



API – Manual de integración Certificados SMS



Contenido

1. Servicios SMS	2
2. Credenciales	2
3. Remitente	2
4. Razón Social	2
5. Actas de entrega	3
6. Funcionalidades	4
7. Métodos.....	4
8. Ejemplo de integración.....	11
9. Códigos de Error	15
10. Estados de mensaje	16
11. Caracteres admitidos.....	16
12. Terminología.....	17
13. Anexo	17

1. Servicios SMS

Antes de enviar SMS Certificados utilizando el API, es obligatorio aceptar las condiciones de uso del servicio. Para ello, debe entrar en el servicio Web por primera vez y aceptar las condiciones de uso.

Para utilizar los servicios SMS del API, debe integrar un cliente con el siguiente servicio web:

<https://certificadossms.didimo.es/api/Service.asmx>

2. Credenciales

Para hacer uso del servicio debe disponer de unas credenciales (usuario y contraseña) facilitadas por soporte. Las credenciales son específicas del API y distintas de las utilizadas para acceder a Aplicateca. Todos los métodos del servicio deben ser informados con las credenciales.

Puede solicitarlas contactando por email a soporte.aplicateca@linkmobility.com indicando el nombre de la empresa y persona de contacto.

3. Remitente

Para hacer uso del servicio es recomendable disponer de un remitente predefinido en el servicio Certificados SMS. Para ello, debe haber entrado en el interfaz Web y haber definido un remitente en el menú “Configuración”.

Los remitentes pueden informarse en el uso del API, pero en caso de no hacerlo se utiliza el predefinido.

4. Razón Social

Para hacer uso del servicio es indispensable informar de la razón social en el Certificados SMS. Para ello, debe haber entrado en el interfaz Web y haber informado la razón social en el menú “Configuración”.

La razón social se utiliza en la información reflejada en el acta de entrega.



La entidad Link Mobility Spain S.L.U. con CIF B85871333 debidamente inscrita en el Registro de Operadores de Red y autorizada para el servicio de Almacenamiento y Reenvío de mensajes cortos, según resolución de la Comisión Nacional de los Mercados y la Competencia con número de expediente RO2010/1448 a través de su servicio de SMS Certificado, CERTIFICA que:

El usuario dado de alta con razón social Didimo Servicios Mviles usando el remitente COLEGIO envió un SMS certificado al destinatario +34685787127 con el siguiente texto:

"Estimada Ana, le han concedido la beca de estudios y comedor del curso 2017-2018. Un saludo (SMS Certificado)"

El mensaje, según consta en nuestros registros, fue enviado a las 18:01 del día 28 de Noviembre de 2017, y finalizó a las 18:01 del día 28 de Noviembre de 2017 con el estado Finalizado en Destinatario, lo cual se certifica a instancias del propio interesado a los efectos probatorios conforme al derecho que estime pertinente.

El código del envío es: 1d22ef9b-fafb-4668-89fc-3cebc41cb94d.

Madrid, a 28 de Noviembre de 2017

Documento certificado por 51649248B JOSE RAMIRO CARAMES (R:B84481183) <jcarames@sigma-data.com>.

Firma electrónica del Servicio de SMS certificado

Fecha: 2017.11.28 18:01:15

Reason: Sellado electrónico de Sigma Data Services

Location: Madrid

5. Actas de entrega

La entidad encargada de ejecutar la firma digital es Sigma Data Services, para evitar cualquier tipo de modificación posterior. Se añade un sello de tiempo digital para validar la fecha y hora de emisión del certificado. La firma digital y el sello de tiempo están emitidos por la Autoridad Certificadora Firma Profesional.

Una vez generada el acta de entrega en formato PDF, se pone a disposición del usuario para que lo guarde localmente. Las actas se guardan en los servidores de certificados SMS durante cinco años, no obstante, el usuario es responsable del almacenamiento a largo plazo de sus actas de entrega.

Sobre Sigma Data Services

Sigma Data Services es una compañía del sector de las Tecnologías de la Información líder en servicios y soluciones avanzadas en procesos de captura, gestión, tratamiento, certificación y transferencia de datos y documentos. Sigma, a través de diferentes Autoridades de Certificación, provee el servicio de certificación y archivo de mensajes, a largo

plazo, garantizando la integridad de los mismos y en entorno jurídico seguro para los negocios basados en transacciones digitales.

Sobre Firma Profesional

Firma Profesional nació como un proyecto de diversos colegios profesionales con el fin de actuar con total independencia como Autoridad de Certificación Digital de los Profesionales. Firma Profesional es una CA reconocida por el Ministerio de Industria, Turismo y Comercio.

Firma Profesional, líder en el sector privado de certificados, es un operador global de firma que emite Certificados Digitales y genera sobre ellos una serie de servicios de valor añadido para el mercado. Empresa pionera en España como Autoridad de Certificación, emite certificados digitales especializados tanto para los profesionales, sus colegios y colectivos, como para las empresas y sus empleados.

El acta de entrega del SMS certificado contiene la siguiente información en referencia al mensaje:

- Texto del mensaje
- Remitente del mensaje
- Móvil del destinatario
- Fecha de envío: fecha de entrega del SMS desde el servicio de mensajería de LINK MOBILITY SPAIN al operador móvil
- Fecha de Finalización: fecha de recepción del estado final del SMS desde el operador móvil a LINK MOBILITY SPAIN

El acta de entrega del SMS certificado, en formato PDF, se puede obtener de diferentes maneras:

- **Email:** En el interfaz Web, página Configuración, se debe indicar la dirección de correo electrónico donde se quieren recibir las actas de entrega. Las actas se envían desde la siguiente cuenta de correo: smscertificados@didimo.es. Es recomendable tenerlo en cuenta para no recibir estos emails como spam.
- **Interfaz Web:** Todas las actas de entrega están disponibles para descargar desde el interfaz Web, página Informes, independientemente de que se reciban por email o no.
- **Método GetCertifyFile**

6. Funcionalidades

El API ofrece las siguientes funcionalidades:

- Consulta de disponibilidad del servicio (método [Ping](#))
- Obtención de identificador único (GUID) de mensaje (método [GetMessageId](#))
- Envío de un mensaje (método [CreateMessage](#))
- Envío de N mensajes (método [CreateSend](#))
- Obtención del estado de un mensaje (método [GetMessageStatus](#))
- Consulta de Créditos disponible para enviar SMS (método [GetCredits](#))
- Descarga de acta de certificación (método [GetCertifyFile](#))

7. Métodos

El conjunto de métodos es el siguiente:

Service

Las operaciones siguientes son compatibles. Para una definición formal, revise la [descripción de servicios](#).

- [CreateMessage](#)
- [CreateSend](#)
- [GetCertifyFile](#)
- [GetCredits](#)
- [GetMessageId](#)
- [GetMessageStatus](#)
- [Ping](#)

Todos los métodos deben informarse con las credenciales de uso de API facilitadas por soporte y cualquier otro dato necesario para su funcionalidad.

Parámetros de credenciales:

- **login** : nombre de usuario (ver credenciales), obligatorio, tipo **String**, de longitud no definida
- **password** : contraseña de usuario (ver credenciales), obligatorio, tipo **String**, de longitud no definida

Todos los métodos devuelven una estructura informando del éxito o fallo en la invocación del método. Dicha estructura presenta los siguientes miembros:

- **ErrorCode** : Integer, con valores 0 para éxito y distinto de 0 para error (ver apartado de códigos de error)
- **ErrorDescription** : en caso de que ErrorCode sea distinto de 0 informa una descripción del error.
- **Result** : informa el retorno específico al método invocado (ver cada método en detalle)

Ping

Haga uso de este método para consultar la disponibilidad del servicio.

Entrada (parámetros):

- **login**
- **password**

Salida: estructura **PingResult** con los siguientes miembros:

- **ErrorCode**
- **ErrorDescription**
- **Result** : cadena formada por el literal **OK** y la fecha/hora de la invocación en formato **yyyyMMdd HHmmss**, ejemplo **OK - 20120411 213140**.

GetMessageId

Haga uso de este método para obtener un identificador único (GUID) de mensaje (ver métodos de envío de mensajes).

Entrada (parámetros):

- [login](#)
- [password](#)

Salida: estructura [GetMessageIdResult](#) con los siguientes miembros:

- [ErrorCode](#)
- [ErrorDescription](#)
- [Result](#) : GUID como [String](#), ejemplo [6293a9bb-566b-4f43-b087-33afcfec1f58](#).

CreateMessage

Haga uso de este método para enviar un único mensaje.

Entrada (parámetros):

- [login](#)
- [password](#)
- [id](#) : [String](#), identificador GUID del mensaje, opcional, en caso de informarse debe ser un GUID válido, en caso de no hacerlo el API generará uno. Puede hacer uso del método [GetMessageId](#) como utilidad para obtener identificadores GUID válidos.
- [name](#) : [String](#), nombre del envío, opcional, en caso de informarse debe ser inferior a 250 caracteres, en caso de no informarse el API generará un nombre con formato [API_SEND_YYYYMMDDHHMM](#), ejemplo [API_SEND_201204111810](#).
- [sender](#) : [String](#), remitente del mensaje, opcional, en caso de informarse debe tener longitud máxima de 11 caracteres. En caso de no informarse, el API utilizará el remitente pre-definido del usuario (ver Remitente)
- [text](#) : [String](#), texto del mensaje, obligatorio, puede tener hasta 306 caracteres. En caso de tener hasta 160 caracteres resultará en un único SMS, para más de 160 caracteres y hasta 306 resultará en 2 SMS. Se recomienda terminar el texto con “(SMS CERTIFICADO)” para que el receptor del mensaje sea consciente de que es un SMS certificado.
- [mobile](#) : [String](#), destinatario del mensaje, obligatorio, puede informarse con formato internacional (p.e. 0034699999999 ó +34699999999), o en formato nacional (p.e. 699999999).
- [scheduleDate](#) : [String](#), fecha/hora de envío del mensaje, opcional, en caso de informarse debe ser una fecha/hora válida en formato [YYYY-MM-DDThh:mm:ss](#), en caso de no informarse el API utilizará la fecha/hora actual y el envío será inmediato
- [sendEmail](#): [Boolean](#), determina si se envía el acta de certificación por e-mail. Es Opcional. En caso de indicar que sí se quiere enviar el acta, el servicio enviará el acta en formato .PDF a la cuenta de correo principal configurada en la interfaz web. El acta se entregará por e-mail desde la siguiente cuenta de correo: smscertificados@didimo.es. Si no se envían las actas por email, se puede acceder a ellas usando el interfaz Web o con el método [getCertifyfile](#).

Salida: estructura [CreateMessageResult](#) con los siguientes miembros:

- [ErrorCode](#)
- [ErrorDescription](#)
- [WarningMessages](#): estructura [WarningMessagesResult](#) con los siguientes miembros:
 - o [WarningDescription](#): [String](#), Descripción del mensaje de aviso.Los mensajes de avisos se informan cuando no se han realizado correctamente alguna validación y/o proceso, pero que no impiden la terminación completa de la petición.
- [Result](#) : estructura [CreateMessageResultDetail](#) con los siguientes miembros
 - o [Id](#) : [String](#), identificador GUID del mensaje informado por el cliente o generado por el sistema.
 - o [TotalMessages](#) : [Integer](#), número de SMS resultantes a partir del texto.

Ejemplo 1: Se informan datos correctos, sin identificador de mensaje y el resultado es satisfactorio:

Llamada: `servicio.CreateMessage("usuario", "clave", null, "APISend 1 mensaje", "mysender1", "test de 1 mensaje", "607903114", null, false);`

Retorno (serializado como XML):

```
<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <WarningMessages/>
  <Result>
    <Id>9fcbe8d9-3ffe-44ac-b9e2-9d67648041ac</Id>
    <TotalMessages>1</TotalMessages>
  </Result>
</result>
```

Ejemplo 2: Se informan datos correctos, con identificador de mensaje y el resultado es satisfactorio:

Llamada: `servicio.CreateMessage("usuario", "clave", "7F360C11-B655-4EB0-AD54-11BEFA99800A", "APISend 1 mensaje", "mysender2", "test de 1 mensaje", "607903114", null, false);`

Retorno (serializado como XML):

```
<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <WarningMessages/>
  <Result>
    <Id>7F360C11-B655-4EB0-AD54-11BEFA99800A</Id>
    <TotalMessages>1</TotalMessages>
  </Result>
</result>
```

Ejemplo 3: Se informan datos con móvil incorrecto, sin identificador de mensaje y el resultado es de error:

Llamada: `servicio.CreateMessage("usuario", "clave", null, "APISend 1 mensaje", "mysender3", "test de 1 mensaje", "807903114", null, false);`

Retorno (serializado como XML):

```
<result>
  <ErrorCode>3</ErrorCode>
  <ErrorDescription>Error de validación: El móvil informado ('807903114') no es
válido </ErrorDescription>
  <WarningMessages/>
  <Result>
    <Id></Id>
    <TotalMessages></TotalMessages>
  </Result>
</result>
```

Ejemplo 4: Se informan datos correctos y el resultado es satisfactorio, pero se informa de un aviso:

Llamada: `servicio.CreateMessage("usuario", "clave", null, "APISend 1 mensaje", "mysender3", "test de 1 mensaje", "807903114", null, true);`

Retorno (serializado como XML):

```
<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <WarningMessages>
    <WarningDescription>El usuario no tiene informada ninguna cuenta de correo
para enviar el acta</WarningDescription>
  </WarningMessages>
  <Result>
    <Id>7F360C11-B655-4EB0-AD54-11BEFA99800A</Id>
    <TotalMessages>1</TotalMessages>
  </Result>
</result>
```

CreateSend

Haga uso de este método para enviar uno o más mensajes.

Entrada (parámetros):

- **login**
- **password**
- **send** : estructura tipo **Send**, que contiene los siguientes miembros:
 - **Name** : **String**, nombre del envío, opcional, en caso de informarse debe ser inferior a 250 caracteres, en caso de no informarse el API generará un nombre con formato **API_SEND_YYMMDDHHMM**, ejemplo **API_SEND_201204111810**.
 - **Sender** : **String**, remitente del mensaje, opcional, en caso de informarse debe tener longitud máxima de 11 caracteres. En caso de no informarse el API utilizará el remitente pre-definido del usuario (ver Remitente)
 - **ScheduleDate** : **String**, fecha/hora de envío del mensaje, opcional. En caso de informarse debe ser una fecha/hora válida en formato **YYYY-MM-DDThh:mm:ss**, en caso de no informarse el API utilizará la fecha/hora actual y el envío será inmediato
 - **Messages** : lista de estructuras **Message**, conteniendo los datos específicos de cada mensaje, donde cada elemento **Message** contiene los siguientes miembros:
 - **id** : **String**, identificador GUID del mensaje, opcional. En caso de informarse debe ser un GUID válido. En caso de no hacerlo el API generará uno. Puede hacer uso del método **GetMessageId** como utilidad para obtener identificadores GUID válidos.
 - **text** : **String**, texto del mensaje, obligatorio, puede tener hasta 306 caracteres. En caso de tener hasta 160 caracteres resultará en un único SMS, para más de 160 caracteres y hasta 306 resultará en 2 SMS. Se recomienda terminar el texto con "(SMS CERTIFICADO)" para que el receptor del mensaje sea consciente de que es un SMS certificado.
 - **mobile** : **String**, destinatario del mensaje, obligatorio, puede informarse con formato internacional (p.e. 0034699999999 ó +34699999999), o en formato nacional (p.e. 699999999).
- **sendEmail**: **Boolean**, determina si se envía el acta de certificación por e-mail. Es Opcional. En caso de indicar que sí se quiere enviar el acta, el servicio enviará el acta en formato .PDF a la cuenta de correo principal configurada en la interfaz web. El acta se entregará por e-mail desde la siguiente cuenta de correo: smscertificados@didimo.es. Si no se envían las actas por email, se puede acceder a ellas usando el interfaz Web o con el método **getCertifyfile**.

Salida: estructura [CreateSendResult](#) con los siguientes miembros:

- [ErrorCode](#)
- [ErrorDescription](#)
- [WarningMessages](#): estructura [WarningMessagesResult](#) con los siguientes miembros:
 - o [WarningDescription](#): [String](#), Descripción del mensaje de aviso.
Los mensajes de avisos se informan cuando no se han realizado correctamente alguna validación y/o proceso, pero que no impiden la terminación completa de la petición.
- [Result](#) : lista de estructuras [CreateSendResultDetail](#) informando del detalle de resultado para cada mensaje, donde cada [CreateSendResultDetail](#) contiene los siguientes miembros
 - o [ErrorCode](#)
 - o [ErrorDescription](#)
 - o [Id](#) : [String](#), identificador GUID del mensaje informado por el cliente o generado por el sistema.
 - o [TotalMessages](#) : [Integer](#), número de SMS resultantes a partir del texto.

En este método hay 2 niveles de error:

- Global: afecta a todo el envío y es informado por [ErrorCode](#) y [ErrorDescription](#) a nivel de [CreateSendResult](#). En caso de haber error global no se llega a procesar ningún mensaje.
- Detallado: cuando a nivel global no ha habido error y se procesan mensajes, este nivel detalla el resultado para cada uno de los mismos, y es informado por [ErrorCode](#) y [ErrorDescription](#) a nivel de [CreateSendResultDetail](#).

Ejemplos de errores globales y detallados:

- Error Global
 - o se informa un envío con credenciales no válidas y el método devuelve un resultado con [ErrorCode 2](#) (Error de autenticación) en [CreateSendResult](#).
 - o se informa un envío con una fecha/hora no válidas 2020-01-01X09:00:00 y el método devuelve un resultado con [ErrorCode 3](#) (Error de validación: La programación informada no es una fecha/hora válida) en [CreateSendResult](#).
- Error Detallado:
 - o se informa un envío con un mensaje cuyo texto no es válido y el método devuelve un resultado con [ErrorCode 0](#) (no hay error) en [CreateSendResult](#) y con [ErrorCode 3](#) (Error de validación) en [CreateSendResultDetail](#) de dicho mensaje.
 - o se informa un envío con un mensaje cuyo móvil no es válido y el método devuelve un resultado con [ErrorCode 0](#) (no hay error) en [CreateSendResult](#) y con [ErrorCode 3](#) (Error de validación) en [CreateSendResultDetail](#) de dicho mensaje.

GetMessageStatus

Haga uso de este método para obtener el estado de un mensaje y la fecha de finalización que se informa en el acta de certificación.

Entrada (parámetros):

- [login](#)
- [password](#)
- [id](#) : [String](#), identificador GUID del mensaje a consultar su estado

Salida: estructura [GetMessageIdResult](#) con los siguientes miembros:

- [ErrorCode](#)
- [ErrorDescription](#)
- [Result](#) : estructura [GetMessageIdResultDetail](#) con los siguientes miembros

- **StatusCode** : *Integer*, informa un código de estado (ver estados de mensaje).
- **StatusDescription** : *String*, informa una descripción de estado (ver estados de mensaje).
- **FinalizeDate** : *String*, informa la fecha de finalización del mensaje que se refleja en el acta de certificación, en formato dd/MM/yyyy HH:mm:ss. La fecha de finalización se informa vacía si no se ha generado el acta.

Ejemplos de estados (retornos serializados como XML):

```
<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <Result>
    <StatusCode>10</StatusCode>
    <StatusDescription>Pendiente Entrega a Operador</StatusDescription>
    <FinalizeDate></FinalizeDate>
  </Result>
</result>

<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <Result>
    <StatusCode>14</StatusCode>
    <StatusDescription>Pendiente Entrega a Destinatario</StatusDescription>
    <FinalizeDate></FinalizeDate>
  </Result>
</result>

<result>
  <ErrorCode>0</ErrorCode>
  <ErrorDescription></ErrorDescription>
  <Result>
    <StatusCode>19</StatusCode>
    <StatusDescription>Finalizado en Destinatario</StatusDescription>
    <FinalizeDate>13/10/2011 13:20:55</FinalizeDate>
  </Result>
</result>
```

GetCredits

Este método sirve para consultar el número de créditos para enviar SMS.

Entrada (parámetros):

- **login**
- **password**

Salida: estructura **GetCreditsResult** con los siguientes miembros:

- **ErrorCode**
- **ErrorDescription**
- **Credits**: *integer*, Total de créditos disponibles.

GetCertifyFile

Este método permite descargar el acta de certificación correspondiente a un mensaje.

Entrada (parámetros):

- **login**
- **password**
- **id** : **String**, identificador GUID del mensaje que se quiere obtener el acta de certificación.

Salida: **byte array**, el cual el cliente del API debe guardar como PDF.

Los ficheros de acta de mensaje se nombran como <id_mensaje>.pdf, por ejemplo, un mensaje certificado con id **9fcbe8d9-3ffe-44ac-b9e2-9d67648041ac** genera un acta como fichero **9fcbe8d9-3ffe-44ac-b9e2-9d67648041ac.pdf**

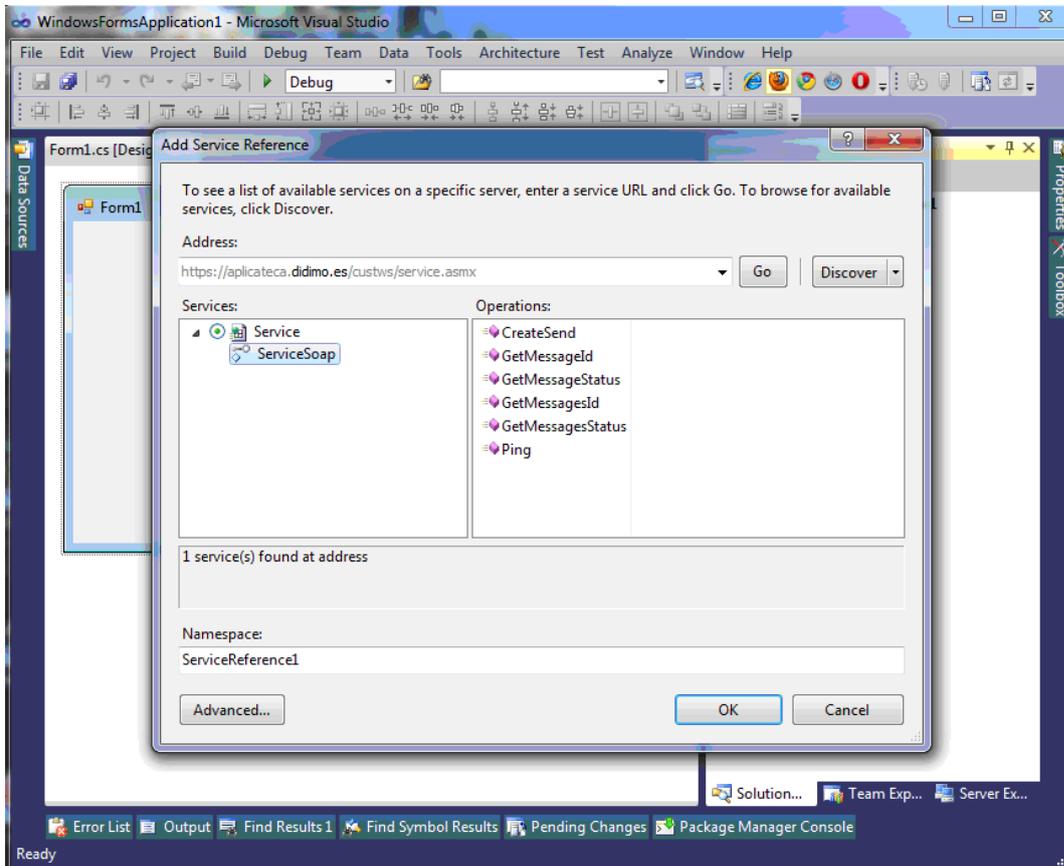
Hay que tener en cuenta que la creación del acta de envío lleva un tiempo desde el momento en que se informa el mensaje. Por ello la descarga de actas de certificación puede generar una excepción cuando la misma no ha sido generada (o el mensaje no es certificado).

Excepciones que se generan por el método “**GetCertifyFile**”:

- Autenticación: “**Error de autenticación:...**”
- Validación: “**Error de validación: El mensaje con el id informado no es un mensaje certificado**”
- Acta no generada: “**Acta Certificada no encontrada: El acta del mensaje para el id informado no se ha generado**”
- Mensaje no encontrado: “**Mensaje no encontrado: No se encontraron mensajes para el id informado**”
- Error desconocido: “**Error desconocido**”. En caso de recibir una excepción de este tipo, póngase en contacto con soporte (soporte.aplicateca@linkmobility.com).

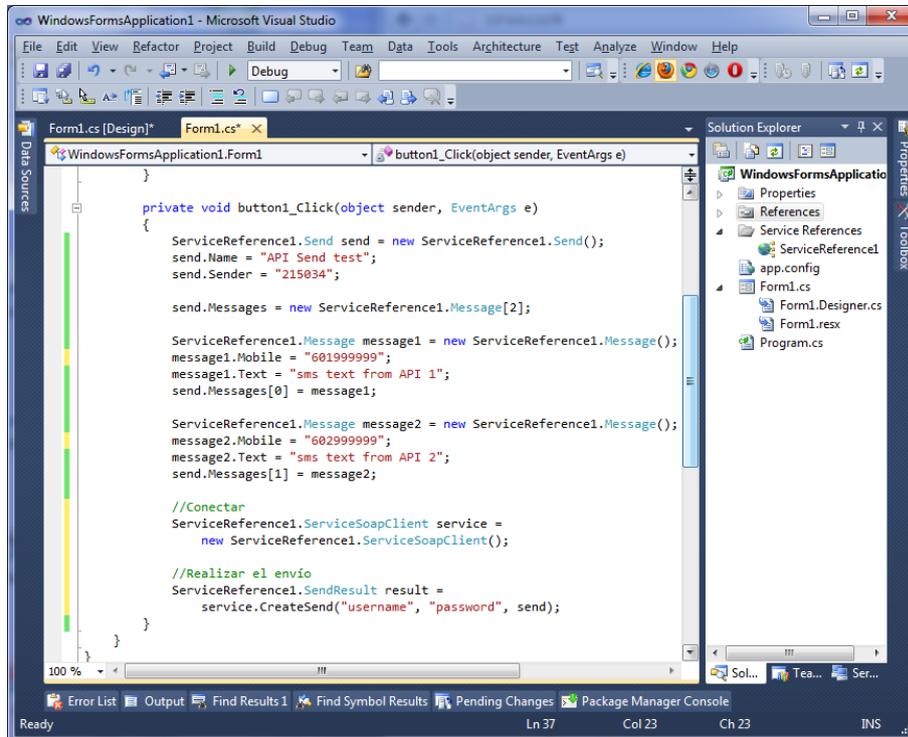
8. Ejemplo de integración

Como ejemplo se muestran capturas de pantalla de una integración usando Microsoft VisualStudio 2010



Importación de métodos 1

Tras la importación de métodos por WSDL se autogenera código para poder utilizar el servicio. Un ejemplo de la realización de un envío sería:



Ejemplo de envío con CreateMessage:

```

string id = new Guid().ToString();
string name = "API Send test";
string sender = "miremitente";
string text = "sms text from API";
string mobile = "601999999";
string scheduleDate = "";
Nullable<Boolean> sendMail = null;

//Conectar
ServiceReference1.ServiceSoapClient service =
    new ServiceReference1.ServiceSoapClient();

//Realizar el envío
ServiceReference1.CreateMessageResult result =
    service.CreateMessage(
        "username", "password"
        , id, name, sender, text, mobile, scheduleDate, sendMail
    );

//Resultado
if(result.ErrorCode != 0)
{
    // acciones ante error
    // ...mostramos error
    Console.WriteLine(String.Format("error: {0} ({1})"
        , result.ErrorCode
        , result.ErrorDescription
    );
}

```

```
        ));  
        // ...otras acciones  
    }  
    else  
    {  
        // acciones ante éxito  
        // ...mostramos id del mensaje y total de SMS  
        Console.WriteLine(String.Format("id: {0}"  
            , result.Result.Id  
        ));  
        Console.WriteLine(String.Format("sms: {0}"  
            , result.Result.TotalMessages  
        ));  
    }  
}
```

Ejemplo de envío con CreateSend:

```
Nullable<Boolean> sendMail = null;  
  
ServiceReference1.Send send = new ServiceReference1.Send();  
send.Name = "API Send test";  
send.Sender = "miremitente";  
  
send.Messages = new ServiceReference1.Message[2];  
  
ServiceReference1.Message message1 = new ServiceReference1.Message();  
message1.Mobile = "601999999";  
message1.Text = "sms text from API 1";  
message1.Id = new Guid().ToString();  
send.Messages[0] = message1;  
  
ServiceReference1.Message message2 = new ServiceReference1.Message();  
message2.Mobile = "602999999";  
message2.Text = "sms text from API 2";  
message2.Id = new Guid().ToString();  
send.Messages[1] = message2;  
  
//Conectar  
ServiceReference1.ServiceSoapClient service =  
    new ServiceReference1.ServiceSoapClient();  
  
//Realizar el envío  
ServiceReference1.SendResult result =  
    service.CreateSend("username", "password", send, sendMail);  
  
//Resultado  
if(result.ErrorCode != 0)  
{  
    // acciones ante error global  
    // ...mostramos error  
    Console.WriteLine(String.Format("error: {0} ({1})"  
        , result.ErrorCode  
        , result.ErrorDescription  
    ));  
    // ...otras acciones  
}
```

```
}
else
{
    // acciones ante éxito global
    // ...mostramos los mensajes
    for (int i = 0; i < result.Result.Length; i++)
    {
        if(result.Result[i].ErrorCode != 0)
        {
            // acciones ante error en el mensaje
        }
        else
        {
            // acciones ante éxito en el mensaje
        }
    }
}
```

Ejemplo de utilización del método GetCertifyFile:

```
byte[] result = this.ws.GetCertifyFile("usuario"
    , "password"
    , "00000000-0000-0000-0000-000000000000");

String filePath = @"C:\";
String fileName = String.Format("{0}.pdf", "00000000-0000-0000-0000-000000000000");

using (FileStream fs = new FileStream(Path.Combine(filePath, fileName),
    FileMode.Create, FileAccess.Write))
{
    fs.Write(result, 0, result.Length);
    fs.Flush();
    fs.Close();
}
```

Nota: Para la prueba de los ejemplos puede utilizarse, entre otros entornos, Microsoft Visual C# Express disponible de forma gratuita en <http://www.microsoft.com/express>

9. Códigos de Error

Errores del API (ErrorCode, ErrorDescription):

- 0 éxito, no tiene descripción
 - Tiene lugar en cualquier método
- 1 "Error desconocido"
 - Puede tener lugar en cualquier método
- 2 "Error de autenticación: ..."
 - Puede tener lugar en cualquier método
- 3 "Error de validación: ..."

- Puede tener lugar en cualquier método donde haya otros parámetros además de las credenciales
- 4 "Error de crédito: ..."
 - Puede tener lugar en los métodos CreateMessage y CreateSend
- 5 "Error al preparar los mensajes: ..."
 - Puede tener lugar en los métodos CreateMessage y CreateSend
- 6 "Error al crear el envío: ..."
 - Puede tener lugar en los métodos CreateMessage y CreateSend
- 7 "Mensaje no encontrado: ..."
 - Puede tener lugar en el método GetMessageStatus

10. Estados de mensaje

CODIGO NUMERICO - DESCRIPCION DE ESTADO

- 1 Pendiente Entrega a Operador
- 3 Entregando a Plataforma
- 4 Finalizado. Entrega a Plataforma Fallida *** estado final ***
- 5 Pendiente Entrega a Plataforma
- 7 Supera Logitud Máxima *** estado final ***
- 8 Pendiente Entrega a Operador
- 9 Pendiente Entrega a Operador
- 12 Pendiente Entrega a Destinatario
- 16 Pendiente Finalización. Entregado a Destinatario
- 17 Finalizado en Operador *** estado final ***
- 19 Finalizado en Destinatario *** estado final ***
- 21 Finalizado. Expirado en Operador *** estado final ***
- 22 Finalizado. Rechazado en Operador *** estado final ***
- 23 Finalizado. Destinatario Incorrecto en Operador *** estado final ***
- 26 Finalizado. Rechazado en Plataforma *** estado final ***
- 27 Finalizado. Rechazado en Plataforma. Crédito Insuficiente *** estado final ***
- 30 Finalizado. Servicio No Disponible *** estado final ***
- 31 Finalizado. Rechazado en Plataforma *** estado final ***
- 32 Finalizado. Rechazado en Plataforma. Mensaje No Válido *** estado final ***
- 33 Cancelado por duplicidad *** estado final ***
- 34 Cancelado por contener caracteres no válidos *** estado final ***
- 36 Cancelado por no válido *** estado final ***

11. Caracteres admitidos

El texto solamente admite caracteres GSM7, aunque algunos caracteres no GSM7 (por ejemplo, vocales acentuadas), son reemplazadas por caracteres equivalentes para que sean admitidos (por ejemplo, *á* es sustituido por *a*).

Puede encontrar información sobre GSM7 en los siguientes enlaces:

http://en.wikipedia.org/wiki/GSM_03.38

<http://www.developershome.com/sms/gsmAlphabet.asp>

<http://www.devindia.net/software/misc/validgsmchar.pdf>

La longitud máxima del texto es de **459 caracteres**, siempre teniendo en cuenta que los caracteres GSM7 Extendidos (^ {} \ ~ [] | €) cuentan como 2 caracteres. Por ejemplo:

- el texto **El carácter (es un paréntesis** ocupa 30 caracteres y se computa por 30
- el texto **El carácter [es un corchete** ocupa 28 caracteres pero se computa por 29

Como se ha mencionado, el texto del mensaje se trata para realizar sustituciones de caracteres no GSM7 de forma que el mensaje sea admisible. Así, determinadas vocales acentuadas (por ejemplo á) son sustituidas por sus versiones sin acento. Y aquellos caracteres no GSM7 que no tienen un equivalente son reemplazados por ? (por ejemplo °).

Reemplazos:

CARACTERES NO GSM7	SUSTITUCIONES
á,â,ã (mayusc. y minusc.)	a
ê,ë,é (mayusc. y minusc.)	e
í,î,ï (mayusc. y minusc.)	i
ô,ó,ò (mayusc. y minusc.)	o
ú,û (mayusc. y minusc.)	u
ý,ÿ (mayusc. y minusc.)	y
ç	Ç
´	´
° a	Estos caracteres se eliminan
otros	?

12. Terminología

MT (Mobile Terminated): Esta terminología es usada para referirse al envío de un mensaje a hacia el Terminal

MO (Mobile Originated): Esta terminología es usada para referirse al envío de un mensaje desde el Terminal.

Push: Se denomina con esta terminología a la acción de enviar un mensaje a un terminal móvil.

Pull: Se denomina con esta terminología a la acción de envío de un mensaje por parte de un Terminal móvil a la plataforma.

Pull – Push: Se denomina con esta terminología a la acción conjunta de envío por parte de un Terminal móvil de un mensaje a la plataforma, y la respuesta de la misma.

13. Anexo



La aplicación cliente del servicio web debe tener en cuenta si la salida a Internet requiere o no de un cortafuegos para implementar en la misma la correspondiente funcionalidad de autenticación.

Para realizar pruebas de desarrollo póngase en contacto con soporte.aplicateca@linkmobility.com