



API – Validador de móviles Certificados SMS



Contenido

1. Servicios SMS	2
2. Créditos.....	2
3. Credenciales	2
4. Funcionalidades	2
5. Métodos.....	2
6. Códigos de Error	5
7. Estados de móviles	5
8. Recomendaciones.....	6
9. Anexo	6

1. Servicios SMS

Antes de utilizar el API “Validador de Móviles” de Certificados SMS, 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 el API, debe integrar un cliente con el siguiente servicio web:

<https://certificadosms.didimo.es/mobilevalidate/Service.asmx>

2. Créditos

El servicio Validador de móviles está integrado en Certificados SMS y por tanto, consume créditos de Certificados SMS. Cada 20 validaciones de móviles se consume 1 crédito. Si se realizan menos de 20 validaciones, se descuenta 1 crédito y se “guardan” las validaciones sobrantes para las próximas validaciones.

Si al validar, se recibe algún [código de error](#), no se descuenta ningún crédito.

3. 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.

4. Funcionalidades

El API ofrece las siguientes funcionalidades:

- Consulta de disponibilidad del servicio (método [Ping](#))
- Obtención de identificador único (GUID) de petición (método [GetRequestId](#))
- Envío de una petición de validación con N móviles (método [CreateRequest](#))
- Obtención del estado de N móviles (método [GetMobileStatus](#))

5. Métodos

El conjunto de métodos es el siguiente:

Service

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

- [CreateRequest](#)
- [GetMobileStatus](#)
- [GetRequestId](#)
- [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**.

GetRequestId

Haga uso de este método para obtener un identificador único (GUID) de una petición de validación

Entrada (parámetros):

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

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

- **ErrorCode**
- **ErrorDescription**
- **Result** : GUID como **String**, ejemplo **6293a9bb-566b-4f43-b087-33afcfec1f58**.

CreateRequest

Haga uso de este método para realizar una petición de validación de uno o más móviles.

Entrada (parámetros):

- **login**
- **password**
- **request** : estructura tipo **Request**, que contiene los siguientes miembros:
 - **Id** : **String**, identificador GUID de la petición, opcional. En caso de informarse debe ser un GUID válido. En caso de no hacerlo el API generará uno. Puede hacer uso del método **GetRequestId** como utilidad para obtener identificadores GUID válidos
 - **Mobiles** : lista del tipo **String**, contiene los móviles que se quieran validar, obligatorio, puede informarse con formato internacional (p.e. 003469999999 ó +3469999999), o en formato nacional (p.e. 699999999).

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

- **ErrorCode**
- **ErrorDescription**
- **Result** estructura tipo **CreateRequestResultDetail** informando del detalle de resultado de la petición, que contiene los siguientes miembros:
 - **Id** : **String**, identificador GUID de la petición informado por el cliente o generado por el sistema.
 - **TotalMobiles** : **Integer**, número de móviles enviados por el cliente para validar.

GetMobileStatus

Haga uso de este método para obtener el estado de la validación de un móvil

Entrada (parámetros):

- **login**
- **password**
- **searchData** : estructura del tipo **SearchData**, que contiene los siguientes miembros:
 - **RequestId** : **String**, identificador GUID de la petición informado por el cliente o generado por el sistema.
 - **Mobiles** : lista del tipo **String**, contiene los móviles que se quieran consultar, puede informarse con formato internacional (p.e. 003469999999 ó +3469999999), o en formato nacional (p.e. 699999999).

NOTA: Debe informarse al menos uno de los dos miembros de la estructura **SearchData**, p.e.:

 - si se quiere consultar el estado de todos los móviles correspondientes a una misma petición, hay que informar solo el **RequestId**.
 - si se quiere consultar el estado de uno o más móviles de diferentes peticiones, hay que informar solo los móviles a través del miembro **Mobiles**
 - si se quiere consultar uno o más móviles correspondientes a una petición, debe informarse ambos miembros de la estructura.

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

- **ErrorCode**
- **ErrorDescription**
- **Result** : estructura **GetMobileStatusResultDetail** con los siguientes miembros
 - **StatusCode** : **Integer**, informa un código de estado (ver estados de móviles).
 - **StatusDescription** : **String**, informa una descripción de estado (ver estados de móviles).
 - **RequestId** : **String**, identificador GUID de la petición informado por el cliente o generado por el sistema.
 - **Mobile** : **String**, contiene el móvil que se ha verificado.

6. Códigos de Error

Errores del API (ErrorCode, ErrorDescription):

- 0 éxito, no tiene descripción
 - Tiene lugar en cualquier método
- 1 "Error de autenticación: ..."
 - Puede tener lugar en cualquier método
- 2 "Error de validación: ..."
 - Puede tener lugar en cualquier método donde haya otros parámetros además de las credenciales
- 3 "Error de crédito: ..."
 - Puede tener lugar en el método CreateRequest
- 4 "Error en la creación de la petición: ..."
 - Puede tener lugar en los métodos CreateRequest
- 5 "Error al realizar la búsqueda: ..."
 - Puede tener lugar en el método GetMobileStatus
- 6 "Error desconocido"
 - Puede tener lugar en cualquier método

7. Estados de móviles

Código	Descripción
1	Pendiente de Validar
2	Procesando
3	Móvil No Válido
4	Móvil Válido
5	Móvil Restringido o Bloqueado
6	Móvil Inexistente en el Operador
7	Móvil sin Servicio
8	Móvil Apagado o Fuera de Cobertura
9	Validación Expirada
10	Error de Conexión con Operador
11	Error de Conexión con el Móvil

Los estados con código 1 y 2 son estados intermedios que están en proceso de validación, son susceptibles de cambiar de estado. Al no ser estados finales no se recomienda realizar envío de SMS.

El estado con código 9 es un estado final que se asigna cuando se ha intentado 3 veces obtener el estado de un terminal y no se ha podido realizar con éxito. Al no poder establecer conexión con el terminal, es un indicativo que el móvil no recibirá SMS.



El estado 10 se asigna cuando ha habido un error de conexión con el operador y no se ha podido realizar la consulta del estado del terminal. En caso de que se observe un alto porcentaje póngase en contacto con soporte.aplicateca@linkmobility.com y revisaremos la causa lo antes posible.

Los estados 8 y 11 son susceptibles de cambiar de estado y por tanto pueden llegar a recibir SMS. Son estados indeterminados.

Los estados 3, 5, 6 y 7 son indicativos que el móvil no recibirá SMS. Son estados que difícilmente volverán al estado 4.

Los estados 4 también son susceptibles de cambiar de estado, por este motivo se recomienda realizar el envío de SMS en un período inferior a las 24/48hs.

8. Recomendaciones

Es recomendable realizar envío de SMS a aquellos móviles que tengan el código de estado 4 (Móvil Válido) en un período inferior a las 24/48hs. después de haber realizado la validación.

La validación de cada móvil requiere de un tiempo mínimo para comprobar el estado del terminal, por lo tanto al realizar una petición de validación con el método [CreateRequest](#) no se devuelve inmediatamente el estado del móvil. Por cada móvil se realiza una primera validación para comprobar si el formato es válido y luego una segunda validación que determina el estado del terminal.

Algunos estados pueden resolverse rápidamente, mientras que otros pueden llevar un poco más de tiempo dependiendo del terminal y la cobertura de cada operadora.

El estado de cada móvil puede consultarse mediante el método [GetMobileStatus](#).

9. 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