

# API Web Services TEA Email

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

- [API Web Services - TEA Email envíos transaccionales](#)

# API Web Services - TEA Email envíos transaccionales

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

Definir que es un envío transaccional, resaltando su funcionalidad y ejemplificar como consumir el WS.

## 1 Definición de envío de un “Correo Transaccional”

### 1.1 Definición

Un correo transaccional es aquel que es enviado bajo petición del usuario, normalmente se manejan correos transaccionales para confirmaciones de compras, cambios de contraseñas, etc. Es un correo que responde a una acción del cliente(receptor del correo) y no se trata de un envío masivo de un mismo correo a un conjunto de cuentas(audiencia).

Una petición al Webservice de TEA Email, se realiza para que se dispare un correo a determinadas cuentas, en el sistema TEA Email debe estar previamente configurado el envío, es decir, publicado el mensaje y configurado el Transactional Process.

El Transactional Process nos definirá exactamente qué datos y en qué orden se deben de proporcionar en el momento en que se invoca el Webservice de TEA Email.

En la figura anterior es importante resaltar las siguientes propiedades:

The screenshot shows a web application window titled "Audiencia 1 - Transactional Processes". It contains several configuration sections:

- Import configuration:** Fields for Name (Reporte Saldo), Description (Reporte Saldo), Action (INSERT\_UPDATE), Key (008ccb8d3e624ff3bcd1b6d091e604), and Test key (008ccb8d3e624ff3bcd1b6d091e604-Test). There is a checkbox for "The file has headers".
- File Parameters:** Dropdowns for Column delimiter (TAB) and Record delimiter (ENTER). There are also fields for "Other Column:" and "Other Record:", and a "Reload data" button.
- Fields configuration:** A table with columns EMAIL, GRUPO, and SALDO. There are buttons for "Add field", "Clear", and "Load from file".
- Message configuration:** A field for "Launch name" with the value "Test Close Date\_HTML\_Spanish".

At the bottom right, there are "Ok" and "Cancel" buttons.

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 Requisitos para uso del API

Debe de existir un Transactional Process previamente definido. Una vez definido, este genera una llave o "Key" que es un identificador único para cada Transactional Process y es necesario para poder instanciar el servicio web (Web Service) Esta llave sirve para "validar" que paquete de importación es el que se quiere instanciar.

## 3 Localización de los servicios Web para Transactional Process

El web service (WS) se encuentran en:

<http://TeaDomain/ws/services/ImportService?wsdl>

Donde '**TeaDomain**' es el nombre de la **TEA** sobre la cual se desea consumir el WS.

**Por ejemplo:** al acceder desde 'Soap' vamos a reemplazar la ruta:

'<https://192.168.177.11:8080/ws/services/ImportService?wsdl>' por la de nuestro ambiente:

- <https://teamx.adport02.com/ws/services/ImportService?wsdl>
  - <https://teatp.adport02.com/ws/services/ImportService?wsdl>

## 4 Métodos del servicio Web para Transactional Process.

El servicio WEB contiene los siguientes métodos:

### 4.1 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.

### 4.2 Ejemplo de 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);

}

```

TRALIX México S. de R. L. de C. V. 4 Confidencial

API Web Services Para TEA Email

TRLX-TEA 6

Configuración del endPoint <client>

```

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

</client>

```

### 4.3 Ejemplo de 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';
$params=array('importKey'=>'124917014be244a6972581f80286b8ab',$Content);

$client->use_curl = TRUE;
$result = $oClient->call('newImportJob',$params); ?>

```

### 4.4 Ejemplo de invocación JAVA

```

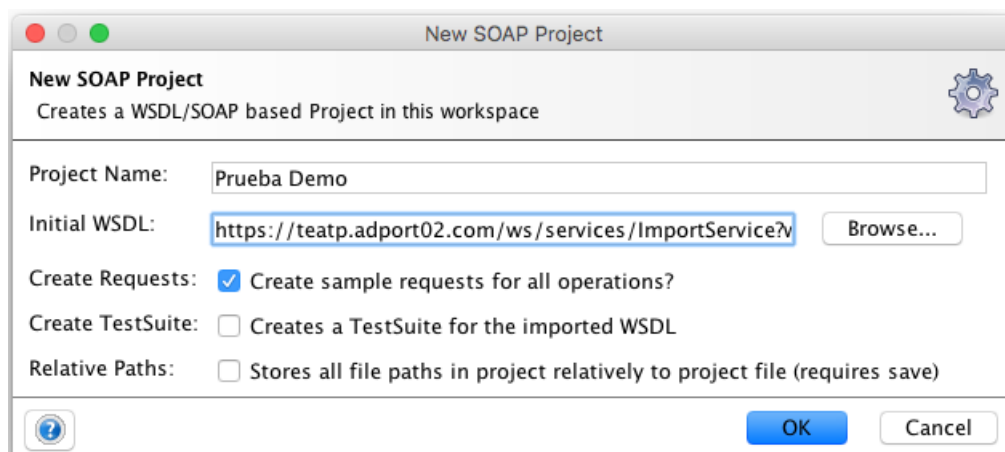
ImportClient importClient = new ImportClient();
ImportPortType importPortType =
importClient.getImportHttpSoap11Endpoint("http://IPTEA.net/ws/services/ImportService");

importPortType.newImportJob(importKey, newImportContent);

importPortType.newImportJob("b47b0bb98ccd435687a790ba69849594",
"campo1|campo2|campo3\r\notrocampo1|otrocampo2|otrocampo3\r\n");

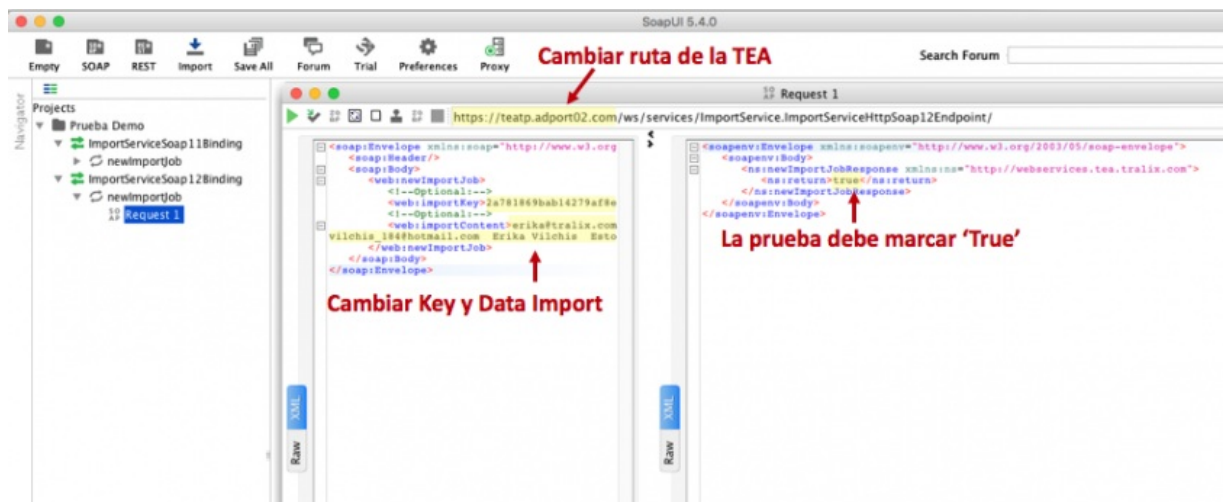
```

### 4.5 Ejemplo desde SOAP



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.



## 5 Manejo de errores de los servicios Web para Transactional Process

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


## 6 Envíos con Attachment


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 “Attachement 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.

Ejemplo del layout:

EMAIL	ruta (Attachement field)	NOMBRE	APELLIDO
<a href="mailto:erika@tralix.com">erika@tralix.com</a>	/bienvenidas/001EstadoCuenta-Enero2019.pdf	Erika	Arias
<a href="mailto:Alejandra@tralix.com">Alejandra@tralix.com</a>	/bienvenidas/004EstadoCuenta-Enero2019.pdf	Alejandra	Morales

 **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.