QQ CATALYST API TEST HARNESS

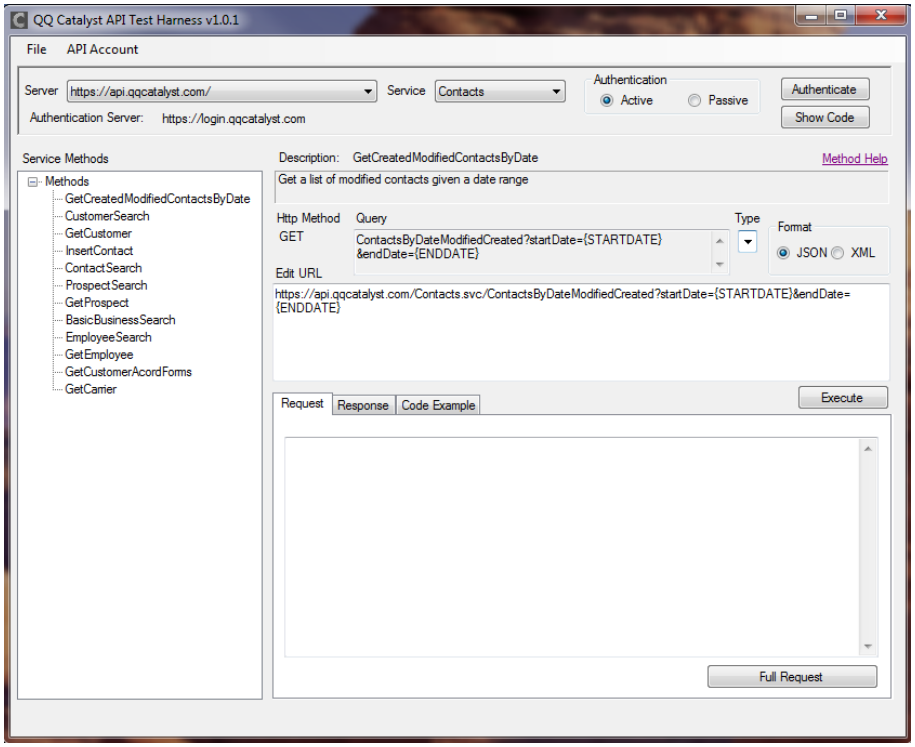
WELCOME!

The QQ Catalyst API Test Harness is an application created by the API development team at QQ Solutions, Inc. The sole purpose of this application is to aid our partners in development as you navigate our API services. This application will walk you through the authentication processes, interacting with the API services, and provide you with code snippets every step of the way. We are committed to having an easily consumable and highly usable API. If during the process of developing your integration, you have suggestions for improving the API, test harnesses, or any of the supporting documentation, please email us your feedback. We would love to hear from you!

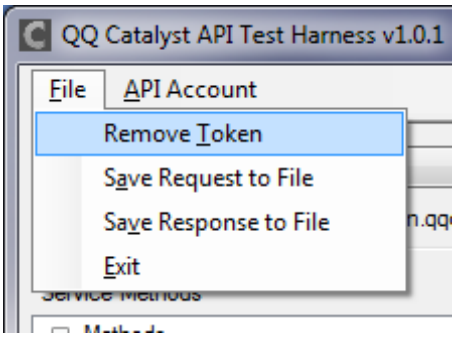
NAVIGATING THE TEST HARNESS

In order to provide you with more insight into our API, we have built the test harness to walk you through the details of every interaction available through the API. The main screen will provide visibility to every level of detail for each method request and response.

MAIN SCREEN



FILE



REMOVE TOKEN

This operation will remove the authentication token from your local file system. You should remove your token to test your authentication scripts once you have created them to ensure a clean authentication will succeed. This action also is required if you would like to change the credentials with which you have previously authenticated in Catalyst.

SAVE REQUEST TO FILE

This menu item allows you to save a request stream to a text file document. This is very useful for building customized messages for testing your integration with the Catalyst API. You can create individual requests and save them for future use.

SAVE RESPONSE TO FILE

This menu item allows you to save a response stream to a text file document. This is very useful for building test scripts for the result of your integration with the Catalyst API. Individual response files can be saved and compared during testing to ensure consistency in response.

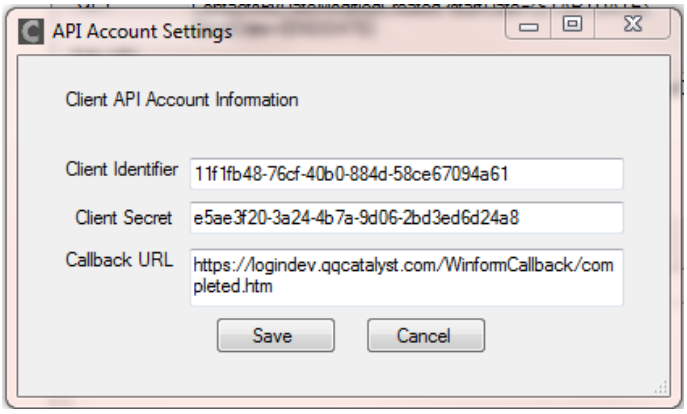
API ACCOUNT

CATALYST API WEBSITE

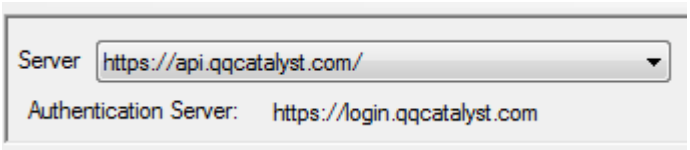
This link will forward you to our API page on GitHub. This page contains the most recent version of the API Test Harness and all associated documentation.

SETUP API ACCOUNT

After registering with QQ Solutions, Inc as a Catalyst API partner, you will receive a client identifier and a client secret. These tokens are provided to you from QQ Solutions, Inc. The information you provide us as part of registration will include the callback URL used during the authentication routine. If you do not have a callback URL, we can provide you with a dummy URL to move forward with development of your integration. The purpose of this callback URL is explained further in the authentication section of this document.

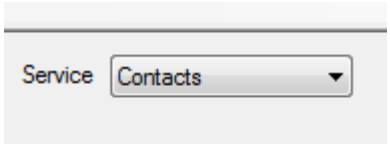


SERVER



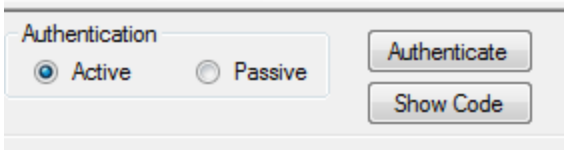
The server listing shows the location for both the API server and the Authentication server. The Authentication server will not change and is static for all environments. The API server is generally static and will only have other choices in very specific circumstances.

SERVICE



The service list will show the categories of data available via the API for consumption. This is the list of services currently available within the Catalyst API. This list is populated dynamically from our API, and recognizes changes in real-time.

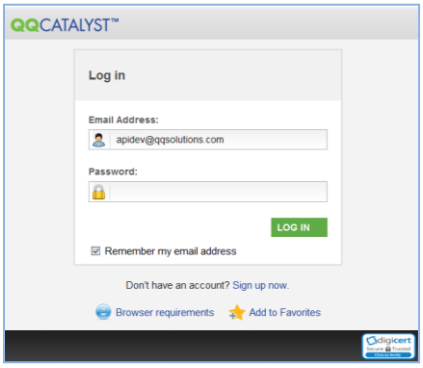
AUTHENTICATION



Our API allows for both active and passive authentication of users. Routines for authentication are very similar for each method, but require minor differentiation.

ACTIVE

An active authentication means that your users will have to visually type in a user name and password prior to connecting your application to the Catalyst API. We recommend this approach for any integrating application that presents a UI to the user. This will only be required when the user has no token saved for the application. This could be caused by any of the following reasons: they have never logged in with your application, they have deleted their authentication token, or they have not selected 'Remember my email address' on the authentication screen. When a user is actively authenticating, the following screen will be presented to them:



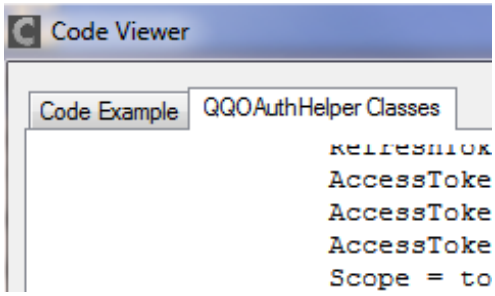
350 Fairway Drive, Suite 101 • Deerfield Beach, FL 33441 • p. 800.940.6600 • f. 800.940.3299 • www.QQSolutions.com

PASSIVE

A passive authentication allows your application to authenticate non-visually to the API. We recommend this approach for any integrating application that never presents a UI to the user (e.g. Web services, scheduled runs, etc). In order to leverage the passive authentication, your application will have to store identifiers that allow complete identification against the API. These identifiers will be passed to Catalyst API as part of the authentication routine.

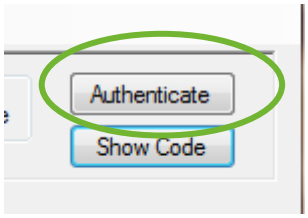
QQOAUTHHELPER CLASSES

In order to help you authenticate quickly and easily against the Catalyst API, we have included code for classes that will wrap up some of the logic and make it more easily consumed by your application. When you click the 'Show Code' button, you will see a second tab with the code for these helper classes.



Feel free to copy the code as-is and place in your application. The samples illustrated in our application leverage the same helpers.

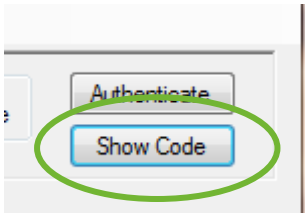
AUTHENTICATE



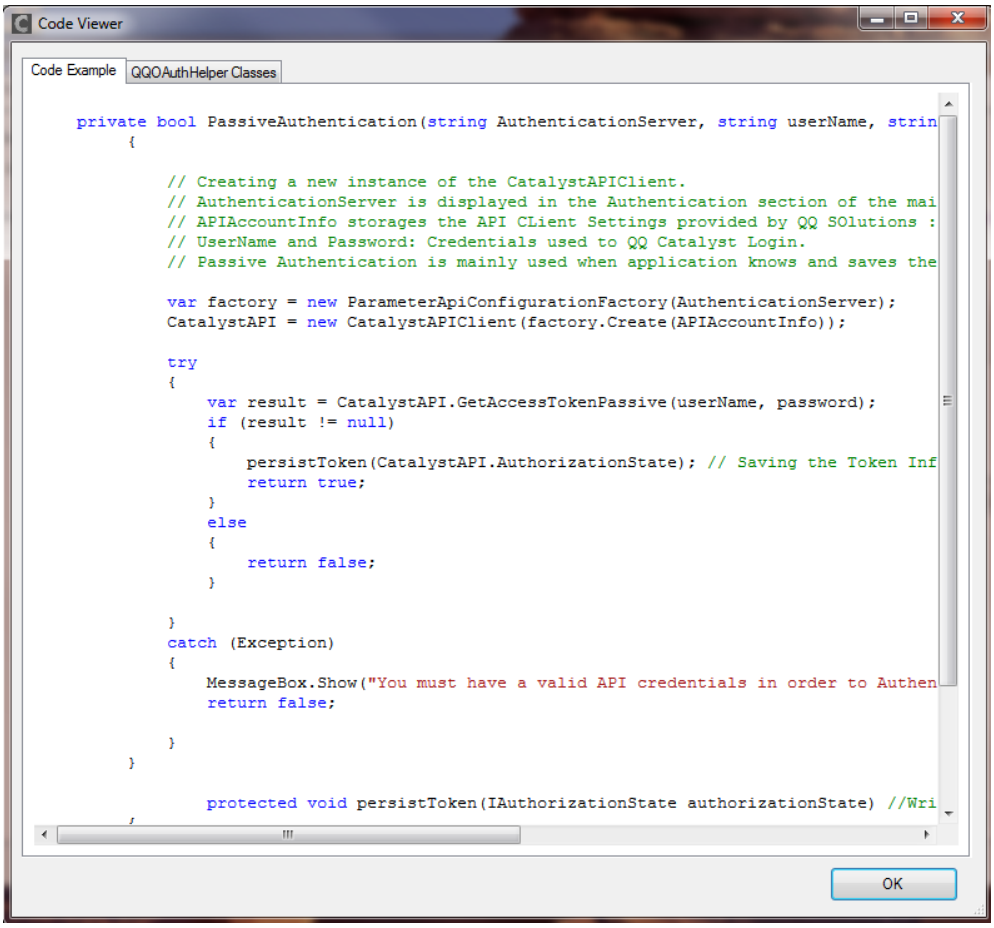
Clicking this button will authenticate the sample application against the live Catalyst API. This allows you to begin working with the various http requests and checking the responses. You must authenticate against Catalyst prior to making any requests. If you are not authenticated, the services will return an http error 401.

350 Fairway Drive, Suite 101 • Deerfield Beach, FL 33441 • p. 800.940.6600 • f. 800.940.3299 • www.QQSolutions.com

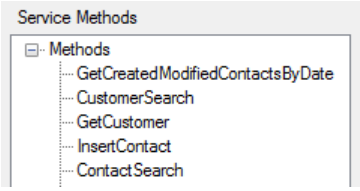
SHOW CODE



Clicking this button will display source code illustrating the authentication routines you have selected (passive or active). We encourage you to copy this code as-is and use it within your application as a basis for your authentication routines.

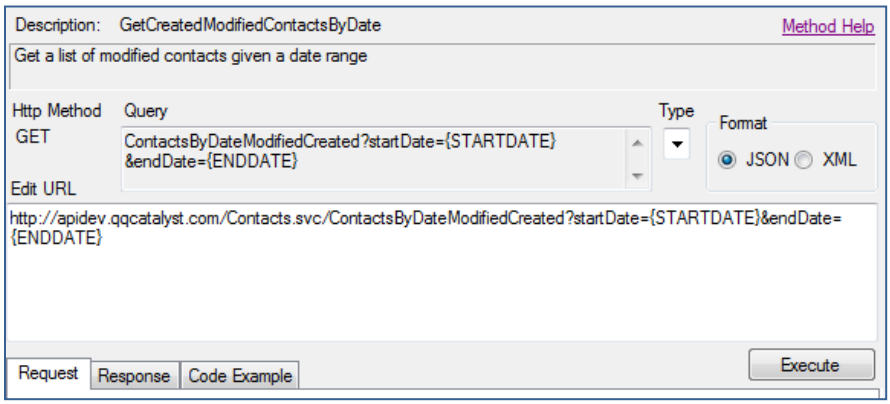


SERVICE METHODS



The service methods list reflects the individual methods available for the category selected in the 'Service' list. This navigable list will populate the details of the service method properties based on the selected method from this list. This list is populated dynamically from our API, and recognizes changes in real-time.

METHOD PROPERTIES



As you select a method from the list, the properties will display all of the associated information pertaining specifically to the selected method. This allows you to have a working default request, while allowing you to customize the content and test your own data calls.

DESCRIPTION

The description space will display both the name and summary of the method. Clicking the 'Method Help' link will take you to the live service help page for the selected method.

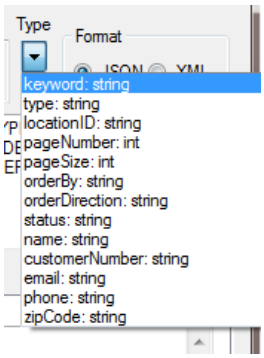
HTTP METHOD

The HTTP method area will show you the details of the request expected by the API. The method will show you the verb that indicates the type of request being made. We use *GET*, *PUT*, *POST*, and *DELETE* methods throughout the API.

QUERY

The query area will display the query string of the method. For example, *Customer?keyword={KEYWORD}&type={TYPE}&locationID={LOCATIONID}&pageNumber={PAGENUMBER}&pageSize={PAGESIZE}&orderBy={ORDERBY}&orderDirection={ORDERDIRECTION}&status={STATUS}&name={NAME}&customerNumber={CUSTOMERNUMBER}&email={EMAIL}&phone={PHONE}&zipCode={ZIPCODE}* will be the query string for the CustomerSearch method at time of document creation. This allows you to focus solely on the parameters expected by the method without having to read the entire Uri.

TYPE



The type list will expand to display the list of parameters in the query of the selected method. As part of the display, the list will indicate the expected data type for each parameter. This will aid you in creating the proper query with type safe data.

FORMAT

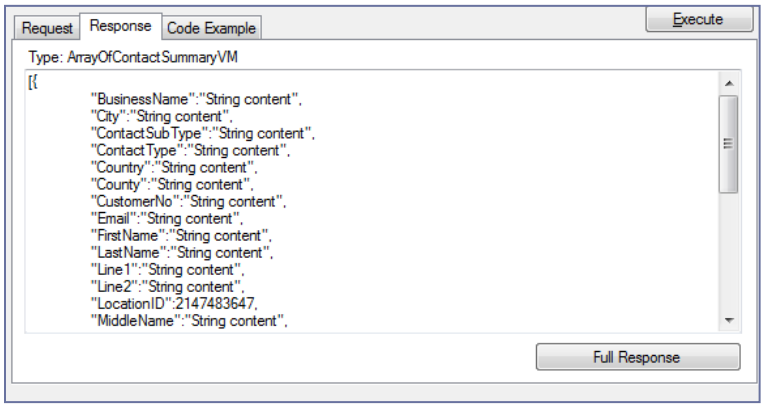
Our API is built to be platform agnostic and enable our partners to develop with a wide range of tools. Based on the selection of the format option, our API will accept and respond to your application in the way best suited to your needs. Our service will accept both XML and JSON data as a payload. In addition, when setting the proper expectation in your GET calls, our API will respond to you in the format you desire as well. This enables you to choose the best communication contract for your application.

EDIT URL

You can build and send live requests by changing the values within the Edit area. The request will default to the sample schema of the associated method, but is fully editable for your testing requirements. Clicking the 'Execute' button will send the request from the edit area to the live API server.

350 Fairway Drive, Suite 101 • Deerfield Beach, FL 33441 • p. 800.940.6600 • f. 800.940.3299 • www.QQSolutions.com

REQUEST / RESPONSE / CODE EXAMPLE



The Request/Response/Code Example area is intended to display all of the details associated method call. We will provide you with a full sample of the request, the response and a C# code sample showing you exactly how to call this method.

REQUEST

When the request requires more than just a query, the Request tab will show the details of the expected. This will also reflect your choice of communication format. This will allow you to build a request for testing the insert and update methods of the API from this sample application. You can also easily copy and paste this request body into you IDE and begin populating from your application. Once you have the request built to your requirements, select 'Execute' to send the request to the live API service. You can also save the full request for later use via the 'File' menu.

FULL REQUEST

Choosing the full request button, a dialog will open allowing you to see the entire request in one larger edit control. You can make changes in this window to suit your testing requirements. By selecting 'Ok', your changes will be reflected in the request tab so you can 'Execute' from there.

RESPONSE

The response tab will show you the return value you can expect from the selected method call. We will indicate the defined contract type as well as the individual values and data types that you can expect. You can also save the full response for later use via the 'File' menu.

350 Fairway Drive, Suite 101 • Deerfield Beach, FL 33441 • p. 800.940.6600 • f. 800.940.3299 • www.QQSolutions.com

FULL RESPONSE

Choosing the full response button, a dialog will open allowing you to see the entire response in one larger edit control. You can make changes in this window to suit your testing requirements. By selecting 'Ok', your changes will be reflected in the response tab.

CODE EXAMPLE

The code example tab will give you a full C# example of building the request and handling the response for the selected method. You are welcome to copy and paste this code directly into your application to enable quick development to our API.

FULL CODE EXAMPLE

Choosing the full code example button, a dialog will open allowing you to see the entire snippet in one larger edit control. This dialog will also display the *QQOAuthHelper* classes for use during authentication.

AUTHENTICATION

The authentication framework we use is the industry standard OAuth2 protocol. As a developer, you must contact QQ Solutions for access privileges. By giving us a little bit of information about yourself and your intentions to use the API, we will in return give you an Access Token and Access Key. With these two pieces of information, you will be able to prompt the user to login with our login page and access user data. If you are not familiar with OAuth2 standards or processes, please proceed to the OAuth 2.0 Standards page to learn more about the technology used.

Data security is our number one concern. Users can be confident that their data is secure and only authorized access will be permitted. Our API login process will always require a user to log in through the QQ Catalyst secure web channels. By using OAuth2 standards and active authentication, no third-party developer can ever gather users' login credentials. Instead, they will redirect QQ Catalyst users to our login page.

If you are developing in Visual Studio, we highly recommend that you use the tools provided in the DotNetOpenAuth libraries, which can be found here or in NuGet inside Visual Studio. OAuth2 requires that you do several things before you can access user data. After you have successfully signed up with QQ Solutions for access to the API, and we have received the required information from you including a valid callback URL to send the user custom token back to you, you can begin to call the OAuth Login Server. The two URLs that you will need to access the login server are:

- Authorize: https://login.qqcatalyst.com/oauth/authorize
- Token: https://login.qqcatalyst.com/oauth/token

QQSOLUTIONS™
we provide the answers

TECHNOLOGY

SERVICES

Our service methods are as RESTful web services. We use the WCF platform to ensure proper security, elasticity, availability and portability of our API. Our API service is hosted on the Microsoft Azure™ platform which provides a guaranteed uptime reflected in the Windows Azure Cloud Services, Virtual Machines, and Virtual Network SLA. We are committed to the stability and performance of our API. If you have any questions or feedback regarding our technology platform, send us an email.

SAMPLE APPLICATION

Our sample application is written using C# and .net 4.0. We used a Win Forms application to present all of the code to you visually and easily. We do not provide the source code to the sample application itself, as it is simply a means to display the code you are after. All of the code snippets displayed are available for your use as you create your integrating application.

THANK YOU!

Thank you for your interest in becoming an API partner with QQ Solutions, Inc. We are committed to serving our customers and our partners by creating the best and most reliable platforms in the industry. We are interested in hearing about your experiences with our API. If you have feedback, please send us an email at apidev@qqsolutions.com.