**SOAP-API Web Service Bronto Integration with C#**

On GitHub: <https://github.com/tlilov>

Bronto’s API was built on the SOAP web service. Here you will find how to properly integrate the API functions and objects to deliver extensible and testable solution in C#. <http://dev.bronto.com/category/api/soap/>

**Prerequisites**

Create Service Reference to the Bronto API using the bronto WSDL. <https://api.bronto.com/v4?wsdl>

**Run**

Step 1. Create Interface IBrontoApiWrapper.cs around the methods from the API that we need.

public interface IBrontoApiWrapper {

writeResult AddContacts(EmailSystem.com.bronto.api.contactObject[] contacts);

writeResult AddDeliveries(EmailSystem.com.bronto.api.deliveryObject[] deliveries);

-------

writeResult UpdateContacts(EmailSystem.com.bronto.api.contactObject[] contacts);

}

Step 2. Implement the intereface from step 1, in BrontoApiWrapper.cs

public class BrontoApiWrapper : IBrontoApiWrapper {

public writeResult AddContacts(contactObject[] contacts) {

return api.addContacts(contacts);

}

Step 3. Create a service class BrontoEmailService.cs to expose the methods from step 2.

public string AddContact(Contact contact) {

-----

List<contactObject> contacts = new List<contactObject>();

contacts.Add(translator.TranslateContact(contact, GetContactFields()));

var result = wrapper.AddContacts(contacts.ToArray());

-----}

Step 4. In the service class use translator helper methods to map and convert the specific client objects to the corresponding API objects needed by the API (example contacts, messages, tokens, orderData etc)

contacts.Add(translator.TranslateContact(contact, GetContactFields()));

public contactObject TranslateContact(Contact contact, Dictionary<string, string> ContactFields = null) {

foreach (KeyValuePair<string, string> kvp in ContactFields) {

Step 5. In the service class provide, logging and error handling around each API method call.

Logger.Debug("AddContact(): Email " + contact.Email);

Logger.Error("Call to AddContact() when contact was null.");

|  |
| --- |
|  |

Step 6. Use a client (web app, ajax, windows app etc) to call the service class methods that expose the API.