reduce the amount of columns by passing in a SoQL query.

Here is fluent C# style syntax to construct a SoQL query that would look like the following:

> ***$select=legal_name AS legalname,doing_business_as_name AS dba,date_issued AS issuedon,city,state,zip_code,latitude,longitude&$order=:id ASC***

Again the power of the SODA.NET library is that query isn’t manually crafted. The .Select, .As(), and .Order() methods are dynamically building it for us.

The Model

In the section above you probably noticed that the dataset is being returned as a PagedList collection of the **BusinessLocation** type. That is the next thing to take note.

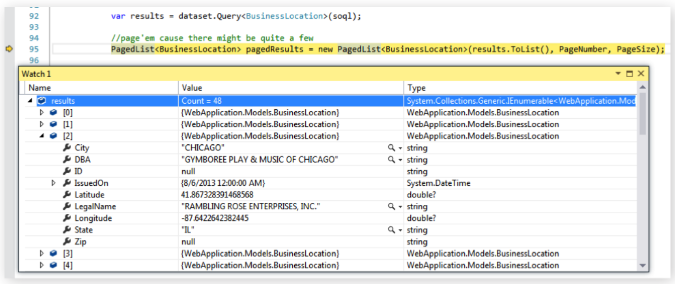
Below is the full code for the BusinessLocation type. Notice how the names of the values in the .Select() method from above line up with the names of the properties of the type, as well as, the values of the aliases in the .As() method.

Again the SODA.NET library handles serializing and de-serializing the JSON that the API endpoint returns. You do not have to spend the time casting and building an enumeration of objects. It is mostly automatic, much like Entity Framework handles building POCO objects from SQL database tables.

```
using System;

namespace WebApplication.Models
{

    public class BusinessLocation
    {

        public string ID { get; set; }
        public string LegalName { get; set; }
        public string DBA { get; set; }
        public string City { get; set; }
        public string State { get; set; }
        public string Zip { get; set; }
        public DateTime IssuedOn { get; set; }
        public double? Latitude { get; set; }
        public double? Longitude { get; set; }
    }
}
```

The Result

After the **GetBusinessLocations** method is done executing you end up with a result that looks like this:



```
            var results = dataset.Query<BusinessLocation>(soql);

            //page'em cause there might be quite a few
            PagedList<BusinessLocation> pagedResults = new PagedList<BusinessLocation>(results.ToList(), PageNumber, PageSize);
```

Watch 1

Name	Value	Type
results	Count = 48	System.Collections.Generic.IEnumerable<WebApplication.Mod
[0]	{WebApplication.Models.BusinessLocation}	WebApplication.Models.BusinessLocation
[1]	{WebApplication.Models.BusinessLocation}	WebApplication.Models.BusinessLocation
[2]	{WebApplication.Models.BusinessLocation}	WebApplication.Models.BusinessLocation
City	"CHICAGO"	string
DBA	"GYMBOREE PLAY & MUSIC OF CHICAGO"	string
ID	null	string
IssuedOn	{8/6/2013 12:00:00 AM}	System.DateTime
Latitude	41.867328391468568	double?
LegalName	"RAMBLING ROSE ENTERPRISES, INC."	string
Longitude	-87.6422642382445	double?
State	"IL"	string
Zip	null	string
[3]	{WebApplication.Models.BusinessLocation}	WebApplication.Models.BusinessLocation
[4]	{WebApplication.Models.BusinessLocation}	WebApplication.Models.BusinessLocation

Working with Datasets

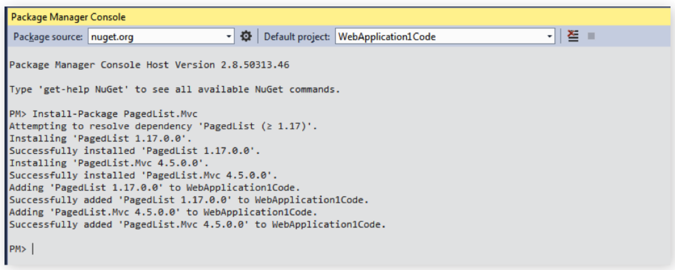
Paging

Because open data datasets usually contain large amounts of rows, showing everything on one web page may be problematic. In the sample application for this eBook I implemented a simple PageList type built for ASP.NET MVC.

To install the **PagedList.MVC NuGet** library, open up Visual Studio to your solution. From the Tools menu, select **NuGet Package Manager** and then **Package Manager Console**.

In the **Package Manager Console** window, make sure the Package source is nuget.org and the Default project is <Your Solution Name>, then enter the following command:

Install-Package PagedList.MVC



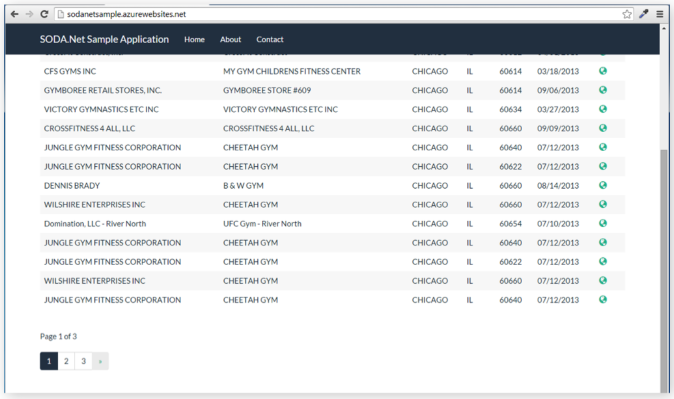
Package source: nuget.org
Default project: WebApplication1Code

```
Package Manager Console Host Version 2.8.50313.46

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package PagedList.Mvc
Attempting to resolve dependency 'PagedList (≥ 1.17)'.
Installing 'PagedList 1.17.0.0'.
Successfully installed 'PagedList 1.17.0.0'.
Installing 'PagedList.Mvc 4.5.0.0'.
Successfully installed 'PagedList.Mvc 4.5.0.0'.
Adding 'PagedList 1.17.0.0' to WebApplication1Code.
Successfully added 'PagedList 1.17.0.0' to WebApplication1Code.
Adding 'PagedList.Mvc 4.5.0.0' to WebApplication1Code.
Successfully added 'PagedList.Mvc 4.5.0.0' to WebApplication1Code.

PM> |
```

Once you have installed the package you simply have to add a using statement to your C# class that returns a PagedList collection, add an @model declaration line to your view that you want to use the pager on, and link up the CSS file that makes the pager look nice.

```
@model PagedList.IPagedList<WebApplication.Models.BusinessLocation>
@using PagedList.Mvc;
<link href="~/Content/Styles/PagedList.css" rel="stylesheet" type="text/css" />
```

When properly implemented, you should see the following:



CFS GYMS INC	MY GYM CHILDRENS FITNESS CENTER	CHICAGO	IL	60614	03/18/2013	●
GYMBOREE RETAIL STORES, INC.	GYMBOREE STORE #609	CHICAGO	IL	60614	09/06/2013	●
VICTORY GYMNASTICS ETC INC	VICTORY GYMNASTICS ETC INC	CHICAGO	IL	60634	03/27/2013	●
CROSSFITNESS 4 ALL, LLC	CROSSFITNESS 4 ALL, LLC	CHICAGO	IL	60660	09/09/2013	●
JUNGLE GYM FITNESS CORPORATION	CHEETAH GYM	CHICAGO	IL	60640	07/12/2013	●
JUNGLE GYM FITNESS CORPORATION	CHEETAH GYM	CHICAGO	IL	60622	07/12/2013	●
DENNIS BRADY	B & W GYM	CHICAGO	IL	60660	08/14/2013	●
WILSHIRE ENTERPRISES INC	CHEETAH GYM	CHICAGO	IL	60660	07/12/2013	●
Domination, LLC - River North	UFC Gym - River North	CHICAGO	IL	60654	07/10/2013	●
JUNGLE GYM FITNESS CORPORATION	CHEETAH GYM	CHICAGO	IL	60640	07/12/2013	●
JUNGLE GYM FITNESS CORPORATION	CHEETAH GYM	CHICAGO	IL	60622	07/12/2013	●
WILSHIRE ENTERPRISES INC	CHEETAH GYM	CHICAGO	IL	60660	07/12/2013	●
JUNGLE GYM FITNESS CORPORATION	CHEETAH GYM	CHICAGO	IL	60640	07/12/2013	●

Page 1 of 3

1 2 3 »

Filtering

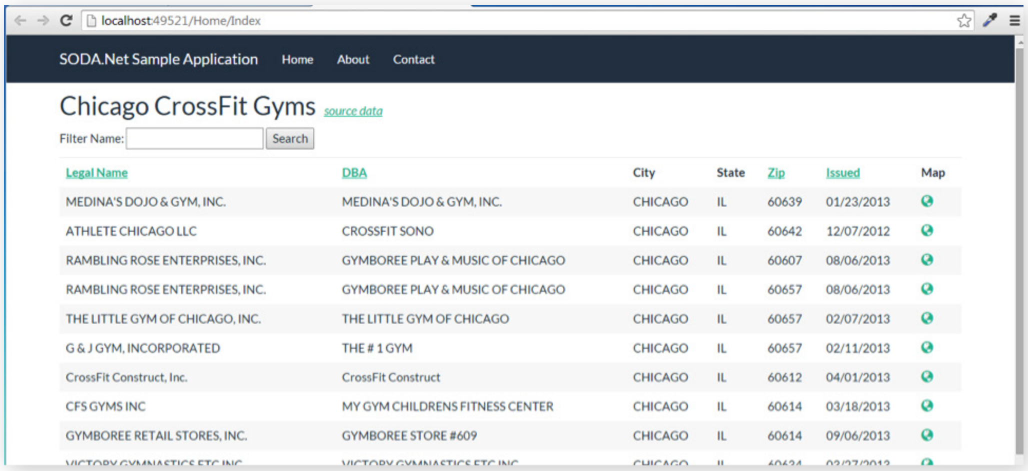
To filter a dataset you have two options. The first option is a **FullTextSearch** method the SodaClient object offers. This method takes your query and translates it into the SoQL syntax of:

*http://data.city.org/resource/xxxx-yyyy.json?**$q=SearchForTermGoesHere***

The full text search will perform a search through all columns in a dataset for the value specified

The second way to filter a dataset is to use the .Where() clause in your fluent C# SoQL call. For full information on the options available, check out the queries documentation at **http://dev.socrata.com/docs/queries.html**.

The sample application for this eBook has the FullTextSearch example built in to it.



localhost:49521/Home/Index

SODA.Net Sample Application Home About Contact

Chicago CrossFit Gyms source data

Filter Name: Search

Legal Name	DBA	City	State	Zip	Issued	Map
MEDINA'S DOJO & GYM, INC.	MEDINA'S DOJO & GYM, INC.	CHICAGO	IL	60639	01/23/2013	●
ATHLETE CHICAGO LLC	CROSSFIT SONO	CHICAGO	IL	60642	12/07/2012	●
RAMBLING ROSE ENTERPRISES, INC.	GYMBOREE PLAY & MUSIC OF CHICAGO	CHICAGO	IL	60607	08/06/2013	●
RAMBLING ROSE ENTERPRISES, INC.	GYMBOREE PLAY & MUSIC OF CHICAGO	CHICAGO	IL	60657	08/06/2013	●
THE LITTLE GYM OF CHICAGO, INC.	THE LITTLE GYM OF CHICAGO	CHICAGO	IL	60657	02/07/2013	●
G & J GYM, INCORPORATED	THE # 1 GYM	CHICAGO	IL	60657	02/11/2013	●
CrossFit Construct, Inc.	CrossFit Construct	CHICAGO	IL	60612	04/01/2013	●
CFS GYMS INC	MY GYM CHILDRENS FITNESS CENTER	CHICAGO	IL	60614	03/18/2013	●
GYMBOREE RETAIL STORES, INC.	GYMBOREE STORE #609	CHICAGO	IL	60614	09/06/2013	●

There are two parts to making the filter work. First the View must have a simple HTML form as seen below:

```
<div class=”row”>
    <div class=”col-md-12”>
        @using (Html.BeginForm(“Index”, “Home”, FormMethod.Get))
        {
            <p>

                Filter Name: @Html.TextBox(“SearchQuery”, ViewBag.CurrentFilter as string)
                <input type=”submit” value=”Search” />
            </p>

        }
    </div>
</div>
```

Secondly, you must call the .FullTextSearch method in your GetBusinessLocations helper also seen below:

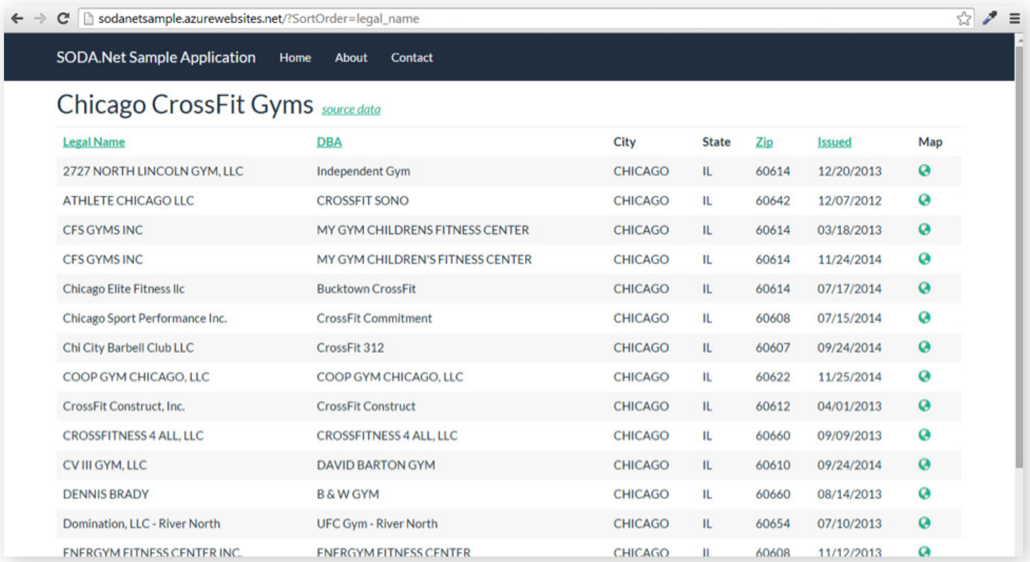
```
//using SoQL and a fluent query building syntax
var soql = new SoqlQuery().Select(columns)

if(!string.IsNullOrEmpty(SearchQuery))
{

    soql = new SoqlQuery().FullTextSearch(SearchQuery);
}

var results = dataset.Query<BusinessLocation>(soql);
```

Author’s note**: At time of publishing this eBook, the City of Chicago’s FullTextSearch method issues a “500 Internal Server Error” for any query that contains a $q= parameter. But other open data portals do correctly filter datasets, such as this one: **http://data.cityofnewyork.us/resource/spgx-ssye.json?$q=Brian

Sorting

In the screenshot below, is the dataset ordered by the Legal Name column instead of the natural sort order. This is handled on the backend of the code through the .Order() clause of the SoQL query.

```
//using SoQL and a fluent query building syntax
var soql = new SoqlQuery().Select(columns)
        .As(aliases)
        .Order(SoqlOrderDirection.ASC,  new[] {
"<OrderByColumnName>" });
```


sodanetsample.azurewebsites.net/?SortOrder=legal_name
SODA.Net Sample Application
Home
About
Contact

Chicago CrossFit Gyms source data

Legal Name	DBA	City	State	Zip	Issued	Map
2727 NORTH LINCOLN GYM, LLC	Independent Gym	CHICAGO	IL	60614	12/20/2013	●
ATHLETE CHICAGO LLC	CROSSFIT SONO	CHICAGO	IL	60642	12/07/2012	●
CFS GYMS INC	MY GYM CHILDRENS FITNESS CENTER	CHICAGO	IL	60614	03/18/2013	●
CFS GYMS INC	MY GYM CHILDREN'S FITNESS CENTER	CHICAGO	IL	60614	11/24/2014	●
Chicago Elite Fitness llc	Bucktown CrossFit	CHICAGO	IL	60614	07/17/2014	●
Chicago Sport Performance Inc.	CrossFit Commitment	CHICAGO	IL	60608	07/15/2014	●
Chi City Barbell Club LLC	CrossFit 312	CHICAGO	IL	60607	09/24/2014	●
COOP GYM CHICAGO, LLC	COOP GYM CHICAGO, LLC	CHICAGO	IL	60622	11/25/2014	●
CrossFit Construct, Inc.	CrossFit Construct	CHICAGO	IL	60612	04/01/2013	●
CROSSFITNESS 4 ALL, LLC	CROSSFITNESS 4 ALL, LLC	CHICAGO	IL	60660	09/09/2013	●
CV III GYM, LLC	DAVID BARTON GYM	CHICAGO	IL	60610	09/24/2014	●
DENNIS BRADY	B & W GYM	CHICAGO	IL	60660	08/14/2013	●
Domination, LLC - River North	UFC Gym - River North	CHICAGO	IL	60654	07/10/2013	●
ENERGYM FITNESS CENTER INC.	ENERGYM FITNESS CENTER	CHICAGO	IL	60608	11/12/2013	●

The Controller

The controller in the sample application is a pretty simple one. The class handles setting up the configuration for paging, sorting, filtering, and calling the helper class of **GetBusinessLocations**. The only noteworthy part of this code is the fact that the SortOrder parameter handles which column the dataset is sorted by, as well as which sort direction is going to be used.

Since the main focus of this eBook is about using the SODA.NET API this code is a good example of how to simply call out to the SODA API and return a PagedList collection, the rest is pretty boilerplate code.

```
using System.Web.Mvc;
using WebApplication.Helpers;
namespace WebApplication.Controllers
{
    public class HomeController : Controller
    {
        public ViewResult Index(string SortOrder, string CurrentFilter, string SearchQuery, int? CurrentPage)
        {
            //Handle Paging
            int pageSize = 20;
            int pageNumber = (CurrentPage ?? 1);

            //Handle Sorting by Column
            if (string.IsNullOrEmpty(SortOrder))
            {
                SortOrder = “”;
            }

            ViewBag.CurrentSort = SortOrder;
            ViewBag.NameSortParm = (SortOrder == “legal_name”) ? “legal_name_desc” : “legal_name”;
            ViewBag.DBASortParm = (SortOrder == “doing_business_as_name”) ? “doing_business_as_name_desc” :
“doing_business_as_name”;
            ViewBag.ZipSortParm = (SortOrder == “zip_code”) ? “zip_code_desc” : “zip_code”;
            ViewBag.IssuedSortParm = (SortOrder == “date_issued”) ? “date_issued_desc” : “date_issued”;

            bool sortAsc = (!SortOrder.Contains(“_desc”)) ? true : false;

            //Handle Filtering
            if (!string.IsNullOrEmpty(SearchQuery))
            {
                CurrentPage = 1;
            }
            else
            {
                SearchQuery = CurrentFilter;
            }

            ViewBag.CurrentFilter = SearchQuery;

            if (!string.IsNullOrEmpty(SearchQuery))
            {
                SearchQuery = SearchQuery.ToUpper().Trim();
            }
```

CONTINUED ON FOLLOWING PAGE...

CONTINUED FROM PREVIOUS PAGE...

```
            //Call out to OpenData API
            var bigData = SODAHelper.GetBusinessLocations(SearchQuery, pageNumber, pageSize, SortOrder.
Replace("_desc", ""), sortAsc);

            return View(bigData);

        }

    }
}
```

The main line, of course, is the **SODAHelper.** **GetBusinessLocations() line of code**. This is where the controller calls the SodaClient.

The View

Once the data is returned to the View, there is little work left to display the table on the web page. Basically, the model's data is bound to a row in a HTML table via standard ASP. NET MVC practices. Note that the correct declarations were applied to make the PagedList collection work correctly at the top of the page then render the PagedListPager html element below the main table thanks to the PagedList.MVC library.

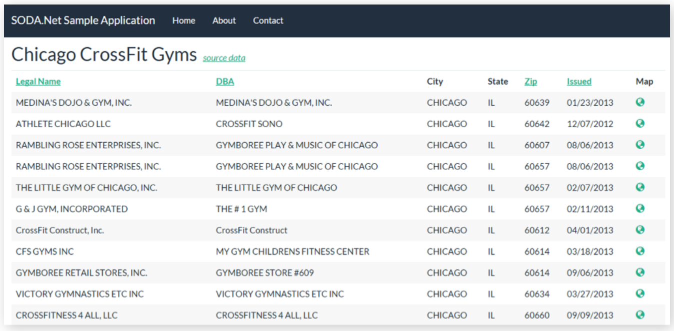
```
@model PagedList.IPagedList<WebApplication.Models.BusinessLocation>
@using PagedList.Mvc;
<link href="~/Content/Styles/PagedList.css" rel="stylesheet" type="text/css" />

@{
    ViewBag.Title = "Home Page";
}

<div class="row">
    <div class="col-md-12">
        <h2>Chicago CrossFit Gyms<small><a href="https://data.cityofchicago.org/Community-Economic-Development/
Chicago-Crossfit-Locations/xj6s-q5eb" target="_blank"><i>source data</i></a></small></h2>
    </div>
</div>


<div class="row">
    <div class="col-md-12">
        <hr class="compact" />
        <table class="table table-striped table-hover ">
            <thead>
                <tr>
                    <th>
                        @Html.ActionLink("Legal Name", "Index", new { SortOrder = ViewBag.NameSortParm,
CurrentFilter = ViewBag.CurrentFilter })
                    </th>
                    <th>
```

CONTINUED ON FOLLOWING PAGE...

CONTINUED FROM PREVIOUS PAGE...

```
                        @Html.ActionLink("DBA", "Index", new { SortOrder = ViewBag.DBASortParm,
CurrentFilter = ViewBag.CurrentFilter })
                    </th>
                    <th>
                        City
                    </th>
                    <th>
                        State
                    </th>
                    <th>
                        @Html.ActionLink("Zip", "Index", new { SortOrder = ViewBag.ZipSortParm,
CurrentFilter = ViewBag.CurrentFilter })
                    </th>
                    <th>
                        @Html.ActionLink("Issued", "Index", new { SortOrder = ViewBag.IssuedSortParm,
CurrentFilter = ViewBag.CurrentFilter })
                     </th>
                    <th>Map</th>
                </tr>
            </thead>
            <tbody>
            @foreach (var item in Model)
            {
                <tr>
                    <td>
                        @Html.DisplayFor(modelItem => item.LegalName)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.DBA)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.City)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.State)
                    </td>
                    <td>
                        @Html.DisplayFor(modelItem => item.Zip)
                    </td>
                    <td>
                        @String.Format("{0:MM/dd/yyyy}", item.IssuedOn)
                    </td>
                    <td>
                        @if (item.Latitude != null)
                        {
                            <a href="https://google.com/maps/place/@item.Latitude,@item.Longitude"
target="_blank"><span class="glyphicon glyphicon-globe"></span></a>
                        }
                    </td>
                </tr>
            }
            </tbody>
        </table>

        <br />
        Page @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) of @Model.PageCount

        @Html.PagedListPager(Model, CurrentPage => Url.Action("Index", new { CurrentPage, SortOrder = ViewBag.
CurrentSort, CurrentFilter = ViewBag.CurrentFilter }))

    </div>
</div>
```

After correctly adding all of the parts of the MVC structure you should now have a working page that displays the data from the API endpoint.



SODA.Net Sample Application   Home   About   Contact

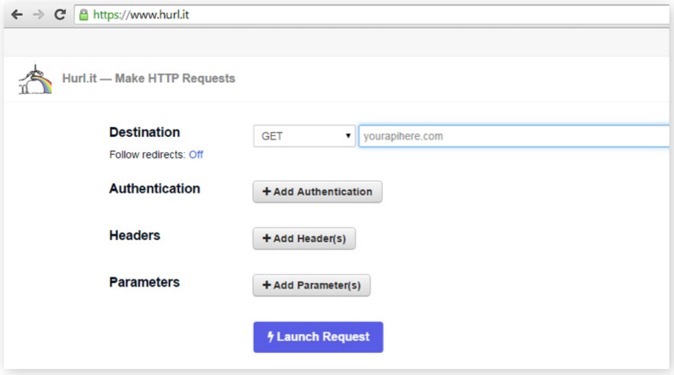
Chicago CrossFit Gyms source data

Legal Name	DBA	City	State	Zip	Issued	Map
MEDINA'S DOJO & GYM, INC.	MEDINA'S DOJO & GYM, INC.	CHICAGO	IL	60639	01/23/2013	●
ATHLETE CHICAGO LLC	CROSSFIT SONO	CHICAGO	IL	60642	12/07/2012	●
RAMBLING ROSE ENTERPRISES, INC.	GYMBOREE PLAY & MUSIC OF CHICAGO	CHICAGO	IL	60607	08/06/2013	●
RAMBLING ROSE ENTERPRISES, INC.	GYMBOREE PLAY & MUSIC OF CHICAGO	CHICAGO	IL	60657	08/06/2013	●
THE LITTLE GYM OF CHICAGO, INC.	THE LITTLE GYM OF CHICAGO	CHICAGO	IL	60657	02/07/2013	●
G & J GYM, INCORPORATED	THE # 1 GYM	CHICAGO	IL	60657	02/11/2013	●
CrossFit Construct, Inc.	CrossFit Construct	CHICAGO	IL	60612	04/01/2013	●
CFS GYMS INC	MY GYM CHILDRENS FITNESS CENTER	CHICAGO	IL	60614	03/18/2013	●
GYMBOREE RETAIL STORES, INC.	GYMBOREE STORE #609	CHICAGO	IL	60614	09/06/2013	●
VICTORY GYMNASTICS ETC INC	VICTORY GYMNASTICS ETC INC	CHICAGO	IL	60634	03/27/2013	●
CROSSFITNESS 4 ALL, LLC	CROSSFITNESS 4 ALL, LLC	CHICAGO	IL	60660	09/09/2013	●

Troubleshooting

As always, the Visual Studio debugger is your friend when it comes to issues with your ASP.NET code, but sometimes it is nice to have other go-to resources. As mentioned, there is documentation on the Socrata Developers site. The SODA.NET GitHub repository has a list of issues that you can review. Contacting Socrata is also a possibility.

There is one more tool worth noting that I really found helpful when first learning SODA and SoQL, and that is **http://hurl.it**. Hurl.it is like a live version of the popular utility curl. The tool lets you execute HTTP requests and has a simple interface to add in parameters.

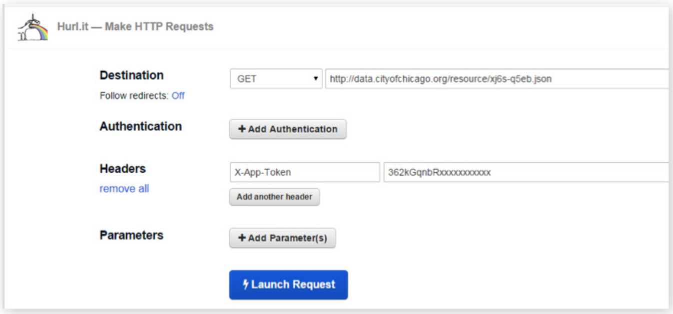


https://www.hurl.it

Hurl.it — Make HTTP Requests

Destination: GET yourapihere.com
Follow redirects: Off
Authentication: + Add Authentication
Headers: + Add Header(s)
Parameters: + Add Parameter(s)
Launch Request

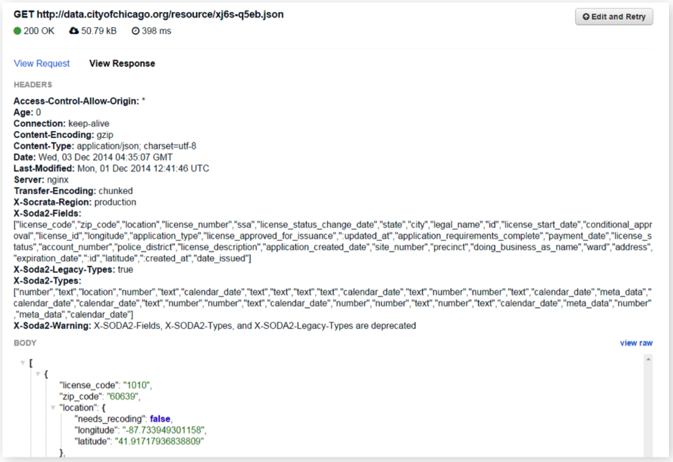
I frequently used this tool to test out my queries to my API endpoint. A sample call is below:



Hurl.it — Make HTTP Requests

Destination: GET http://data.cityofchicago.org/resource/xj6s-q5eb.json
Follow redirects: Off
Authentication: + Add Authentication
Headers: X-App-Token 362kGqnbRxxxxxxxxxxx
remove all
Add another header
Parameters: + Add Parameter(s)
Launch Request

Clicking on **Launch Request** gave me the results (and errors) that my ASP.NET code was finding:



```
GET http://data.cityofchicago.org/resource/xj6s-q5eb.json
200 OK   50.79 kB   398 ms

View Request   View Response

HEADERS
Access-Control-Allow-Origin: *
Age: 0
Connection: keep-alive
Content-Encoding: gzip
Content-Type: application/json; charset=utf-8
Date: Wed, 03 Dec 2014 04:35:07 GMT
Last-Modified: Mon, 01 Dec 2014 12:41:46 UTC
Server: nginx
Transfer-Encoding: chunked
X-Socrata-Region: production
X-Soda2-Fields:
["license_code","zip_code","location","license_number","ssa","license_status_change_date","state","city","legal_name","id","license_start_date","conditional_appr
oval","license_id","longitude","application_type","license_approved_for_issuance",":updated_at","application_requirements_complete","payment_date","license_s
tatus","account_number","police_district","license_description","application_created_date","site_number","precinct","doing_business_as_name","ward","address",
"expiration_date",":id","latitude",":created_at","date_issued"]
X-Soda2-Legacy-Types: true
X-Soda2-Types:
["number","text","location","number","text","calendar_date","text","text","text","text","calendar_date","text","number","number","text","calendar_date","meta_data","
calendar_date","calendar_date","text","number","number","text","calendar_date","number","number","text","number","text","calendar_date","meta_data","number"
,"meta_data","calendar_date"]
X-Soda2-Warning: X-SODA2-Fields, X-SODA2-Types, and X-SODA2-Legacy-Types are deprecated

BODY                                                                          view raw
[
  {
    "license_code": "1010",
    "zip_code": "60639",
    "location": {
      "needs_recoding": false,
      "longitude": "-87.733949301158",
      "latitude": "41.91717936838809"
    },
```

API Limits

Another note worth mentioning is Socrata and the Socrata Open Data portal does have a set of API limits. You can make a certain number of requests without an application token, but they come from a shared pool and you're eventually going to be cut off. When you include an App Token that limit is raised.

*The **API Limit** is **1000 per hour** when using **an App Token***

What's Next

This eBook has shown you how to use the Socrata Open Data API with ASP.NET MVC from a data consumption standpoint. The API does also allow for creating new datasets, adding items into datasets, and updating items inside of datasets. For more on how to accomplish these advanced tasks, please see the Socrata Developers site at: **http://dev.socrata.com/publishers/getting-started.html**

Summary

You learned a lot in this eBook about programming ASP.NET MVC and C# with consuming the Socrata Open Data API using a great open source library SODA.NET. Specifically covered:

- What is SODA
- How to access a SODA dataset
- How to setup a Socrata developer account
- How to install SODA.NET
- How to consume an API endpoint via SODA.NET in ASP.NET MVC
- Viewing, sorting, paging, and filtering a dataset

A big thank you goes out to the excellent team at Socrata for providing a strong platform to work on, and an equally big thanks goes out to the City of Santa Monica for providing the excellent SODA.NET C# client library.