
Integración WebServices

01 de Octubre de 2012

TABLA de CONTENIDOS

INTEGRACION DE ENTRADA WEBSERVICES	2
Introducción	2
Pre-requisitos	2
Estándar de comunicación	2
Servicios disponibles	3
Descripción de los métodos	3
Método SendDocument: Generación de un documento	4
Método FiscalStatus: Consulta de estado de autorización	6
Método Tracking: Consulta de tracking de documento	8
Método RecoverXML: Recuperación de documento XML	9
Método RecoverPDF: Recuperación de documento PDF	11
Método Print: Impresión de documento XML	13
Ejemplos de respuesta	16
Método FiscalStatus	16
Método Tracking	16
Método RecoverXML	17
Método RecoverPDF	22

INTEGRACION DE ENTRADA WEBSERVICES

Introducción

En las implementaciones del producto DTEBox Manager es necesario realizar integraciones con sistemas externos de los clientes.

DTEBox Manager ofrece varias alternativas de integración entre las que se encuentran:

- ✓ File System utilizando archivos (TXT, XML, IDOC, etc).
- ✓ Web Services.
- ✓ Base de Datos.
- ✓ Conectores personalizados.

En el presente documento se estará describiendo en detalle la integración a través de Web Services con DTEBox Manager.

Pre-requisitos

Este documento está dirigido a personas con experiencia en programación e integración de aplicaciones y deben contar con los siguientes conocimientos:

- ✓ Lenguaje XML.
- ✓ Tecnología XML Web Services.
- ✓ Conocimientos básicos del producto DTEBox.

Estándar de comunicación

Para poder soportar una mayor cantidad de plataformas en la integración, los servicios expuestos por DTEBox son servicios REST. Lo cual quiere decir que pueden ser consumidos utilizando las funciones básicas del protocolo HTTP 1.1, GET y POST, las cuales están soportadas prácticamente en todos los lenguajes de programación modernos.

En este documento se exponen ejemplos de como utilizar estos servicios en el lenguaje de programación c#.

Servicios disponibles

DTEBox Manager provee los siguientes servicios WEB:

- ✓ **SendDocumentRequest:** Recibe los datos y convierte el documento al formato XML establecido y además aplica la firma digital sobre el mismo.
- ✓ **FiscalStatus:** Entrega el estado de la autorización del o los documentos enviados a SII.
- ✓ **Tracking:** Entrega el Tracking del documento entregado por SII.
- ✓ **RecoverXML:** Recupera el XML del DTE para un documento específico.
- ✓ **RecoverPDF:** Recupera el documento PDF para un DTE específico y generado anteriormente.
- ✓ **Print:** Permite realizar una impresión a solicitud de un documento electrónico en una impresora disponible en el DTE Box.

Servicios disponibles

La autenticación de los web services se realiza a través de una llave de autenticación definida en el DTEBox, para cada ambiente por separado, en la opción Integración.

Para efecto de la construcción de la integración es necesario agregar un header tanto al método POST como al GET, con la clave "AuthKey" y como valor el vigente en el DTEBox.

Ej:

```
request.Headers.Add("AuthKey", "975ffa9e-bbcd-4951-82f2-b081a6d43628");
```

Descripción de los métodos

A continuación se presenta una descripción detallada del método disponible en el Web Services, sus parámetros de entrada y lo que se debería esperar como respuesta de la invocación.

Para poder soportar una mayor cantidad de plataformas en la integración, los servicios expuestos por DTEBox son servicios REST. Lo cual quiere decir que pueden ser consumidos utilizando las funciones básicas del protocolo HTTP 1.1, GET y POST, las cuales están soportadas prácticamente en todo los lenguajes de programación modernos.

En este documento se exponen ejemplos de como utilizar estos servicios en el lenguaje de programación c#.

Método SendDocument: Generación de un documento

Este método se consume vía POST pasando los parámetros agrupados en un XML con el formato especificado a continuación:

```
<SendDocumentRequest xmlns="http://gdexpress.cl/api">
<Environment>{0}</Environment><Type>{1}</Type><Content>{2}</Content>
</SendDocumentRequest>
```

Ej.

```
<SendDocumentRequest xmlns="http://gdexpress.cl/api">
<Environment>T</Environment><Type>33</Type><Content>...contenido en formato
texto del documento a enviar...</Content></SendDocumentRequest>
```

La URI a donde debe ser enviado el POST es:

```
http://{IP_DTEBOX}/api/Core.svc/Core/SendDocument
```

La respuesta del request HTTP por POST a esta URL es un xml con la siguiente estructura:

```
<?xml version="1.0" encoding="utf-8"?>
<SendDocumentResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Data>...información del resultado...</Data>
  <Description i:nil="true" />
  <Result>0</Result>
</SendDocumentResult>
```

El tag <Result> indica el éxito de la llamada, 0 significa OK, distinto de 0 significa que ocurrió algún error en cuyo caso el tag <Description> contiene la descripción del error.

En el caso del retorno OK el tag <Data> contienen la información solicitada. A continuación se presenta una porción código en c# para el consumo de este servicio:

```
static void ej_SendDocument()
{
    //establecer valores a los parámetros del servicio
    char ambiente = 'T';           //Ambiente 'T': Test, 'P': Producción
    string tipoDTE = "33";        //Tipo de documento según codificación SII "33": Factura
    //Electrónica, "34": Factura Exenta Electrónica
    string documento = "";        //Contenido del documento que se carga de un archivo

    //cargar el contenido del documento a enviar
}
```

```

using (System.IO.StreamReader sr = new
System.IO.StreamReader(@"D:\Work\SandBox\GDE.Tests.DTEBox.APIs\GDE.Tests.DTEBox.APIs\resources\docu
ment.txt"))
{
    documento = sr.ReadToEnd();
    sr.Close();
}

string serviceURI = ConfigurationManager.AppSettings["ApiURI"];
string invokeURI = string.Format("{0}/{1}", serviceURI, "SendDocument");

#region Enviar los parametros por POST
string result = string.Empty;
string descripcion = string.Empty;

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(invokeURI);
request.Headers.Add("AuthKey", "975ffa9e-bbcd-4951-82f2-b081a6d43628");
request.Method = "POST";
request.ContentType = "application/xml; charset=utf-8";
request.Timeout = 30000;
request.Headers.Add("SOAPAction", invokeURI);

// Construir el XML a enviar en el POST
string postXMLTemplate = "<SendDocumentRequest
xmlns='http://gdexpress.cl/api'><Environment>{0}</Environment><Type>{1}</Type><Content>{2}</Conte
nt></SendDocumentRequest>";
string postXML = string.Format(postXMLTemplate, ambiente, tipoDTE,
System.Web.HttpUtility.HtmlEncode(documento));

// enviar el flujo de bytes del XML
byte[] postXMLBytes = System.Text.Encoding.UTF8.GetBytes(postXML);
request.ContentLength = postXMLBytes.Length;
using (var sw = request.GetRequestStream())
{
    sw.Write(postXMLBytes, 0, postXMLBytes.Length);
    sw.Flush();

// Leer la respuesta
XDocument response = null;
WebResponse webResponse = request.GetResponse();
using (Stream s = webResponse.GetResponseStream())
{
    XmlReader xr = XmlReader.Create(s);
    response = XDocument.Load(xr);
}

result = response.Descendants().First(n => n.Name.LocalName == "Result").Value;
descripcion = response.Descendants().First(n => n.Name.LocalName == "Description").Value;
}
#endregion

//procesar el resultado
switch (result)
{
    case "0":
        Console.WriteLine("El documento electrónico fue generado con éxito.");
        break;
    default:

```

```

        string errMsg = string.Format("El documento electrónico no pudo ser generado. Error {0}: {1}",
result, descripcion);
        Console.WriteLine(errMsg);
        break;
    }
    Console.ReadLine();
}

```

Parámetro	Descripción	Validación
Environment	Nombre del área creada en DTEBox	El área debe existir en DTEBox.
Type	Código del tipo de documento que se está enviando.	El código debe encontrarse en la tabla de códigos de tipos de documentos.
Content	<p>Contenido del documento en forma de cadena de caracteres (text plano).</p> <p>El formato de este texto se encuentra en el documento "Estructura Archivos TXT", en la hoja "Doc's Tributarios Electrónicos"</p>	

Para permitir la impresión en la factura de descripciones asociadas a los ítemes de facturación, se define un campo de Descripción dentro del formato de DocumentContent, el cual utiliza como carácter de reemplazo para los saltos de línea la cadena \${{

Método FiscalStatus: Consulta de estado de autorización

Este método se consume vía GET pasando los parámetros como segmentos de la URI, en el orden especificado a continuación:

http://{IP_DTEBOX}/api/Core.svc/Core/FiscalStatus/{Ambiente}/{Rut}/{TipoDTE}/{Folio}

Ej.

http://{IP_DTEBOX}/api/Core.svc/Core/FiscalStatus/T/76129486-5/33/733

La respuesta del request HTTP por Get a esta URL es un xml con la siguiente estructura:

```
<?xml version="1.0" encoding="utf-8"?>
<FiscalStatusResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Data>El SII ha rechazado este envío: RCT-Rechazado por Error en Caratula</Data>
  <Description i:nil="true" />
  <Result>0</Result>
  <Status>9</Status>
  <TrackId>3</TrackId>
</FiscalStatusResult>
```

EL tag <Result> indica el éxito de la llamada, 0 significa OK, distinto de 0 significa que ocurrió algún error en cuyo caso el tag <Description> contiene la descripción del error.

En el caso del retorno OK los tags <Status> y <TrackId> contienen la información solicitada.

A continuación se presenta una porción código en c# para el consumo de este servicio:

```
private static XDocument HttpGetXml(string invokeURI)
{
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(invokeURI);
    request.Headers.Add("AuthKey", "975ffa9e-bbcd-4951-82f2-b081a6d43628");

    XDocument response = null;
    WebResponse webResponse = request.GetResponse();
    using (Stream s = webResponse.GetResponseStream())
    {
        XmlReader xr = XmlReader.Create(s);
        response = XDocument.Load(xr);
    }
    return response;
}
static void ej_FiscalStatus()
{
    //establecer valores a los parámetros del servicio
    char ambiente = 'T'; //Ambiente 'T': Test, 'P': Producción
    string rutEmisor = "76129486-5"; //RUT de la empresa emisora del documento
    string tipoDTE = "33"; //Tipo de documento según codificación SII "33": Factura
    Electrónica, "34": Factura Exenta Electrónica
    long folio = 73; //Número de folio del documento que se desea verificar

    //invocar el servicio
    string serviceURI = ConfigurationManager.AppSettings["ApiURI"];
    string invokeURI = string.Format("{0}/{1}/{2}/{3}/{4}/{5}", serviceURI, "FiscalStatus", ambiente,
    rutEmisor, tipoDTE, folio);

    XDocument response = HttpGetXml(invokeURI);

    string result = response.Descendants().First(n => n.Name.LocalName == "Result").Value;
    string descripcion = response.Descendants().First(n => n.Name.LocalName == "Description").Value;

    //procesar el resultado
    switch (result)
    {
```

```

        case "0":
            string status = response.Descendants().First(n => n.Name.LocalName == "Status").Value;
            string trackid = response.Descendants().First(n => n.Name.LocalName == "TrackId").Value;
            Console.WriteLine(string.Format("El estado fiscal del documento es: {0} y se envió al SII con
TrackID: {1}", status, trackid));
            break;
        default:
            Console.WriteLine("Error consultando el estado fiscal del documento.");
            Console.WriteLine(string.Format("Error {0}: {1}", result, descripcion));
            break;
    }
    Console.ReadLine();
}

```

Método Tracking: Consulta de tracking de documento

Este método se consume vía GET pasando los parámetros como segmentos de la URI, en el orden especificado a continuación:

`http://{IP_DTEBOX}/api/Core.svc/Core/Tracking/{Ambiente}/{Rut}/{Tipo DTE}/{Folio}`

Ej.

`http://{IP_DTEBOX}/api/Core.svc/Core/Tracking/T/76129486-5/33/733`

La respuesta del request HTTP por Get a esta URL es un xml con la siguiente estructura:

```

<?xml version="1.0" encoding="utf-8"?>
<TrackingResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Data>...xml con la información del tracking...</Data>
  <Description i:nil="true" />
  <Result>0</Result>
</TrackingResult>

```

El tag <Result> indica el éxito de la llamada, 0 significa OK, distinto de 0 significa que ocurrió algún error en cuyo caso el tag <Description> contiene la descripción del error.

En el caso del retorno OK el tag <Data> contienen la información solicitada.

A continuación se presenta una porción código en C# para el consumo de este servicio:

```

private static XDocument HttpGetXml(string invokeURI)
{
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(invokeURI);
    request.Headers.Add("AuthKey", "975ffa9e-bbcd-4951-82f2-b081a6d43628");

    XDocument response = null;
    WebResponse webResponse = request.GetResponse();
    using (Stream s = webResponse.GetResponseStream())

```

```

    {
        XmlReader xr = XmlReader.Create(s);
        response = XDocument.Load(xr);
    }
    return response;
}

static void ej_Tracking()
{
    //establecer valores a los parámetros del servicio
    char ambiente = 'T';           //Ambiente 'T': Test, 'P': Producción
    string rutEmisor = "76129486-5"; //RUT de la empresa emisora del documento
    string tipoDTE = "33";        //Tipo de documento según codificación SII "33": Factura
    //Electrónica, "34": Factura Exenta Electrónica
    long folio = 73;              //Número de folio del documento que se desea verificar

    //invocar el servicio
    string serviceURI = ConfigurationManager.AppSettings["ApiURI"];
    string invokeURI = string.Format("{0}/{1}/{2}/{3}/{4}/{5}", serviceURI, "Tracking", ambiente,
rutEmisor, tipoDTE, folio);

    XDocument response = HttpGetXml(invokeURI);

    string result = response.Descendants().First(n => n.Name.LocalName == "Result").Value;
    string descripcion = response.Descendants().First(n => n.Name.LocalName == "Description").Value;

    //procesar el resultado
    switch (result)
    {
        case "0":
            string data = response.Descendants().First(n => n.Name.LocalName == "Data").Value;
            Console.WriteLine("Tracking del documento: ");
            Console.WriteLine(data);
            break;
        default:
            Console.WriteLine("Error consultando el tracking del documento.");
            Console.WriteLine(string.Format("Error {0}: {1}", result, descripcion));
            break;
    }
    Console.ReadLine();
}
}

```

Método RecoverXML: Recuperación de documento XML

Este método se consume vía GET pasando los parámetros como segmentos de la URI, en el orden especificado a continuación:

`http://{IP_DTEBOX}/api/Core.svc/Core/RecoverXML/{Ambiente}/{Rut}/{TipoDTE}/{Folio}`

Ej.

http://{IP_DTEBOX}/api/Core.svc/Core/RecoverXML/T/76129486-5/33/733

La respuesta del request HTTP por Get a esta URL es un xml con la siguiente estructura:

```
<?xml version="1.0" encoding="utf-8"?>
<RecoverDocumentResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Data>...XML del DTE solicitado...</Data>
  <Description i:nil="true" />
  <Result>0</Result>
</RecoverDocumentResult>
```

El tag <Result> indica el éxito de la llamada, 0 significa OK, distinto de 0 significa que ocurrió algún error en cuyo caso el tag <Description> contiene la descripción del error.

En el caso del retorno OK el tag <Data> contienen la información solicitada.

A continuación se presenta una porción código en c# para el consumo de este servicio:

```
private static XDocument HttpGetXml(string invokeURI)
{
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(invokeURI);
    request.Headers.Add("AuthKey", "975ffa9e-bbcd-4951-82f2-b081a6d43628");

    XDocument response = null;
    WebResponse webResponse = request.GetResponse();
    using (Stream s = webResponse.GetResponseStream())
    {
        XmlReader xr = XmlReader.Create(s);
        response = XDocument.Load(xr);
    }
    return response;
}

static void ej_RecoverXML()
{
    //establecer valores a los parámetros del servicio
    char ambiente = 'T';           //Ambiente 'T': Test, 'P': Producción
    string rutEmisor = "76129486-5"; //RUT de la empresa emisora del documento
    string tipoDTE = "33";        //Tipo de documento según codificación SII "33":
    Factura Electrónica, "34": Factura Exenta Electrónica
    long folio = 73;              //Número de folio del documento que se desea verificar

    //invocar el servicio
```

```

string serviceURI = ConfigurationManager.AppSettings["ApiURI"];
string invokeURI = string.Format("{0}/{1}/{2}/{3}/{4}/{5}", serviceURI, "RecoverXML",
ambiente, rutEmisor, tipoDTE, folio);

XDocument response = HttpGetXml(invokeURI);

string result = response.Descendants().First(n => n.Name.LocalName == "Result").Value;
string descripcion = response.Descendants().First(n => n.Name.LocalName ==
"Description").Value;

//procesar el resultado
switch (result)
{
    case "0":
        //deserializar la respuesta
        string path =
@"D:\Work\SandBox\GDE.Tests.DTEBox.APIs\GDE.Tests.DTEBox.APIs\resources\DTE.xml";
        string data = response.Descendants().First(n => n.Name.LocalName ==
"Data").Value;
        File.WriteAllText(path, data);
        Console.WriteLine(string.Format("El archivo XML asociado al documento fue
guardado en: {0}", path));
        break;
    default:
        Console.WriteLine("Error consultando el tracking del documento.");
        Console.WriteLine(string.Format("Error {0}: {1}", result, descripcion));
        break;
}
Console.ReadLine();
}

```

Método RecoverPDF: Recuperación de documento PDF

Este método se consume vía GET pasando los parámetros como segmentos de la URI, en el orden especificado a continuación:

```
http://{IP_DTEBOX}/api/Core.svc/Core/RecoverPDF/{Ambiente}/{Rut}/{Tipo
DTE}/{Folio}
```

Ej.

```
http://{IP_DTEBOX}/api/Core.svc/Core/RecoverPDF/T/76129486-5/33/733
```

La respuesta del request HTTP por Get a esta URL es un xml con la siguiente estructura:

```
<?xml version="1.0" encoding="utf-8"?>
<RecoverDocumentResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
```

```

<Data>...PDF del DTE solicitado codificado en formato base64...</Data>
<Description i:nil="true" />
<Result>0</Result>
</RecoverDocumentResult>

```

El tag <Result> indica el éxito de la llamada, 0 significa OK, distinto de 0 significa que ocurrió algún error en cuyo caso el tag <Description> contiene la descripción del error.

En el caso del retorno OK el tag <Data> contienen la información solicitada (PDF del DTE solicitado codificado en formato base64).

A continuación se presenta una porción código en c# para el consumo de este servicio:

```

private static XDocument HttpGetXml(string invokeURI)
{
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(invokeURI);
    request.Headers.Add("AuthKey", "975ffa9e-bbcd-4951-82f2-b081a6d43628");

    XDocument response = null;
    WebResponse webResponse = request.GetResponse();
    using (Stream s = webResponse.GetResponseStream())
    {
        XmlReader xr = XmlReader.Create(s);
        response = XDocument.Load(xr);
    }
    return response;
}

static void ej_RecoverPDF()
{
    //establecer valores a los parámetros del servicio
    char ambiente = 'T'; //Ambiente 'T': Test, 'P': Producción
    string rutEmisor = "76129486-5"; //RUT de la empresa emisora del documento
    string tipoDTE = "33"; //Tipo de documento según codificación SII "33":
    Factura Electrónica, "34": Factura Exenta Electrónica
    long folio = 73; //Número de folio del documento que se desea verificar

    //invocar el servicio
    string serviceURI = ConfigurationManager.AppSettings["ApiURI"];
    string invokeURI = string.Format("{0}/{1}/{2}/{3}/{4}/{5}", serviceURI, "RecoverPDF",
    ambiente, rutEmisor, tipoDTE, folio);

    XDocument response = HttpGetXml(invokeURI);

    string result = response.Descendants().First(n => n.Name.LocalName == "Result").Value;
    string descripcion = response.Descendants().First(n => n.Name.LocalName ==
    "Description").Value;

    //procesar el resultado

```

```

switch (result)
{
    case "0":
        //deserializar la respuesta
        string path =
@"D:\Work\SandBox\GDE.Tests.DTEBox.APIs\GDE.Tests.DTEBox.APIs\resources\DTE.pdf";
        string data = response.Descendants().First(n => n.Name.LocalName ==
"Data").Value;
        File.WriteAllBytes(path, System.Convert.FromBase64String(data));
        Console.WriteLine(string.Format("El archivo PDF asociado al documento fue
guardado en: {0}", path));
        break;
    default:
        Console.WriteLine("Error consultando el tracking del documento.");
        Console.WriteLine(string.Format("Error {0}: {1}", result, descripcion));
        break;
}
Console.ReadLine();
}

```

Método Print: Impresión de documento XML

Este método se consume vía GET pasando los parámetros como segmentos de la URI, en el orden especificado a continuación:

```

http://{IP_DTEBOX}/api/Core.svc/Core/Print/{Ambiente}/{Rut}/{Tipo
DTE}/{Folio}/{Printer}

```

Ej.

```

http://{IP_DTEBOX}/api/Core.svc/Core/Print/T/76129486-5/33/733/Epson-LX300

```

La respuesta del request HTTP por Get a esta URL es un xml con la siguiente estructura:

```

<?xml version="1.0" encoding="utf-8"?>
<RecoverDocumentResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Data></Data>
  <Description i:nil="true" />
  <Result>0</Result>
</RecoverDocumentResult>

```

El tag <Result> indica el éxito de la llamada, 0 significa OK, distinto de 0 significa que ocurrió algún error en cuyo caso el tag <Description> contiene la descripción del error.

En tag <Data> siempre se retorna vacío.

A continuación se presenta una porción código en c# para el consumo de este servicio:

```
private static XDocument HttpGetXml(string invokeURI)
{
    HttpWebRequest request = (HttpWebRequest)WebRequest.Create(invokeURI);
    request.Headers.Add("AuthKey", "975ffa9e-bbcd-4951-82f2-b081a6d43628_sdfsdf");

    XDocument response = null;
    WebResponse webResponse = request.GetResponse();
    using (Stream s = webResponse.GetResponseStream())
    {
        XmlReader xr = XmlReader.Create(s);
        response = XDocument.Load(xr);
    }
    return response;
}

static void ej_Print()
{
    //establecer valores a los parámetros del servicio
    char ambiente = 'T'; //Ambiente 'T': Test, 'P': Producción
    string rutEmisor = "76129486-5"; //RUT de la empresa emisora del documento
    string tipoDTE = "33"; //Tipo de documento según codificación SII "33":
    Factura Electrónica, "34": Factura Exenta Electrónica
    long folio = 73; //Número de folio del documento que se desea verificar
    string printer = "Epson-LX300"; //Nombre de la impresora a utilizar
    (especificar "none" para usar la impresora predeterminada)

    //invocar el servicio
    string serviceURI = ConfigurationManager.AppSettings["ApiURI"];
    string invokeURI = string.Format("{0}/{1}/{2}/{3}/{4}/{5}/{6}", serviceURI, "RecoverXML",
    ambiente, rutEmisor, tipoDTE, folio, printer);

    XDocument response = HttpGetXml(invokeURI);

    string result = response.Descendants().First(n => n.Name.LocalName == "Result").Value;
    string descripcion = response.Descendants().First(n => n.Name.LocalName ==
    "Description").Value;

    //procesar el resultado
    switch (result)
    {
        case "0":
            Console.WriteLine("El documento fue enviado a impresión");
            break;
        default:
            Console.WriteLine("Error imprimiendo el documento.");
            Console.WriteLine(string.Format("Error {0}: {1}", result, descripcion));
            break;
    }
}
```

```
    Console.ReadLine();  
}
```

Ejemplos de respuesta

Método FiscalStatus

```
<?xml version="1.0" encoding="utf-8"?>
<FiscalStatusResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Data>El SII ha rechazado este envío: RCT-Rechazado por Error en
Caratula</Data>
  <Description i:nil="true" />
  <Result>0</Result>
  <Status>9</Status>
  <TrackId>3</TrackId>
</FiscalStatusResult>
```

Método Tracking

```
<?xml version="1.0" encoding="utf-8"?>
<TrackingResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Data>&lt;?xml version="1.0" encoding="iso-8859-1"?&gt;
&lt;DocumentTracking id="10939"&gt;
  &lt;StatusTrack&gt;
    &lt;IdHistory&gt;48920&lt;/IdHistory&gt;
    &lt;IdDocument&gt;10939&lt;/IdDocument&gt;
    &lt;TimeStamp&gt;2011-09-13 12:33:35&lt;/TimeStamp&gt;
    &lt;IdStatus&gt;1&lt;/IdStatus&gt;
    &lt;Status&gt;Recibido&lt;/Status&gt;
    &lt;Comments /&gt;
    &lt;TrackId /&gt;
  &lt;/StatusTrack&gt;
  &lt;StatusTrack&gt;
    &lt;IdHistory&gt;49120&lt;/IdHistory&gt;
    &lt;IdDocument&gt;10939&lt;/IdDocument&gt;
    &lt;TimeStamp&gt;2011-09-13 12:37:05&lt;/TimeStamp&gt;
    &lt;IdStatus&gt;2&lt;/IdStatus&gt;
    &lt;Status&gt;Enviado SII&lt;/Status&gt;
    &lt;Comments&gt;Enviado al SII con TrackId:
14553776&lt;/Comments&gt;
    &lt;TrackId /&gt;
  &lt;/StatusTrack&gt;
  &lt;StatusTrack&gt;
    &lt;IdHistory&gt;49320&lt;/IdHistory&gt;
    &lt;IdDocument&gt;10939&lt;/IdDocument&gt;
    &lt;TimeStamp&gt;2011-09-13 12:37:36&lt;/TimeStamp&gt;
    &lt;IdStatus&gt;3&lt;/IdStatus&gt;
    &lt;Status&gt;SET Rechazado SII&lt;/Status&gt;
    &lt;Comments&gt;El SII ha rechazado este envío: RCT-
Rechazado por Error en Caratula&lt;/Comments&gt;
    &lt;TrackId&gt;14553776&lt;/TrackId&gt;
  &lt;/StatusTrack&gt;
  &lt;StatusTrack&gt;
```

```

        <IdHistory>49570</IdHistory>
        <IdDocument>10939</IdDocument>
        <TimeStamp>2011-09-13 12:42:16</TimeStamp>
        <IdStatus>21</IdStatus>
        <Status>Reenviar SII</Status>
        <Comments>Cambio de estado
automático.</Comments>
        <TrackId />
    </StatusTrack>
    <StatusTrack>
        <IdHistory>49720</IdHistory>
        <IdDocument>10939</IdDocument>
        <TimeStamp>2011-09-13 12:47:06</TimeStamp>
        <IdStatus>2</IdStatus>
        <Status>Enviado SII</Status>
        <Comments>Enviado al SII con TrackId:
14553808</Comments>
        <TrackId />
    </StatusTrack>
    <StatusTrack>
        <IdHistory>49870</IdHistory>
        <IdDocument>10939</IdDocument>
        <TimeStamp>2011-09-13 12:47:36</TimeStamp>
        <IdStatus>3</IdStatus>
        <Status>SET Rechazado SII</Status>
        <Comments>El SII ha rechazado este envío: RCT-
Rechazado por Error en Caratula</Comments>
        <TrackId>14553808</TrackId>
    </StatusTrack>
</DocumentTracking></Data>
    <Description i:nil="true" />
    <Result>0</Result>
</TrackingResult>

```

Método RecoverXML

```

<?xml version="1.0" encoding="utf-8"?>
<RecoverDocumentResult xmlns="http://gdexpress.cl/api"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <Data><?xml version="1.0" encoding="ISO-8859-1"?>
    <EnvioDTE version="1.0" xmlns="http://www.sii.cl/SiiDte"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.sii.cl/SiiDte EnvioDTE_v10.xsd">
        <SetDTE ID="SETDTE5d6e573551c546ea86a2409bea7f81c5">
            <Caratula version="1.0">
                <RutEmisor>76129486-5</RutEmisor>
                <RutEnvia>76129486-5</RutEnvia>
                <RutReceptor>60803000-K</RutReceptor>
                <FchResol>2011-08-23</FchResol>
                <NroResol>0</NroResol>
                <TmstFirmaEnv>2011-09-09T16:40:32</TmstFirmaEnv>
            </Caratula>
        </SetDTE>
    </EnvioDTE>
    </Data>
</RecoverDocumentResult>

```

```

&lt;SubTotDTE&gt;&lt;TpoDTE&gt;33&lt;/TpoDTE&gt;&lt;NroDTE&gt;1&lt;
;/NroDTE&gt;&lt;/SubTotDTE&gt;&lt;/Caratula&gt;
&lt;DTE version="1.0"&gt;
&lt;Documento ID="F73T33"&gt;
&lt;Encabezado&gt;
&lt;IdDoc&gt;
&lt;TipoDTE&gt;33&lt;/TipoDTE&gt;
&lt;Folio&gt;73&lt;/Folio&gt;
&lt;FchEmis&gt;2011-04-04&lt;/FchEmis&gt;
&lt;FmaPago&gt;1&lt;/FmaPago&gt;
&lt;MntPagos&gt;
&lt;FchPago&gt;2011-05-04&lt;/FchPago&gt;
&lt;MntPago&gt;1618400&lt;/MntPago&gt;
&lt;GlosaPagos&gt;2&lt;/GlosaPagos&gt;
&lt;/MntPagos&gt;
&lt;FchVenc&gt;2011-05-04&lt;/FchVenc&gt;
&lt;/IdDoc&gt;
&lt;Emisor&gt;
&lt;RUTEmisor&gt;76129486-5&lt;/RUTEmisor&gt;
&lt;RznSoc&gt;FIBRA INGENIERIA Y CONTRUCCION
S.A.&lt;/RznSoc&gt;
&lt;GiroEmis&gt;FABRICACION DE ARTICULOS DE
PLASTICOS&lt;/GiroEmis&gt;
&lt;Telefono&gt;4895500&lt;/Telefono&gt;
&lt;Acteco&gt;172990&lt;/Acteco&gt;
&lt;DirOrigen&gt;STA. MARGARITA 0750&lt;/DirOrigen&gt;
&lt;CmnaOrigen&gt;SAN BERNARDO&lt;/CmnaOrigen&gt;
&lt;CiudadOrigen&gt;SANTIAGO&lt;/CiudadOrigen&gt;
&lt;/Emisor&gt;
&lt;Receptor&gt;
&lt;RUTRecep&gt;79645260-9&lt;/RUTRecep&gt;
&lt;RznSocRecep&gt;AGRICOLA POLPAICO
S.A.&lt;/RznSocRecep&gt;
&lt;GiroRecep&gt;Agr?cola&lt;/GiroRecep&gt;
&lt;Contacto&gt;MAURICIO CAVIERES&lt;/Contacto&gt;
&lt;DirRecep&gt;.&lt;/DirRecep&gt;
&lt;CmnaRecep&gt;s/n&lt;/CmnaRecep&gt;
&lt;DirPostal&gt;EL GOLF 99&lt;/DirPostal&gt;
&lt;CmnaPostal&gt;LAS CONDES&lt;/CmnaPostal&gt;
&lt;CiudadPostal&gt;Santiago&lt;/CiudadPostal&gt;
&lt;/Receptor&gt;
&lt;Totales&gt;
&lt;MntNeto&gt;1360000&lt;/MntNeto&gt;
&lt;MntExe&gt;0&lt;/MntExe&gt;
&lt;TasaIVA&gt;19.00&lt;/TasaIVA&gt;
&lt;IVA&gt;258400&lt;/IVA&gt;
&lt;MntTotal&gt;1618400&lt;/MntTotal&gt;
&lt;/Totales&gt;
&lt;/Encabezado&gt;
&lt;Detalle&gt;
&lt;NroLinDet&gt;1&lt;/NroLinDet&gt;

```

```

    <CdgItem>
      <TpoCodigo>INT1</TpoCodigo>
      <VlrCodigo>ROT.201.01330</VlrCodigo>
    </CdgItem>
    <NmbItem>ECOTANK CILINDRICO VERTICAL QUIMICO
30.000 Litros lra Verde.</NmbItem>
    <QtyItem>1.000000</QtyItem>
    <PrcItem>3260000.000000</PrcItem>
    <DescuentoPct>25.00</DescuentoPct>
    <DescuentoMonto>815000</DescuentoMonto>
    <MontoItem>2445000</MontoItem>
  </Detalle>
  <Detalle>
    <NroLinDet>2</NroLinDet>
    <CdgItem>
      <TpoCodigo>INT1</TpoCodigo>
      <VlrCodigo>FLT.220.00001</VlrCodigo>
    </CdgItem>
    <NmbItem>SERVICIO FLETE ROTOMOLDEO</NmbItem>
    <QtyItem>1.000000</QtyItem>
    <PrcItem>180000.000000</PrcItem>
    <DescuentoPct>1.00</DescuentoPct>
    <MontoItem>180000</MontoItem>
  </Detalle>
  <DscRcgGlobal>
    <NroLinDR>1</NroLinDR>
    <TpoMov>D</TpoMov>
    <TpoValor>$</TpoValor>
    <ValorDR>1265000.00</ValorDR>
  </DscRcgGlobal>
  <Referencia>
    <NroLinRef>1</NroLinRef>
    <TpoDocRef>50</TpoDocRef>
    <FolioRef>21127</FolioRef>
    <FchRef>2011-04-01</FchRef>
  </Referencia>
  <TED version="1.0">
    <DD>
      <RE>76129486-5</RE>
      <TD>33</TD>
      <F>73</F>
      <FE>2011-04-04</FE>
      <RR>79645260-9</RR>
      <RSR>AGRICOLA POLPAICO S.A.</RSR>
      <MNT>1618400</MNT>
      <IT1>ECOTANK CILINDRICO VERTICAL QUIMICO
30.</IT1>
      <CAF version="1.0">
        <DA>
          <RE>76129486-5</RE>
          <RS>GDE S.A.</RS>
          <TD>33</TD>

```

```

    <RNG>
      <D>51</D>
      <H>150</H>
    </RNG>
    <FA>2011-09-09</FA>
    <RSAPK>

<M>8F0rgYYOMOh3lidDwMFTexUaiG7ySPcTxLmFH7ElfcvYnJLXnQtUQ74Md
uVHYdYuo/s0z8b69y4FgELBdrsW/Q==</M>
    <E>Aw==</E>
    </RSAPK>
    <IDK>100</IDK>
    </DA>
    <FRMA
algoritmo="SHA1withRSA">AWJd8taPc8UVySvzPlRGyTvxErSGJCN6un38tqk
f0kyKgD7wocuhi2CibC7UtNjYOEv47RqXAtBhLOYWrAJA7Q==</FRMA>
    </CAF>
    <TSTED>2011-09-09T16:40:32</TSTED>
    </DD>
    <FRMT
algoritmo="SHA1withRSA">MPjM+PRb2cOgok0+YkKw1phNjLQaNvNaLtnAnin
+YhUa6L3ZDpgLtsy0jE2adIHz1DU2taWxzlnyV4SbzvoVng==</FRMT>
    </TED>
    <TmstFirma>2011-09-09T16:40:34</TmstFirma>
    </Documento><Signature
xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><l
t;CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-
xml-c14n-20010315" /><SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"
/><Reference
URI="#F73T33"><Transforms><Transform
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
/></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
/><DigestValue>+sjzbDNYkN4HdtfQThYMV+nx5uw=</DigestVal
ue></Reference></SignedInfo><SignatureValue>
Mw9+j3qXgSvLDs/1ahmRRbsUxeb3odbNpA2+3t+rbzx1KecKApH1VuavTrYbVi7D
kRMxBmSL+ThW/xSzLLZ/3m20qwBgRIvN9Dr9zpfuyn5ApQfnhf9JufcDzospixX5
51AbVtRSHJrWx/PEf0k0eLCATU4159umWKKlZ+7dDrk=
</SignatureValue><KeyInfo><KeyValue><RSAKeyVa
lue><Modulus>
qMlxkvFxnSfhu1SmMxEQvnEfat5VKz3noqPj1Hk3Xv2MB4UvwgihCHuIazD+BrN
T9Wb286wGce7IQ7CulWjbjEISXPku9GEVxQYRvasjaIemOOZTLHnQ4zYsX/hbK8Q
ljMTTWd991Njy+MdWIovyBrDJLASgwxUXS7R3lteL8E=
</Modulus><Exponent>AQAB</Exponent></RSAKeyVa
lue></KeyValue><X509Data><X509Certificate>
MIIFzjCCBLagAwIBAgICBkEwDQYJKoZIhvcNAQEFBQAwgAYxCzAJBgNVBAYTAkNM
MRgwFgYDVQQKEw9BY2VwdGEuY29tIFMuQS4xSDBGBGNVBAMTP0FjZXB0YS5jb20g
QXV0b3JpZGFkIENlcnRpZmljYWRvcmeGQ2xhc2UgMyBQZXJzb25hIE5hdHVyYVwWg
LSBWMjEjEEMwGCSqGSIb3DQEJARYPaW5mb0BhY2VwdGEuY29tMRMwEQYDVQQFEwo5
NjKxOTA1MC04MB4XDTEwMzE1MzAyOV0xODUzMTIwMzE1MzAyOVowGyMxCzAJ
BgNVBAYTAkNMRRwGgYDVQQMEwNJTkdFTk1Fuk8gQ09NRVJDSUFMMR0wGwYDVQQD

```

```

ExRNQVVSSUNFIETIQU1JUyBNQVNTVTEjMCEGCSqGS Ib3DQEJARYUbWF1cmljZS5r
aGFtaXNAbWsuY2wxEjAQBGNVBAUTCTU2NjYxNTQtOTCBnzANBqkqhkiG9w0BAQEF
AAOBjQAwgYkCgYEAqMlxkvFxnsSFhu1SmMxEQvnEfat5VKz3noqPJlHk3Xv2MB4U
vwgihCHuIazD+BrNT9Wb286wGce7IQ7CulWbjEISXPKu9GEVxQYRvasjaIemOOZ
TLHnQ4zYsX/hbK8QljMTTWd991Nyj+MdWIovyBrDJLASgwxUXS7R3lteL8ECAwEA
AaOCAqkwggKlMB8GA1UdIwQYMBaAFEDf6PZWEPl0Gp6EzCII0DatKSZRMBOGA1Ud
DgQWBBDfWKhY+0UGJeE3FZA+DXQ7lDxWTALBgNVHQ8EBAMCBPAwHQYDVR0lBBYw
FAYIKwYBBQUHAwIGCCsGAQUFBwMEMBEGCWCsGAGG+EIBAQQEAWIFoDCB8gYDVR0g
BIHqMIHnMIHkBgorBgEEAbVrh2kCMIHVmCCGCCsGAQUFBwIBFhtodHRwOi8vd3d3
LmFjZXB0YS5jb20vQlBTvjIwgakGCCsGAQUFBwICMIGcMBYWD0FjZXB0YS5jb20g
Uy5BLjADAgECGoGBRWwgdG10dWxhciBoYSBzaWRvIHZhbGlkYWRvIGVuIGZvcmlh
IHByZXNlbnNpYWwsIHFlZWRhbmRvIGhhYmlsaXRhZG8gZWwgQ2VydGlmawNnZG8g
cGFyYSB1c28gdHJpYnV0YXJpbywgGfNb3MsIGNvbWVvY2l1vIHkqb3Ryb3MuMFWG
A1UdEgRVMFOgGAYIKwYBBAHBAQKgDBYKOTY5MTkwNTAtOKAmBgorBgEFBQcAEggD
oBgwFgwKOTY5MTkwNTAtOAYIKwYBBAHBAQKBD2luZm9AYWNlcHRhLmNvbTBfbGNV
HREEDBWoBcGCCsGAQQBwQEB0AsWCTU2NjYxNTQtOaA1BgorBgEFBQcAEggDoBcw
FQwJNTY2NjE1NC05BggrBgEEAcEBAoEUbWF1cmljZS5raGFtaXNAbWsuY2wwOwYI
KwYBBQUHAQEELzAtMCsGCCsGAQUFBzABhh9odHRwOi8vb2Nzc5hY2VwdGEuY29t
L0NsYXNlM1YyMDMGA1UdHwQsMCowKKAmoCSGImh0dHA6Ly9jcmwYWNlcHRhLmNv
bs9DbGFzZTNWmi5jcmwwDQYJKoZIhvcNAQEFBQADggEBAAYRB+ijsWOB1LUaSO+C
z8FUonluIu8zeuT7j6sWBVqsXqegfn4ejKX8Z9WAccVdscKRgsIgszH1pu4T2Cxf
ATef2m6BF/j4OGGzchIS0+HDEMGPzduQK2HwcwvVTCMQYVF17yk/s1xVaCB9sHy/
eH/lobxu6Csg0ZZvMOnIT2JLuembud9pDOVT1QdZcI+9Wp2eGupPVwvwp+TM0dVa
0EMzQSf4AxNCq9qj0gIK1YcGSC+xfS1lx1F7BrKhHJUSCxerZetfXcFgJQJm43hY
rKD0Qn/Ep/fIMoaPdixB14e8zXNus2NouFsk+3un/NxsS7Z3W042qs2DiaSKPOZX
oCA=
</X509Certificate></X509Data></KeyInfo></Signature>
</DTE></SetDTE>

</Personalizados></DocPersonalizado dteID="F73T33">
</campoString name="Glosa">NP 12342. Basado en Gu?a
21127 del 1/4</campoString>
</campoString name="Fpago">30 dias fecha factura %
descto</campoString>
</campoString name="VENDEDOR">17</campoString>

</DocPersonalizado></Personalizados></EnvioDTE><
/Data>
<Description i:nil="true" />
<Result>0</Result>
</RecoverDocumentResult>

```

