Add a contact to a contact list - AddContactToContactList

```
<Envelope>
    <Body>
        <AddContactToContactList>
            <COLUMN>
                <NAME>Email</NAME>
                <VALUE>jmalone30@gmail.com</VALUE>
            </COLUMN>
            <CONTACT_LIST_ID>7835197</CONTACT_LIST_ID>
        </AddContactToContactList>
        <AddContactToContactList>
            <COLUMN>
                <NAME>Email</NAME>
                <VALUE>jmalone31@gmail.com</VALUE>
            </COLUMN>
            <CONTACT_LIST_ID>7835198</CONTACT_LIST_ID>
        </AddContactToContactList>
        <AddContactToContactList>
            <COLUMN>
                <NAME>Email</NAME>
                <VALUE>jmalone32@gmail.com</VALUE>
            </COLUMN>
            <CONTACT_LIST_ID>7835199</CONTACT_LIST_ID>
        </AddContactToContactList>
    </Body>
</Envelope>
```

Using the above request, I will get back a response object with 3 results.

```
<Envelope>
    <Body>
        <RESULT>
            <SUCCESS>true</SUCCESS>
        </RESULT>
        <RESULT>
            <SUCCESS>false</SUCCESS>
        </RESULT>
        <Fault>
            <Request/>
            <FaultCode/>
            <FaultString><![CDATA[Recipient Not Found in List]]></FaultString>
            <detail>
                <error>
                    <errorid>1008</errorid>
                    <module/>
                    <class>SP.API</class>
                    <method/>
                </error>
            </detail>
        </Fault>
        <RESULT>
            <SUCCESS>false</SUCCESS>
        </RESULT>
        <Fault>
            <Request/>
            <FaultCode/>
            <FaultString><![CDATA[Recipient Not Found in List]]></FaultString>
            <detail>
                <error>
                    <errorid>1008</errorid>
                    <module/>
                    <class>SP.API</class>
                    <method/>
                </error>
            </detail>
        </Fault>
    </Body>
</Envelope>
```

Notice that the first result was successful, so there is no Fault object. The next 2 results failed, so each is followed by a Fault object. There are 3 issues that I believe would much improve the developer experience.

1) Move the Fault object inside the Result object. This will allow the developer to easily associate the FaultString message with a specific Result.
2) Add some identifying data to the Result object, whether it is successful or not. This will allow the developer to associate the result and fault with the appropriate request object.

The following is an example of what the enhanced response object might look like:

```
<Envelope>
    <Body>
        <RESULT>
            <SUCCESS>true</SUCCESS>
            <CONTACT_LIST_ID>7835197</CONTACT_LIST_ID>
            <COLUMN>
                <NAME>Email</NAME>
                <VALUE>jmalone30@gmail.com</VALUE>
            </COLUMN>
            <Fault/>
        </RESULT>
        <RESULT>
            <SUCCESS>false</SUCCESS>
            <CONTACT_LIST_ID>7835198</CONTACT_LIST_ID>
            <COLUMN>
                <NAME>Email</NAME>
                <VALUE>jmalone31@gmail.com</VALUE>
            </COLUMN>
            <Fault>
                <Request/>
                <FaultCode/>
                <FaultString><![CDATA[Recipient Not Found in
List]]></FaultString>
                <detail>
                    <error>
                        <errorid>1008</errorid>
                        <module/>
                        <class>SP.API</class>
                        <method/>
                    </error>
                </detail>
            </Fault>
        </RESULT>

        <RESULT>
            <SUCCESS>false</SUCCESS>
            <CONTACT_LIST_ID>7835199</CONTACT_LIST_ID>
            <COLUMN>
                <NAME>Email</NAME>
```

```
                <VALUE>jmalone32@gmail.com</VALUE>
            </COLUMN>
            <Fault>
                <Request/>
                <FaultCode/>
                <FaultString><![CDATA[Recipient Not Found in
List]]></FaultString>
                <detail>
                    <error>
                        <errorid>1008</errorid>
                        <module/>
                        <class>SP.API</class>
                        <method/>
                    </error>
                </detail>
            </Fault>
        </RESULT>
    </Body>
</Envelope>
```

This is just an example, but as you can see, it is now much easier to associate the results with the original requests. Maybe requiring the developer to submit a random unique identifier to each contact in the request that you could just spit back in the response would be an easy way to associate them.

As previously mentioned, this would really ease the burden on the developer using this API. Being able to consolidate multiple contacts into a single request increases performance by very measurable gains.

I am sure this same strategy could be implemented with other API calls to achieve similar benefits.

Another thing I noticed (with other API endpoints) is that sometimes the response object will contain a Success element with an uppercase TRUE/FALSE and sometimes lowercase. Uppercase results in having to deserialize the resulting Success element into a string, then providing a getter that casts the string to a Boolean. This represents another development pain that could be alleviated by always returning lowercase, as lowercase true/false is automatically deserialized to a Boolean (at least this is true in C#). At least be consistent!