

API Web Services TEA Email

Indicar la forma de cómo hacer uso del Webservice de TEA Email para el envío de correos transaccionales.

- [API Envíos Transaccionales](#)

API Envíos Transaccionales

1 Introducción

Este documento describe la integración con el servicio Webservice de **TEA Email** para el envío de **correos transaccionales**.

Los correos transaccionales son aquellos enviados como resultado de una acción directa del usuario, por ejemplo, una confirmación de compra, cambio de contraseña o notificación. A diferencia de los correos masivos, estos mensajes están dirigidos a destinatarios específicos y responden a eventos específicos.

Para realizar el envío, el sistema requiere que previamente se configure un **Transactional Process** en TEA Email, el cual define los campos, formato de los datos y comportamiento del mensaje. Este proceso genera una llave ("**Key**") que identifica de forma única el envío y que debe utilizarse para la invocación del Webservice.

Audiencia 1 - Transactional Processes

Import configuration

Name: Reporte Saldo
Description: Reporte Saldo
Action: INSERT_UPDATE
Key: 008ccb8d3e624f3bcedf1b6d091e604
Test key: 008ccb8d3e624f3bcedf1b6d091e604-Test

The file has headers

File Parameters

Column delimiter: TAB
Record delimiter: ENTER
Other Column:
Other Record:

Fields configuration

EMAIL	GRUPO	SALDO
-------	-------	-------

Message configuration

Launch name: Test Close Date_HTML_Spanish

1. **Key**: es el identificador único para el Transactional Process que se está creando, este key se usa para hacer el llamado vía Webservice de la versión productiva.
2. **Test Key**: este key se usa para hacer pruebas del Transactional Process, de decir, una vez que se crea un Transactional Process y este es liberado a producción, es posible hacer modificaciones a ese Transactional Process sin afectar la versión productiva.
3. **Column delimiter**: Es el delimitador que se usará para identificar las diferentes columnas que conforman el contenido enviado a un Transactional Process.
4. **Record delimiter**: Es el delimitador que se usará para distinguir los diferentes registros recibidos para procesarse.

2 Estructura del Servicio

2.1 Estructura general de la petición

El servicio está basado en un Webservice SOAP. El método principal para realizar envíos es `newImportJob`, el cual permite importar los datos para su envío mediante una cadena `String` estructurada.

2.2 Parámetros requeridos

Endpoint

Esta operación se realiza mediante un servicio **SOAP**. El servicio está disponible dependiendo del entorno y configuración de la TEA:

Ruta

`http://TeaDomain/ws/services/ImportService?wsdl`

Nota: reemplace "TeaDomain" por el dominio correspondiente a su instancia de TEA.

Autenticación y Headers

Este servicio no requiere autenticación por token. El acceso se basa en la dirección del Webservice y Key del Transactional Process.

Parámetros

Campo	Descripción
Key	Cadena de texto que representa la llave del Transactional Process (Testing/Productiva).
Content	Cadena de texto con los datos a importar. Las columnas deben ir delimitadas según el formato establecido en el Transactional Process. Se pueden enviar múltiples registros usando el delimitador de registros configurado.

3 Métodos del servicio Web

El servicio WEB contiene los siguientes métodos:

3.1 Void newImportJob (Key, Content)

Este método se usa para colocar en el servidor los datos que serán importados, cuando estos vengan desde un string.

Parámetros de uso:

- i. **Key:** Es un tipo de dato "string", el cual debe contener la llave del Transactional Process que será utilizado para la importación de los datos.
- ii. **Content:** Es un tipo de dato "string", que contiene los datos que serán importados, y la posición de estos datos deben de encontrarse acorde con el orden de las columnas definidas en el Transactional Process. Se pueden recibir más de un registro por petición, solo se deben de separar estos registros por el separador indicado en la definición del Transactional Process.

3.2 Invocación .NET

Mediante Service Reference

```
public void invokeServiceReference() {  
    WSImport.ImportServicePortTypeClient wsImport = new WSImport.ImportServicePortTypeClient();  
    wsImport.newImportJob("[import key]", "[import content]"); }  
}
```

Mediante web references (versiones anteriores al framework 2.0):

```
public void invokeWebReference() {  
    WebReferenceImport.ImportService wsImport = new WebReferenceImport.ImportService(); bool return1;  
}
```

```
bool returnSpecified;
wsImport.newImportJob("[import key]", "[import content]", out return1, out returnSpecified);
}
```

Configuración del endPoint <client>

```
<endpoint address="http://TeaDomain/ws/services/ImportService.ImportServiceHttpSoap11Endpoint/"
binding="basicHttpBinding" bindingConfiguration="ImportServiceSoap11Binding"
contract="WSImport.ImportServicePortType" name="ImportServiceHttpSoap11Endpoint" />
</client>
```

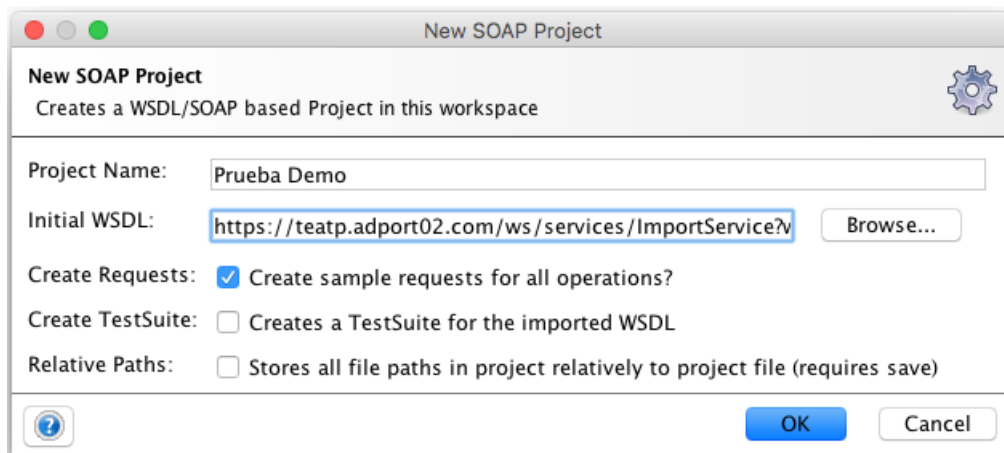
3.3 Invocación PHP

```
<?php
require_once('NUSOAP/nusoap.php');
$oClient = new nusoap_client("http://TeaDomain/ws/services/ImportService?wsdl", true); $oClient-
>setEndPoint('http://TeaDomain/ws/services/ImportService.ImportServiceHttpSoap12Endpoint/');
$content='test@tralix.com|Nombre'; $param=array('importKey'=>'124917014be244a6972581f80286b8ab',$Content);
$client->use_curl = TRUE;
$oResult = $oClient->call('newImportJob',$param); ?>
```

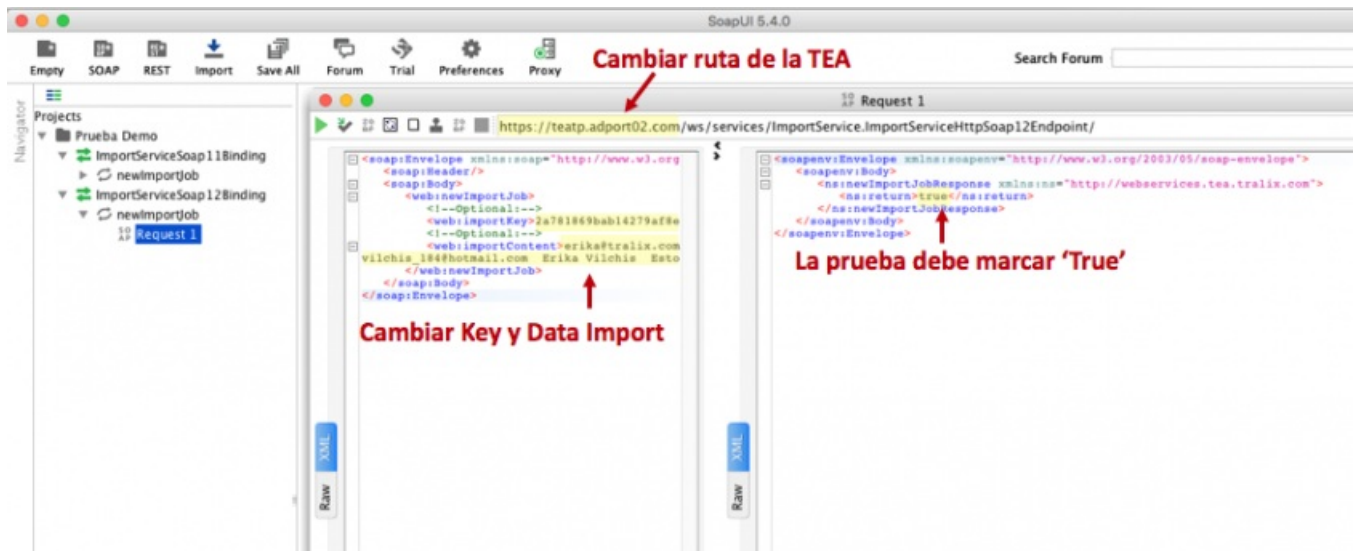
3.4 Invocación Java

```
ImportClient importClient = new ImportClient();
ImportPortType importPortType =
importClient.getImportHttpSoap11Endpoint("http://IPTEA.net/ws/services/ImportService");
importPortType.newImportJob(importKey, newImportContent);
importPortType.newImportJob("b47b0bb98ccd435687a790ba69849594",
"campo1|campo2|campo3\r\notrocampo1|otrocampo2|otrocampo3\r\n");
```

3.5 Ejemplo SoapUI



1. Abrir el programa SOAP
2. Hacer click en el botón SOAP
3. En la ventana se debe proporcionar un nombre de proyecto.
4. Posteriormente se debe ingresar la siguiente liga, cambiando la dirección de la TEA donde desea hacer las pruebas (color amarillo): <https://teatp.adport02.com/ws/services/ImportService?wsdl>
Ejemplo:
 1. En la sección de **ImportServiceSoap 12 Binding**, hacer click para que despliegue la sección de **Request 1** y dar doble click.
 2. Actualizar los siguientes datos: ruta, key, base de pruebas (marcado en color amarillo) y dar click en el icono de play (color verde).
 3. Para que la prueba sea exitosa debe marcar **True**.
 4. Posteriormente recibirá el o los correos electrónicos a los correos que ingresó de prueba.



2.4 Ejemplo de respuesta

En una invocación exitosa, el servicio devuelve un valor booleano `true`.

2.5 Códigos de respuesta

Actualmente no se cuenta con códigos de respuesta para las peticiones vía Webservice.

3 Consideraciones Técnicas

- El `Content` debe respetar el orden y los delimitadores definidos en el Transactional Process.
- Para envíos con archivos adjuntos, asegúrese que los archivos estén cargados en el servidor SFTP de Tralix y que el campo tipo "**Attachment Field**" esté presente en el layout de datos.

Envíos con archivos adjuntos personalizados

TEA tiene la capacidad de enviar **archivos adjuntos personalizados** en correos transaccionales, siempre y cuando éstos se encuentren en la misma TEA. Antes de realizar la invocación del web service se debe asegurar que tenga creado un campo de tipo "Attachment fields" (es el campo que almacenará los nombres de los archivos que le correspondan a cada usuario), el campo es obligatorio siempre que requiera enviar documentos adjuntos y debe formar parte del layout de su base de datos.

Antes de realizar el envío debe asegurarse que los archivos adjuntos se hayan subido exitosamente al SFTP, generar la base de datos con la liga donde se encuentra el archivo y posteriormente hacer la invocación del web service para realizar el lanzamiento del envío.

NOTA: el documento adjunto se hospeda en nuestros servidores vía **SFTP** y es necesario revisar el alcance y costos de almacenamiento con su **Customercare**.

NOTA: para tener acceso al **SFTP** es necesario levantar un ticket al área de **Mesa Servicio**, donde alguno de nuestros ingenieros le proporcionará los accesos y la ruta para depositar sus archivos.

Ejemplo del layout:

EMAIL	RUTA	NOMBRE	APELLIDO
erika@tralix.com	/bienvenidas/001EstadoCuenta-Enero2019.pdf	Erika	Arias

EMAIL	RUTA	NOMBRE	APELLIDO
alejandra@tralix.com	/bienvenidas/004EstadoCuenta-Enero2019.pdf	Alejandra	Morales

4 Contacto y Soporte

Para soporte técnico o dudas sobre el uso del servicio puede comunicarse con el área de soporte técnico de Tralix vía ticket.
