

# Tempos 21

Innovación en Aplicaciones Móviles  
Grupo Atos Origin



Send!

**MANUAL INTEGRACIÓN  
M-DIRECT 2.0**

V1.0

## ÍNDICE

<b>1.- Descripción del servicio M-Direct</b>	<b>3</b>
<b>2.- Especificaciones técnicas</b>	<b>3</b>
2.1.- Descripción de los atributos que se enviarán en cada petición	3
2.2.- SMS de texto	3
2.3.- SMS binario EMS	4
2.4.- Logo NOKIA	4
2.5.- Ringtone NOKIA	4
2.6.- Wap-Push	5
2.7.- Respuestas a la peticiones	5
<b>3.- Tracking via HTTP</b>	<b>7</b>
3.1.- Descripción	7
3.2.- Petición del estado de un mensaje	7
3.3.- Devolución del estado de un mensaje	7
3.3.1.- Ocurre algún error	7
3.3.2.- Acceso correcto	8
<b>4.- Acrónimos</b>	<b>8</b>
<b>5.- Descripción del servicio M-Direct v2.0, Webservice</b>	<b>9</b>
<b>6.- Especificaciones técnicas</b>	<b>9</b>
6.1.- Descripción de los atributos que se enviarán en cada petición	9
6.2.- Cliente	9
6.3.- SMS de texto	10
6.4.- SMS de texto con remitente	10
6.5.- SMS binario EMS	10
6.6.- Logo NOKIA	11
6.7.- Ringtone NOKIA	11
6.8.- Wap-Push	11
6.9.- Obtención estado de mensaje	12
6.10.- Respuestas a la peticiones	12
<b>7.- Estados de los mensajes</b>	<b>12</b>
<b>8.- Acrónimos</b>	<b>12</b>
<b>9.- Anexo</b>	<b>12</b>

## 1.- Descripción del servicio M-Direct

Convierte peticiones remotas de clientes en SMS, EMS, WapPush, Logos Nokia y Ringtones Nokia a los números que el cliente indique de cualquiera de las operadoras españolas existentes: Movistar, Vodafone, Orange, Yoigo y Euskatel dentro del territorio español.

## 2.- Especificaciones técnicas

En esta nueva versión del producto se podrán enviar:

- SMS de texto
- SMS binarios EMS
- Logos NOKIA
- Ringtones NOKIA
- WapPush

Los envíos se realizarán a partir de la llegada de una petición **https con certificado digital cliente**, ya sea GET o POST (**se recomienda utilizar POST**). Si el envío se ha efectuado correctamente se devuelve un identificador de la transacción al cliente. Mediante este identificador el cliente puede posteriormente hacer un seguimiento de los mensajes. El período de vida de los identificadores es de 5 días. Si se produce un error se enviará un texto con el código de error. Ver apartado 2.7.-

### 2.1.- Descripción de los atributos que se enviarán en cada petición

En cada petición de envío se deberán incluir los siguientes atributos:

- user: nombre de usuario (dato proporcionado por Tempos21)
- company: nombre de la empresa en la plataforma de mensajería (dato proporcionado por Tempos21)
- passwd: password para el user anterior. (dato proporcionado por Tempos21)
- gsm: teléfono destino sin prefijo '+34'.
- type: tipo del envío. Podrá ser:
  - sms
  - bin
  - otalogo
  - ringtone
  - wappush

### 2.2.- SMS de texto

Mensajes de texto de hasta 160 caracteres. Solo se podrán enviar aquellos caracteres soportados por el del estándar GSM.

Atributo type: su valor será sms.

Ejemplo de url para el envío:

<https://sms.tempos21.com/mdirect/MDirectT21?user=Usuario&company=Empresa&passwd=Contraseña&gsm=GSM&type=sms&msg=MENSAJE>

### 2.3.- SMS binario EMS

Mensajes de binarios codificados en hexadecimal. En cada mensaje se podrán enviar hasta un máximo de 140 bytes, que codificados en notación hexadecimal serán como máximo 280 caracteres. Será responsabilidad del cliente enviar una trama binaria correcta.

Atributo type: tipo del envío; su valor será bin.

Ejemplo de url para el envío (se indica en negrita el parámetro anterior):

[https://sms.tempos21.com/mdirect/MDirectT21?user=Usuario&company=Empresa&passwd=Contraseña&gsm=GSM&type=\*\*bin\*\*&bin=820C8000424541543A3230300D0A4D454C4F44593A23633223633472342A3362322A33613472342A3361322A33623323633372332A3362312E72312E2363322E2A3362342E72352A3362322A3361332E72332E2363322A3365312A33236631723172332363322E23633472342A3362332E72342A332366332A3361322A336233236333](https://sms.tempos21.com/mdirect/MDirectT21?user=Usuario&company=Empresa&passwd=Contraseña&gsm=GSM&type=<b>bin</b>&bin=820C8000424541543A3230300D0A4D454C4F44593A23633223633472342A3362322A33613472342A3361322A33623323633372332A3362312E72312E2363322E2A3362342E72352A3362322A3361332E72332E2363322A3365312A33236631723172332363322E23633472342A3362332E72342A332366332A3361322A336233236333)

### 2.4.- Logo NOKIA

Se permitirán enviar logos de operadora. Este es un caso especial de envío binario.

Será responsabilidad del cliente enviar una trama binaria correcta.

Atributo type: su valor será otalogo.

Ademas de los atributos comunes se deberá añadir el siguiente atributo:

- oper: código de operador; es necesario para el caso de un logo para vincularlo a un operador. Los valores actuales posibles, que son configurables por el administrador del producto, son:
  - “3431” para Orange
  - “3421” para Vodafone
  - “3411” para Telefónica Móviles

Ejemplo de url para el envío:

[https://sms.tempos21.com/mdirect/MDirectT21?user=Usuario&company=Empresa&passwd=Contraseña&gsm=GSM&type=\*\*otalogo\*\*&oper=xxx&bin=820C8000424541543A3230300D0A4D454C4F44593A23633223633472342A3362322A33613472342A3361322A33623323633372332A3362312E72312E2363322E2A3362342E72352A3362322A3361332E72332E2363322A3365312A33236631723172332363322E23633472342A3362332E72342A332366332A3361322A336233236333](https://sms.tempos21.com/mdirect/MDirectT21?user=Usuario&company=Empresa&passwd=Contraseña&gsm=GSM&type=<b>otalogo</b>&oper=xxx&bin=820C8000424541543A3230300D0A4D454C4F44593A23633223633472342A3362322A33613472342A3361322A33623323633372332A3362312E72312E2363322E2A3362342E72352A3362322A3361332E72332E2363322A3365312A33236631723172332363322E23633472342A3362332E72342A332366332A3361322A336233236333)

### 2.5.- Ringtone NOKIA

Se permitirán enviar ringtones en formato NOKIA. Este es un caso especial de envío binario.

Será responsabilidad del cliente enviar una trama binaria correcta.

Atributo type: su valor será ringtone.

Ejemplo de url para el envío:

[https://sms.tempos21.com/mdirect/MDirectT21?user=Usuario&company=Empresa&passwd=Contraseña&gsm=GSM&type=\*\*ringtone\*\*&bin=820C8000424541543A3230300D0A4D454C4F44593A23633223633472342A3362322A33613472342A3361322A33623323633372332A3362312E72312E2363322E2A3362342E72352A3362322A3361332E72332E2363322A3365312A33236631723172332363322E23633472342A3362332E72342A332366332A3361322A336233236333](https://sms.tempos21.com/mdirect/MDirectT21?user=Usuario&company=Empresa&passwd=Contraseña&gsm=GSM&type=<b>ringtone</b>&bin=820C8000424541543A3230300D0A4D454C4F44593A23633223633472342A3362322A33613472342A3361322A33623323633372332A3362312E72312E2363322E2A3362342E72352A3362322A3361332E72332E2363322A3365312A33236631723172332363322E23633472342A3362332E72342A332366332A3361322A336233236333)

## 2.6.- Wap-Push

Realizará el envío de Wap-push.

Atributo type: su valor será wappush

Los atributos específicos para este tipo de mensaje son:

- url: url del del site WAP. Se permite que tenga los siguientes prefijos:
  - o Ejemplo 1: <http://url...>
  - o Ejemplo 2: <https://url...>
  - o Ejemplo 3: url... (en este último caso, se convertirá automáticamente a <http://url...>)
- msg: Texto que aparecerá en la llegada del Wap-push

El tamaño máximo de un mensaje Wappush será de 99 caracteres, repartidos entre el texto y la url y sin tener en cuenta los siguientes prefijos o sufijos:

- http://
- https://
- <http://www.>
- <https://www.>
- .com/
- .net/
- .org/
- .edu/

Ejemplo de url para el envío:

<https://sms.tempos21.com/mdirect/MDirectT21?user=Usuario&company=EMPRESA&passwd=CONTRASEÑA&gsm=TEL&type=wappush&url=wap.google.com&msg=TEXTO>

## 2.7.- Respuestas a la peticiones

El sistema devuelve un identificador de la transacción, si la inserción de la transacción al sistema se ha realizado correctamente.

Ejemplo de respuesta correcta (ejemplo de identificador):

[ OK | ] 5118302091224821

*"[ ]" indica opcional. Aparecerá el prefijo "OK |" para: 'logo', 'ringtone' y 'wappush'.*

En caso de error el sistema puede retornar lo siguiente:

- **ERROR - Authentication failed | <mensaje\_explicativo\_del\_error>**  
Parámetros de autenticación no válidos.
- **ERROR Plugin Authentication not valid| <mensaje\_explicativo\_del\_error>**  
Está utilizando una modalidad de M-Direct Plus a la que no tiene permisos acceder.
- **ERROR Invalid type [ type\_name ]**  
Tipo de mensaje no válido, vacío o no enviado.
- **ERROR No Contract For Iduser [idUser] Idproduct [idProduct]**

Error de provisión. Si se produce este error póngase en contacto con el servicio de Atención al Cliente de Tempos21

- **ERROR - Error sending message** | <mensaje\_explicativo\_del\_error>  
Se ha producido un error genérico en el envío del mensaje. Si se produce este error póngase en contacto con el servicio de Atención al Cliente de Tempos21.
- **ERROR - Message too large** | <mensaje\_explicativo\_del\_error>  
El mensaje enviado supera el tamaño máximo permitido.
- **ERROR - Invalid parameters** | <mensaje\_explicativo\_del\_error>  
Faltan parámetros en la petición.
- **ERROR - Request timeout:** La petición no ha podido acceder al servicio y ha vencido el tiempo establecido de espera.

### 3.- Tracking via HTTP

#### 3.1.- Descripción

El producto de “tracking vía http” permite saber si un mensaje se ha entregado al terminal móvil destinatario o en su defecto conocer en que punto de su camino hacia el terminal se encuentra.

Puede emplearse con los identificadores de mensaje únicos (UIM) de seguimiento de mensajes retornados por diferentes productos de la plataforma como por ejemplo M-DIRECT.

#### 3.2.- Petición del estado de un mensaje

La URL de la llamada es del tipo: `http://.../limsp/UIMState/index.jsp`

Los parámetros validos de la URL son:

*company*: Nombre de la empresa que ha generado el mensaje.

*user*: Usuario de la empresa que accede a la información.

*passwd*: Contraña del usuario en la plataforma.

*uim*: Identificador único del mensaje en la plataforma de la cual se pretende consultar el estado.

Parámetro opcional:

*info*: Si se utiliza este parámetro su valor debe ser “full”, para utilizar la modalidad extendida del servicio UIMState.

#### Ejemplos:

Llamada normal:

`https://sms.tempos21.com/limspssl/UIMState/index.jsp?company=EMPRESA&user=USUARIO&passwd=CONTRASEÑA&uim=<NUM_UIM>`

Llamada en modo extendido:

`https://sms.tempos21.com/limsp/UIMState/index.jsp?company=empresa&user=usuario&passwd=contraseña&uim=<NUM_UIM>&info=full`

#### 3.3.- Devolución del estado de un mensaje

Una vez realizada correctamente la petición de estado al servicio UIMState pueden darse dos casos.

##### 3.3.1.- *Ocurre algún error.*

En caso de que la petición al servicio generé algún error, este devolverá el resultado con la sintaxis:

[ ERROR|<code>| ] Description

Código de Error	Significado
ERROR -5 Authentication failure	Alguna de las credenciales ( <i>company</i> , <i>user</i> ó <i>passwd</i> ) no es correcta
ERROR -10 UIM not found	No existe el UIM o por algún motivo la plataforma aún no lo ha generado.

### 3.3.2.- Acceso correcto.

En este caso, cuando la información es accesible, se contemplan dos modos de funcionamiento.

Respuesta sin error Modo normal:

Devolverá los siguientes códigos correspondientes. Se utilizará la sintaxis: **OK|<code>|<reason\_code>**

Estado	Descripción	Tipo
<b>OK 00 PENDIENTE DE SER PROCESADO POR LA PLATAFORMA</b>	La plataforma de Tempos 21 aún no ha entregado el mensaje a la operadora.	TEMPORAL NO FACTURABLE
<b>OK 10 ENTREGADO A LA OPERADORA NO CONFIRMADO</b>	Procesado por la plataforma, sin confirmación de la Operadora	TEMPORAL NO FACTURABLE
<b>OK 20 ALMACENADO EN LA OPERADORA</b>	El terminal móvil al que se le envía el SMS está apagado o fuera de cobertura. Almacenado en la operadora a la espera de la entrega al usuario	TEMPORAL FACTURABLE
<b>OK 30 ENTREGADO A LA OPERADORA</b>	El mensaje ha sido entregado a la operadora pero la operadora no proporciona confirmación de entrega al usuario.	FINAL FACTURABLE
<b>OK 40 ENTREGADO AL USUARIO</b>	El mensaje ha sido entregado al usuario	FINAL FACTURABLE
<b>OK 50 RECHAZADO TRAMA INCORRECTA</b>	El SMS es rechazado por la Operadora debido a que el formato de la petición de envío no es correcta	FINAL FACTURABLE
<b>OK 60 ELIMINADO EN LA OPERADORA</b>	El SMS es eliminado ante la imposibilidad de la entrega al usuario	FINAL FACTURABLE
<b>OK 70 RECHAZADO EN LA OPERADORA</b>	SMS rechazado por la operadora al no existir el número de teléfono indicado	FINAL FACTURABLE
<b>OK 80 RECHAZADO POR FALTA DE CONFIRMACIÓN DE LA OPERADORA</b>	SMS Rechazados ante problemas de conectividad y confirmación de la operadora	FINAL NO FACTURABLE
<b>OK 90 RECHAZADO POR LA PLATAFORMA</b>	SMS Rechazado por la plataforma de Tempos 21 debido a que alguno de los parámetros es incorrecto	FINAL NO FACTURABLE
<b>OK 100 RECHAZADO POR FALTA DE CREDITO</b>	No tiene crédito suficiente para realizar envíos a través de la plataforma de Tempos 21	FINAL NO FACTURABLE
<b>OK 110 RECHAZADO INTERFICIE OPERADORA</b>	SMS Rechazados ante problemas de conectividad de la Operadora	FINAL NO FACTURABLE

## 4.- Acrónimos

- SMS: Short Message System.
- EMS: Enhanced Messaging Service.
- WAP-PUSH. Envío de un mensaje que desencadena una conexión WAP (Wireless Application Protocol) a una url.

## 5.- Descripción del servicio M-Direct v2.0, WebService

Convierte peticiones remotas de clientes en SMS, SMS con remitente a los números que el cliente indique de cualquiera de las operadoras españolas existentes: Movistar, Vodafone, Orange, Yoigo y Euskatel, dentro del territorio español.

## 6.- Especificaciones técnicas

Accediendo vía WebService, en esta nueva versión del producto se podrán enviar:

- SMS de texto
- SMS de texto con remitente
- SMS binarios EMS
- Logos NOKIA
- Ringtones NOKIA
- WapPush

Los envíos se realizarán a partir de la llegada de una llamada desde un cliente Axis o compatible. Existirá un servidor creado con Axis en base a un interface (ver [Anexo](#)). A partir del mismo se habrá generado un WSDL (ver [Anexo](#)), que habrá de ser cumplido por el cliente.

Si el envío se ha efectuado correctamente se devuelve un identificador de la transacción al cliente. Mediante este identificador el cliente puede posteriormente hacer un seguimiento de los mensajes. El período de vida de los identificadores es de 5 días. Si se produce un error se enviará un texto con el código de error. Ver apartado 2.7.-

Además, se podrá obtener el estado de un mensaje en base a su UIM.

### 6.1.- Descripción de los atributos que se enviarán en cada petición

En cada petición de envío (no en petición de estado) se deberán incluir los siguientes atributos:

- user: nombre de usuario (dato proporcionado por Tempos21)
- company: nombre de la empresa en la plataforma de mensajería (dato proporcionado por Tempos21)
- passwd: contraseña para el usuario anterior. (dato proporcionado por Tempos21)
- gsm: teléfono destino sin prefijo '+34'.

### 6.2.- Cliente

Se muestra un ejemplo de cliente que accedería a un supuesto servidor MDirect existente:

```
package com.tempos21.mdirect.ws.client;

public class MDirectWSClient
{
    protected final static String wsURL = "http://10.1.2.30:8280/mdirect2ws/services/MDirectWS";
}
```

```

private String endpointAddress;

public static void main(String[] args)
{
    System.out.println("Endpoint address = " + wsURL);

    try
    {
        MDirectWSServiceLocator a = new MDirectWSServiceLocator();
        MDirectWS b = a.getMDirectWS(new java.net.URL(wsURL));
        String res = "UNDEF";

        res = b.sendSms("tempos4", "tempos21", "tempos21", "249984170", "soyeltexto");
        System.out.println("res [" + res + "]");

    }
    catch (Exception ex)
    {
        ex.printStackTrace();
    }
}
}

```

### 6.3.- SMS de texto

Mensajes de texto de hasta 160 caracteres. Solo se podrán enviar aquellos caracteres soportados por el del estándar GSM.

Ejemplo de llamada:

```
res = b.sendSms("tempos4", "tempos21", "tempos21", "646664670", "soyeltexto");
```

### 6.4.- SMS de texto con remitente

Mensajes de texto de hasta 160 caracteres. Solo se podrán enviar aquellos caracteres soportados por el del estándar GSM. Se especifica el texto del remitente.

Ejemplo de llamada:

```
res = b.sendSmsWithSender("t4", "tempos21", "tempos21", "656760665", "soyeltexto",
"soyelsender");
```

### 6.5.- SMS binario EMS

Mensajes de binarios codificados en hexadecimal. En cada mensaje se podrán enviar hasta un máximo de 140 bytes, que codificados en notación hexadecimal serán como máximo 280 caracteres. Será responsabilidad del cliente enviar una trama binaria correcta.

Ejemplo de llamada:

```
res = b.sendBinary("tempos4", "tempos21", "tempos21", "646964670",
"605040B8423F09E060A03AE81EAAF828DC1B48701056A0045C60C037761702E676F6F676C652E636F6D0011035761704944455
F53492F30000103476F6F676C652062792054656D706F733231000101");
```

## 6.6.- Logo NOKIA

Se permitirán enviar logos de operadora. Este es un caso especial de envío binario. Será responsabilidad del cliente enviar una trama binaria correcta.

Además de los atributos comunes se deberá añadir el siguiente atributo:

- oper: código de operador; es necesario para el caso de un logo para vincularlo a un operador. Los valores actuales posibles, que son configurables por el administrador del producto, son:
  - "3431" para Orange
  - "3421" para Vodafone
  - "3411" para Telefónica Móviles

Ejemplo de llamada:

```
res = b.sendLogoNokia("tempos4", "tempos21", "tempos21", "646964670", "<MENSAJE_BINARIO>", "3411");
```

## 6.7.- Ringtone NOKIA

Se permitirán enviar ringtones en formato NOKIA. Este es un caso especial de envío binario. Será responsabilidad del cliente enviar una trama binaria correcta.

Ejemplo de llamada:

```
res = b.sendRingtoneNokia("tempos4", "tempos21", "tempos21", "649684670", "<MENSAJE_BINARIO>");
```

## 6.8.- Wap-Push

Realizará el envío de Wap-push.

Los atributos específicos para este tipo de mensaje son:

- url: url del del site WAP. Se permite que tenga los siguientes prefijos:
  - Ejemplo 1: <http://url...>
  - Ejemplo 2: <https://url...>
  - Ejemplo 3: url... (en este último caso, se convertirá automáticamente a <http://url...>)
- msg: Texto que aparecerá en la llegada del Wap-push

El tamaño máximo de un mensaje Wappush será de 99 caracteres, repartidos entre el texto y la url y sin tener en cuenta los siguientes prefijos o sufijos:

- <http://>
- <https://>
- [http://www.](http://www)
- [https://www.](https://www)
- [.com/](http://www)
- [.net/](http://www)
- [.org/](http://www)
- [.edu/](http://www)

Ejemplo de llamada:

```
res = b.sendWapPush("tempos4", "tempos21", "tempos21", "647984170", "soyeltexto", "www.yahoo.com");
```

## 6.9.- Obtención estado de mensaje

En este caso, los parámetros de la petición serán:

- user: nombre de usuario (dato proporcionado por Tempos21)
- company: nombre de la empresa en la plataforma de mensajería (dato proporcionado por Tempos21)
- passwd: contraseña para el usuario anterior. (dato proporcionado por Tempos21)
- UIM: el identificador del mensaje.

Ejemplo de llamada:

```
res = b. getStatus("tempos4", "tempos21", "tempos21", "6180141772312828");
```

## 6.10.- Respuestas a la peticiones

Ver Manual de Integración M-DirectT21.

## 7.- Estados de los mensajes

Ver Manual de Integración M-DirectT21.

## 8.- Acrónimos

- SMS: Short Message System.
- EMS: Enhanced Messaging Service.
- WAP-PUSH. Mensaje que en su selección desencadena una conexión WAP (Wireless Application Protocol) mediante una url.

## 9.- Anexo

### Interface

```
package com.tempos21.mdirect.ws;

public interface MDirectWS extends java.rmi.Remote
{
    /**
     * Sends a SMS
     * @param user The user
     * @param pwd The password
     * @param company The company
     * @param mobile Destination mobile number
     * @param txt Text to send in the sms
     * @return If there is no error OK | 5118302091224821 <BR>
     *         If there is an error:<br>
     *         ERROR - Authentication failed <BR>
     *         ERROR - Missing Contract <BR>
     *         ERROR - Error sending message <BR>
     *         ERROR - Message too large<BR>
     *         ERROR - Invalid parameters<BR>
     */
}
```

```

public String sendSms(String user, String pwd, String company,
                        String mobile, String txt)
                        throws java.rmi.RemoteException;

/**
 * Sends a SMS changin the sender number
 * @param user The user
 * @param pwd The password
 * @param company The company
 * @param mobile Destination mobile number
 * @param txt Text to send in the sms
 * @param sender Text/Number to send as sender
 * @return If there is no error OK | 5118302091224821 <BR>
 *         If there is an error:<br>
 *         ERROR - Authentication failed <BR>
 *         ERROR - Missing Contract <BR>
 *         ERROR - Error sending message <BR>
 *         ERROR - Message too large<BR>
 *         ERROR - Invalid parameters<BR>
 */
public String sendSmsWithSender(String user, String pwd, String company,
                                String mobile, String txt, String sender)
                                throws java.rmi.RemoteException;

/**
 * Sends a binary SMS
 * @param user The user
 * @param pwd The password
 * @param company The company
 * @param mobile Destination mobile number
 * @param bin
 * @return If there is no error OK | 5118302091224821 <BR>
 *         If there is an error:<br>
 *         ERROR - Authentication failed <BR>
 *         ERROR - Missing Contract <BR>
 *         ERROR - Error sending message <BR>
 *         ERROR - Message too large<BR>
 *         ERROR - Invalid parameters<BR>
 */
public String sendBinary(String user, String pwd, String company,
                           String mobile, String bin)
throws java.rmi.RemoteException;

/**
 * Sends Nokia Logo
 * @param user The user
 * @param pwd The password
 * @param company The company
 * @param mobile Destination mobile number
 * @param logo. Logo binary string in hexadecimal
 * @param carrier
 * @return If there is no error OK | 5118302091224821 <BR>
 *         If there is an error:<br>
 *         ERROR - Authentication failed <BR>
 *         ERROR - Missing Contract <BR>
 *         ERROR - Error sending message <BR>
 *         ERROR - Message too large<BR>
 *         ERROR - Invalid parameters<BR>
 */
public String sendLogoNokia(String user, String pwd, String company,
                              String mobile, String logo, String carrier)
throws java.rmi.RemoteException;

/**
 * Sends Nokia ringtone
 * @param user The user
 * @param pwd The password
 * @param company The company
 * @param mobile Destination mobile number
 * @param ringtone binary string in hexadecimal
 * @return If there is no error OK | 5118302091224821 <BR>
 *         If there is an error:<br>
 *         ERROR - Authentication failed <BR>
 */

```

```

*         ERROR - Missing Contract <BR>
*         ERROR - Error sending message <BR>
*         ERROR - Message too large<BR>
*         ERROR - Invalid parameters<BR>
*/
    public String sendRingtoneNokia(String user, String pwd, String company,
                                   String mobile, String ringtone)
throws java.rmi.RemoteException;

/**
 * Sends a Wap-push
 * @param user The user
 * @param pwd The password
 * @param company The company
 * @param mobile Destination mobile number
 * @param txt Text to send in the Wap-push
 * @param url WAP Url(without http://) Ex. wap.google.com.
 * txt+url length can not be greater than 71 chars
 * @return If there is no error OK | 5118302091224821 <BR>
 *         If there is an error:<br>
 *         ERROR - Authentication failed <BR>
 *         ERROR - Missing Contract <BR>
 *         ERROR - Error sending message <BR>
 *         ERROR - Message too large<BR>
 *         ERROR - Invalid parameters<BR>
 */
    public String sendWapPush(String user, String pwd, String company,
                              String mobile, String txt, String url)
        throws java.rmi.RemoteException;

/**
 * Gets a message status
 * @param user The user
 * @param pwd The password
 * @param company The company
 * @param UIM The Message identifier
 *
 * @return If there is no error:<br>
 *         OK|00|PENDING_LIMSP
 *         OK|10|SENT_LIMSP
 *         OK|20|MOBILE_OFF
 *         OK|30|DELIVERED_CARRIER
 *         OK|40|DELIVERED_USER
 *         OK|100|GEN_ERROR
 *         OK|-1|UNKNOW_STATE
 *
 *         If there is an error:<br>
 *         ERROR|-5|Authentication failure<BR>
 *         ERROR|-10|UIM not found<BR>
 *
 * @throws java.rmi.RemoteException
 */
    public String getStatus(String user, String pwd, String company, String UIM)
        throws java.rmi.RemoteException;
}

```

## WSDL

```

<?xml version="1.0" encoding="UTF-8" ?>
<= <wsdl:definitions targetNamespace="urn:MDirect"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:apachesoap="http://xml.apache.org/xml-soap"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:intf="urn:MDirect"
    xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/" xmlns:impl="urn:MDirect"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
<= <wsdl:message name="sendSmsResponse">
    <wsdl:part name="sendSmsReturn" type="xsd:string" />
</wsdl:message>

```

```

- <wsdl:message name="sendSmsWithSenderRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="xsd:string" />
  <wsdl:part name="in5" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendLogoNokiaResponse">
  <wsdl:part name="sendLogoNokiaReturn" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendLogoNokiaRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="xsd:string" />
  <wsdl:part name="in5" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendBinaryRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="getStatusRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendWapPushRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="xsd:string" />
  <wsdl:part name="in5" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendRingtoneNokiaRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendBinaryResponse">
  <wsdl:part name="sendBinaryReturn" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="getStatusResponse">
  <wsdl:part name="getStatusReturn" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendWapPushResponse">
  <wsdl:part name="sendWapPushReturn" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendRingtoneNokiaResponse">
  <wsdl:part name="sendRingtoneNokiaReturn" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendSmsRequest">
  <wsdl:part name="in0" type="xsd:string" />
  <wsdl:part name="in1" type="xsd:string" />
  <wsdl:part name="in2" type="xsd:string" />
  <wsdl:part name="in3" type="xsd:string" />
  <wsdl:part name="in4" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="sendSmsWithSenderResponse">
  <wsdl:part name="sendSmsWithSenderReturn" type="xsd:string" />
</wsdl:message>
- <wsdl:portType name="MDirectWS">
  - <wsdl:operation name="sendSms" parameterOrder="in0 in1 in2 in3 in4">
    <wsdl:input name="sendSmsRequest" message="intf:sendSmsRequest" />
    <wsdl:output name="sendSmsResponse" message="intf:sendSmsResponse" />
  
```

```

    </wsdl:operation>
- <wsdl:operation name="sendSmsWithSender" parameterOrder="in0 in1 in2 in3 in4 in5">
    <wsdl:input name="sendSmsWithSenderRequest"
        message="intf:sendSmsWithSenderRequest" />
    <wsdl:output name="sendSmsWithSenderResponse"
        message="intf:sendSmsWithSenderResponse" />
    </wsdl:operation>
- <wsdl:operation name="sendBinary" parameterOrder="in0 in1 in2 in3 in4">
    <wsdl:input name="sendBinaryRequest" message="intf:sendBinaryRequest" />
    <wsdl:output name="sendBinaryResponse" message="intf:sendBinaryResponse" />
    </wsdl:operation>
- <wsdl:operation name="sendLogoNokia" parameterOrder="in0 in1 in2 in3 in4 in5">
    <wsdl:input name="sendLogoNokiaRequest" message="intf:sendLogoNokiaRequest" />
    <wsdl:output name="sendLogoNokiaResponse" message="intf:sendLogoNokiaResponse"
        />
    </wsdl:operation>
- <wsdl:operation name="sendRingtoneNokia" parameterOrder="in0 in1 in2 in3 in4">
    <wsdl:input name="sendRingtoneNokiaRequest"
        message="intf:sendRingtoneNokiaRequest" />
    <wsdl:output name="sendRingtoneNokiaResponse"
        message="intf:sendRingtoneNokiaResponse" />
    </wsdl:operation>
- <wsdl:operation name="sendWapPush" parameterOrder="in0 in1 in2 in3 in4 in5">
    <wsdl:input name="sendWapPushRequest" message="intf:sendWapPushRequest" />
    <wsdl:output name="sendWapPushResponse" message="intf:sendWapPushResponse" />
    </wsdl:operation>
- <wsdl:operation name="getStatus" parameterOrder="in0 in1 in2 in3">
    <wsdl:input name="getStatusRequest" message="intf:getStatusRequest" />
    <wsdl:output name="getStatusResponse" message="intf:getStatusResponse" />
    </wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="MDirectWSSoapBinding" type="intf:MDirectWS">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="sendSms">
    <wsdlsoap:operation soapAction="" />
- <wsdl:input name="sendSmsRequest">
    <wsdlsoap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:MDirect" />
    </wsdl:input>
- <wsdl:output name="sendSmsResponse">
    <wsdlsoap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:MDirect" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="sendSmsWithSender">
    <wsdlsoap:operation soapAction="" />
- <wsdl:input name="sendSmsWithSenderRequest">
    <wsdlsoap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:MDirect" />
    </wsdl:input>
- <wsdl:output name="sendSmsWithSenderResponse">
    <wsdlsoap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:MDirect" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="sendBinary">
    <wsdlsoap:operation soapAction="" />
- <wsdl:input name="sendBinaryRequest">
    <wsdlsoap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:MDirect" />
    </wsdl:input>
- <wsdl:output name="sendBinaryResponse">
    <wsdlsoap:body use="encoded"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:MDirect" />
    </wsdl:output>
    </wsdl:operation>
- <wsdl:operation name="sendLogoNokia">

```

```

        <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="sendLogoNokiaRequest">
        <wsdlsoap:body use="encoded"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:MDirect" />
    </wsdl:input>
    - <wsdl:output name="sendLogoNokiaResponse">
        <wsdlsoap:body use="encoded"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:MDirect" />
    </wsdl:output>
</wsdl:operation>
- <wsdl:operation name="sendRingtoneNokia">
    <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="sendRingtoneNokiaRequest">
        <wsdlsoap:body use="encoded"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:MDirect" />
    </wsdl:input>
    - <wsdl:output name="sendRingtoneNokiaResponse">
        <wsdlsoap:body use="encoded"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:MDirect" />
    </wsdl:output>
</wsdl:operation>
- <wsdl:operation name="sendWapPush">
    <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="sendWapPushRequest">
        <wsdlsoap:body use="encoded"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:MDirect" />
    </wsdl:input>
    - <wsdl:output name="sendWapPushResponse">
        <wsdlsoap:body use="encoded"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:MDirect" />
    </wsdl:output>
</wsdl:operation>
- <wsdl:operation name="getStatus">
    <wsdlsoap:operation soapAction="" />
    - <wsdl:input name="getStatusRequest">
        <wsdlsoap:body use="encoded"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:MDirect" />
    </wsdl:input>
    - <wsdl:output name="getStatusResponse">
        <wsdlsoap:body use="encoded"
            encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="urn:MDirect" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="MDirectWSService">
    - <wsdl:port name="MDirectWS" binding="intf:MDirectWSSoapBinding">
        <wsdlsoap:address
            location="https://sms.tempos21.com/mdirectws/services/mdirect2ws/services/MDi
            rectWS" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```