TowerData
December 31, 2017

C#
The C# SDK contains two solutions: example and personalization. The example solution provides several usage examples of the API. The personalization solution is used to compile a DLL containing the TowerData C# API. A pre-compiled DLL is provided that should work with most modern CPUs and Windows operating systems.

For sample usage of direct and bulk APIs, see the example in our GitHub C# SDK. It uses the <code>JavaScriptSerializer</code> class which is available in .NET Frameworks 3.5 and higher in the <code>system.web.extensions</code> element.

> Example Code Path

```
cd <path-to-SDK>\csharp\example\example\bin
```

> Usage Syntax

```
example.exe <your API key> [sample email 1] [sample email 2] [sample email 3]
```

Usage

To run the sample solution open a command prompt and <code>cd</code> into the bin directory of the example. The only required parameter is a valid API key. Optional parameters allow up to three sample email addresses. When you <code>Hit ENTER</code> a series of queries run. See the example code path and usage syntax in the right column.

The first query uses sample email 1 (or the default if not entered) and performs a query by email only. The second query is a bulk query containing an email and NAP. If entered, sample email 1 will be used in this query. The third queries by email iteratively using three different email addresses. Your sample email 1, 2 and 3 will be used for this query. After each query, the results are returned to the console. You'll be prompted to <code>Hit ENTER</code> to proceed to the next query.

Recompiling

A C# compiler is required if recompilation of either solution is necessary. The TowerData C# SDK was compiled with Microsoft Visual Studio Community 2013. Begin at the Windows Start menu and select:

<code> All Programs > Visual Studio 2013 > Visual Studio Tools</code>

The Visual Studio Common Tools folder will open in Windows Explorer. Double-click the shortcut:

<code>Developer Command Prompt for VS2013</code>

In the open command prompt <code>cd</code> to the directory containing the solution you want to recompile. See the example below:

> Clean & Rebuild Command

```
msbuild /p:Configuration=Release /p:Platform="Any CPU" /target:Rebuild
```

<code>cd <path-to-SDK>\csharp\personalization</code>

To clean and rebuild the solution use the command in the right column.

The C# SDK contains two solutions: personalization and example. Personalization is used to compile a DLL containing the TowerData C# API, while example provides several usage examples of the API. A pre-compiled DLL is provided that should work with most modern CPUs and Windows operating systems. For sample usage of direct and bulk APIs, see csharp\example\example\TowerDataExample.cs. The SDK uses the class JavaScriptSerializer, which is available in .NET Frameworks 3.5 and higher, in the System.Web.Extensions assembly.

Usage

To run the sample solution, open a command prompt and cd to the bin directory of example. For instance,

cd <path-to-SDK>\csharp\example\example\bin

The usage syntax is as follows

example.exe <your API key> [sample email 1] [sample email 2] [sample email 3]

The only required parameter is a valid API key. You can also optionally enter up to three sample emails. When you hit enter, a series of queries will be run. The first query will use sample email

1 (or the default if not entered) and perform a query by email only. The second query is a bulk query that contains an email and NAP. If entered, sample email 1 will be used in this query. The third query is querying by email iteratively using three different emails. If entered, your sample email 1, 2 and 3 will be used for this query. After each query, the results are returned to the console. You are prompted to "Hit Enter" to proceed to the next query.

Recompiling

A C# compiler is required if recompilation of either solution is necessary. This SDK was compiled with Microsoft Visual Studio Community 2013. If using Visual Studio Community 2013, from the Windows Start menu select

```
All Programs -> Visual Studio 2013 -> Visual Studio Tools
```

This will open the Visual Studio Common Tools folder in Windows Explorer. Double click the shortcut

```
Developer Command Prompt for VS2013
```

In the command prompt that opens, cd to the directory containing the solution you want to recompile. For example,

```
cd <path-to-SDK>\csharp\personalization
```

Clean and rebuild the solution with the following command

```
msbuild /p:Configuration=Release /p:Platform="Any CPU" /target:Rebuild
```