**CyberImpact - Web service – v1.0.0**

# General information on webservice

The webservice will have two methods. The first will be used to update account data and the second to update contact data. Most validation will be done using plugins in CRM.

There is also three plugin to develop in one assembly.

# Web service config file & Code structure

The web service must have a parameter file that allows us to select the URL of the server. And we must be able to use either windows authentication or select the username/pw for online and IFD installations.

We are providing a project from another customer (TireLink) that works as we want it as a template. Simply replace all reference to TireLink with CyberImpact and start from there.

# Method UpdateCustomer

Parameters:

customerXML (System.Xml.XmlDocument)

<Customer NoDuClient="Int">

<*FieldName*>*FieldValue*</*FieldName*> <!—See field list below -->

</Customer>

Exceptions:

Throw if XML is invalid (Invalid XML)

Throw if NoDuClient is missing (NoDuClient missing from XML)

Any exception return by CRM (make sure to throw the message received).

If the NoDuClient does not exist in the account entity, insert record in CRM. If it exists, update record. Only updated the fields received in the XML in CRM. The webservice will be called through a soap call.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input field | Input Type | Required | CRM field | CRM type |
| NoDuClient | Int | Yes | New\_idclient | whole |
| NomDuCompte | Char(255) |  | Name | Text |
| Telephone | Char (50) |  | Telephone1 | Text |
| Adresse1 | Char (250) |  | address1\_line1 | Text |
| Adresse2 | Char (250) |  | address1\_line2 | Text |
| Ville | Char (80) |  | address1\_city | Text |
| Province | Char (50) |  | address1\_stateorprovince | Text |
| Pays | Char(80) |  | address1\_country | Text |
| CodePostal | Char (10) |  | address1\_postalcode | Text |
| DateDeEssai | Date/Time |  | new\_datecreation | Date/Time |
| NombreDeLogin | int |  | new\_nombredelogin | whole |
| ImportationCompletee | Bool |  | new\_Importationcompletee | Yes/no |
| DateImportation | Date |  | new\_dateImportation | Date |
| GabaritCree | Bool |  | new\_gabaritcree | Yes/no |
| DateGabarit | Date |  | new\_dategabaritcree | Date |
| EnvoiComplete | Bool |  | new\_envoicomplete | Yes/no |
| DateEnvoi | Date |  | new\_dateenvoicomplete | Date |
| DateActivation | Date |  | new\_dateconversioncomptepayant | Date |
| Forfait | Enum |  | new\_forfait | Option set |
| OSBL | Bool |  | new\_osbl | Yes/no |
| Premium | Bool |  | new\_comptepremium | Yes/no |
| Actif | Bool |  | new\_actif | Yes/no |
| EnvoiPermis | Bool |  | new\_envoipermis | Yes/no |
| Revendeur | Bool |  | new\_revendeur | Yes/no |
| NoDeRevendeur | Char(20) |  | new\_noderevendeur | text |
| RevenuForfait | Money |  | new\_revenuforfait | Money |
| RevenuExtra | Money |  | new\_revenuextra | Money |
| CourrielFrom | Char (255) |  | new\_courrielFrom | text |
| ExpirationDemo | Date/Time |  | new\_expirationdudemo | Date/Time |
| BDSupprimee | Bool |  | new\_supprime | Yes/no |
| AccesAPI | Bool |  | new\_accessapi | Yes/no |
| AccesAPISansOptIn | Bool |  | new\_accesapisansoptin | Yes/no |
| EntrepriseFacturation | Char (100) |  | new\_entreprisefacturation | text |
| MéthodeDePaiement | Enum |  | new\_methodedepaiment | Option set |

# Method UpdateContact

Parameters:

contactXML (System.Xml.XmlDocument)

<Contact NoDuClient="Int" Nom="`String" Prenom="String" >

<*FieldName*>*FieldValue*</*FieldName*> <!—See field list below -->

</ Contact>

Exceptions:

Throw if XML is invalid (Invalid XML)

Throw if NoDuClient is missing (NoDuClient missing from XML)

Throw if Nom is missing (Nom missing from XML)

Throw if Prenom is missing (Prenom missing from XML)

Any exception return by CRM (make sure to throw the message received).

Find in the contact entity, the contact with the same NoDuClient, Prenom and Nom. If not found, insert record. If found, update record.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input field | Input Type | Required | CRM field | CRM type |
| NoDuClient | int | Yes | parentcustomerid | Lookup |
| Nom | Char (100) | Yes | Lastname | text |
| Prenom | Char (100) | yes | Firstname | text |
| ContactTechnique | Bool |  | New\_contacttechnique | Yes/no |
| ContactFacturation | Bool |  | New\_contactfacturation | Yes/no |
| CyberBulletin | Bool |  | New\_cyberbulletin | Yes/no |
| Langue | Enum |  | New\_languecyberbulletin | Option set |
| Actif | Bool |  | New\_actif | Yes/no |

## Plugin Account Validation (CyberImpact.Plugin.Account.Validations)

On the Pre Insert or Update of an account:

* Validate that if new\_Importationcompletee = true, then new\_dateImportation must be completed. If not, throw InvalidPluginExecutionException with « *NameOfField* is required » (where *NameOfField* is the schema name of the missing field).
* Validate that if new\_gabaritcree = true, then the new\_dategabaritcree must be completed. If not, throw InvalidPluginExecutionException with « *NameOfField* is required ».
* Validate that if new\_envoicomplete = true, then the new\_dateenvoicomplete must be completed. If not, throw InvalidPluginExecutionException with « *NameOfField* is required ».
* Validate that if new\_dateconversioncomptepayant is not null, new\_datedechangementdeforfait must be completed. If not, throw InvalidPluginExecutionException with « *NameOfField* is required ».
* Validate that if new\_datedechangementdeforfait is not null, new\_forfait must be completed. If not, throw InvalidPluginExecutionException with « *NameOfField* is required ».
* If new\_supprime becomes true or new\_actif becomes false, set the new\_datefermeturedecompte to the current date.

For each validations above, perform the validation only if the triggering field is present in the CRM target. If so, use other fields from the target if present else the pre image.

## Plugin Account History (CyberImpact.Plugin.Account.History)

On the Post Create or Update of an account:

* If new\_forfait, new\_comptepremium or new\_osbl changes, create a new record in the new\_forfaithistorique table with the following mapping:

|  |  |
| --- | --- |
| Account | new\_forfaithistorique |
| new\_idclient | new\_idclient |
| *System.DateTime.Now* | new\_dateactivation |
| new\_forfait | new\_forfait |
| new\_osbl | new\_osbl |
| new\_comptepremium | new\_premium |

* If new\_revenuforfait or new\_revenuextra changes, create a new record in the new\_revenuhistorique entity with the following mapping:

|  |  |
| --- | --- |
| Account | new\_revenuhistorique |
| new\_idclient | new\_idclient |
| new\_revenuforfait | new\_revenuforfait |
| new\_revenuextra | new\_revenuextra |
| *System.DateTime.Now* | new\_datedurevenu |

## Plugin Entity Name (CyberImpact.Plugin.Entity.SetName)

On the Pre Create or Update of an account:

If the target Entity is new\_forfaithistorique, set the new\_name field to new\_compte.name + *space* + new\_dateactivation (using format ddMMyyyy).

If the target Entity is new\_revenuhistorique, set the new\_name field to new\_compte.name + space + New\_datedurevenu (using format ddMMyyyy).