

SMTP-API Documentación Técnica

1. Descripción.
2. Especificaciones técnicas.
3. Ejemplos de uso
4. Códigos de respuesta de error
5. Guardado de datos

1. Descripción.

Teenvio es una herramienta de email marketing; esto quiere decir que es una herramienta que maneja grandes volúmenes de correo electrónico y hablar de correo electrónico es hablar de SMTP.

SMTP es un protocolo de transferencia de correo electrónico. La primera especificación oficial consta de 1982 y se ha ido mejorando a lo largo del tiempo.

Teenvio, al igual que cualquier otro servicio de email marketing, utiliza internamente SMTP. Con la implantación de este nuevo servicio permitimos adicionalmente, a la utilización de nuestra herramienta, que los propios usuarios de teenvio puedan comunicarse con nuestro sistema a través de dicho protocolo.

Si SMTP es el protocolo para enviar correos electrónicos y teenvio es una herramienta que permite realizar campañas de email marketing tiene sentido que puedas utilizar el protocolo SMTP para enviar correos a través de teenvio.com, uniendo la facilidad del uso del SMTP con la potencia de los servidores de salida de teenvio y sus amplias estadísticas resultantes en campañas de email marketing.

2. Especificaciones técnicas.

2.1. Estándares

El servidor SMTP está creado siguiendo los estándares aplicables, especialmente los siguientes:

<http://tools.ietf.org/html/rfc821> - 1982 - SIMPLE MAIL TRANSFER PROTOCOL

<http://tools.ietf.org/html/rfc1521> - 1993 - MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies

<http://tools.ietf.org/html/rfc1522> - 1993 - MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text

<http://tools.ietf.org/html/rfc1869> - 1995 - SMTP Service Extensions

<http://tools.ietf.org/html/rfc1870> - 1995 - SMTP Service Extension for Message Size Declaration

<http://tools.ietf.org/html/rfc2392> - 1998 - Content-ID and Message-ID Uniform Resource Locators

<http://tools.ietf.org/html/rfc2821> - 2001 - Simple Mail Transfer Protocol

<http://tools.ietf.org/html/rfc3207> - 2002 - SMTP Service Extension for Secure SMTP over Transport Layer Security

Con esto se pretende garantizar que cualquier cliente SMTP que cumpla dichos estándares pueda comunicarse con nuestro servidor SMTP sin problemas.

Igualmente permite, dadas las especificaciones abiertas y públicas, utilizar dichas especificaciones para conectar directamente al servidor SMTP desde un desarrollo propio en cualquier lenguaje de programación y plataforma.

2.2. Parámetros

El servidor SMTP-API de teenvio escucha en dos puertos TCP/IP con las siguientes características:

puerto tcp	2500	46500
host	api.teenvio.com	api.teenvio.com
cifrado inicial	-	ssl
cifrado bajo demanda	si (STARTTLS/TLS)	-
tamaño máximo mensaje	500kb	500kb
requiere login	si	si
método para el login	AUTH LOGIN	AUTH LOGIN

2.3. Particularidades

Para que teenvio acepte el mensaje debe al menos tener dos partes, una de **texto plano** (Content-Type:text/plain) y otra de **texto HTML** (Content-Type:text/html), si no teenvio rechazará el

mensaje por estar incompleto.

Es **obligatorio identificarse** en la herramienta con el nombre de usuario y contraseña que se utiliza en teenvio.

Los **archivos adjuntos son ignorados** a menos que estén referenciados en el html como imágenes incrustadas (ver [rfc2392](#))

Al finalizar correctamente un envío, **se manda** junto con el OK **el id del envío generado**, útil para; por ejemplo, consultar su estadística mediante api <https://github.com/teenvio/SOAP-API>

Existen unas **cabeceras específicas** de teenvio que pueden incluirse en la cabecera principal de un mensaje para personalizar ciertos parámetros del envío:

2.3.1 Cabecera de mensaje

X-Teenvio-ShowCab

Cabecera que le indica a teenvio si debe o no mostrar la cabecera con el típico mensaje “Si no puede ver correctamente pinche aquí”, por defecto está activada, para desactivarla debe establecerse a “no”: X-Teenvio-ShowCab: no

2.3.2 Redes sociales

X-Teenvio-SocialLinks

Cabecera que le indica a teenvio si debe o no mostrar los iconos con los links a los perfiles sociales en el pié del envío. Por defecto está desactivado, para activarlo se debe establecer a “yes”:

X-Teenvio-SocialLinks: yes

2.3.3 Idioma

X-Teenvio-Language

Cabecera que le indica a teenvio qué idioma debe utilizar en los textos de cabecera y pié que se insertan de forma automática, es un valor numérico, por defecto está establecido en ‘3’, correspondiente al idioma Español/Castellano, la actual lista de Idiomas es la siguiente:

id	nombre
3	Español/Castellano
4	Inglés

5	Italiano
6	Francés
7	Alemán
8	Holandés
11	Portugués
13	Noruego
14	Sueco
15	Catalán
16	Finlandés
17	Gallego
18	Euskera

Ejemplo que indicaría idioma inglés:

X-Teenvio-Language: 4

Puede ver un ejemplo de uso en la sección 3.2.2. Protocolo

3. Ejemplos de uso

3.1 Utilizando el servidor SMTP en un cliente de correo electrónico:

La forma más sencilla de utilizar el SMTP-API de teenvio es configurando tu cliente de correo electrónico para que utilice nuestro servidor smtp. De esta forma, si envías una campaña de email marketing utilizando dicho SMTP el envío lo gestionará teenvio.

Configuración de servidor de correo saliente SMTP desde gestores de correo

Servidor SMTP - api.teenvio.com

Puerto - 46500

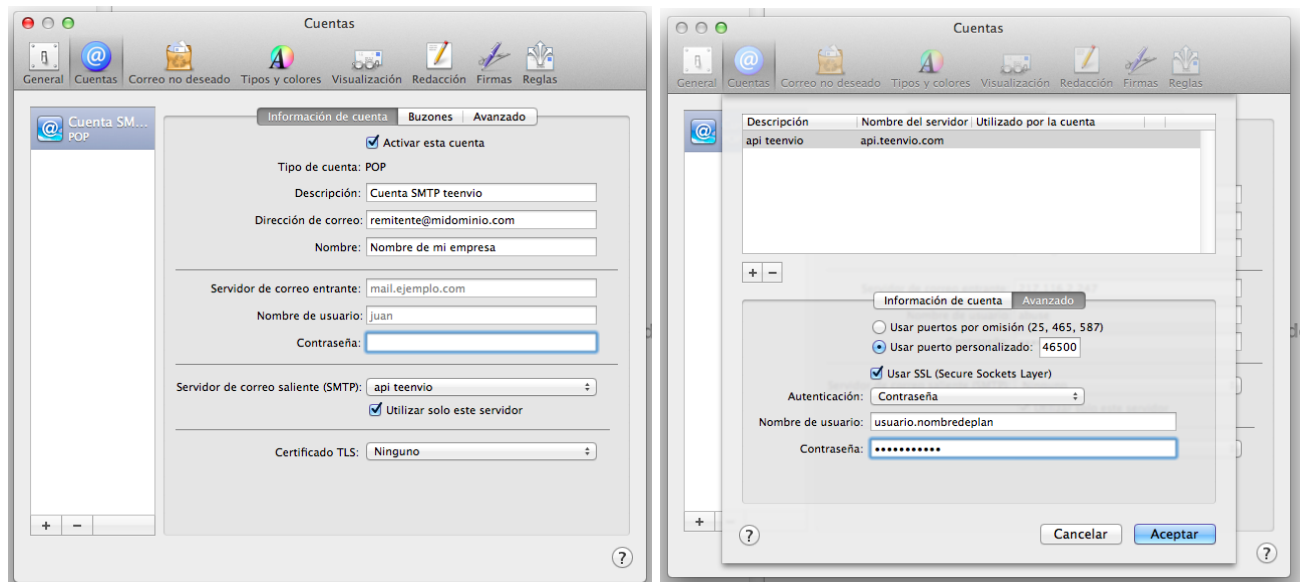
Seguridad - SSL/TLS

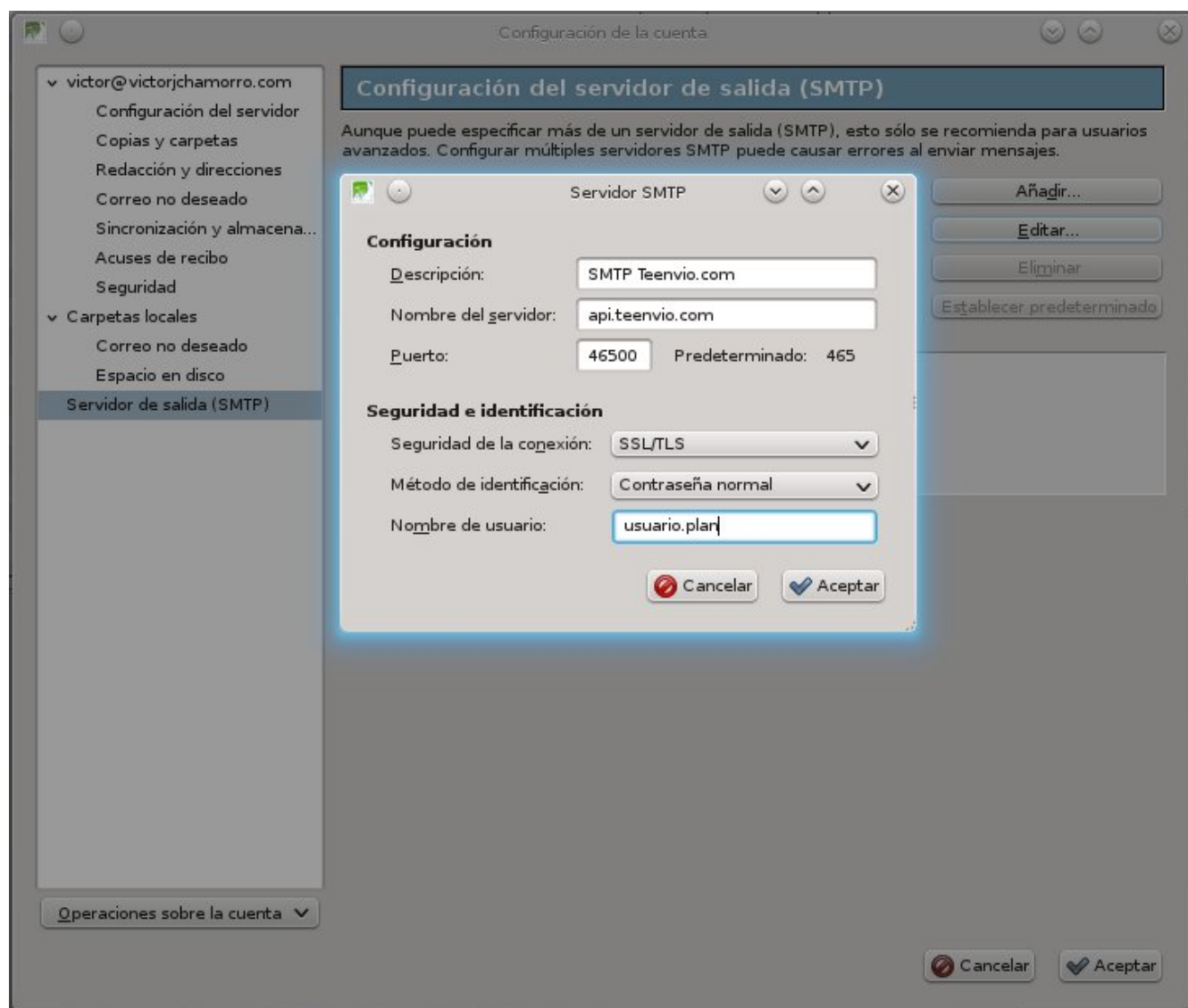
Método de identificación- contraseña normal

Nombre de usuario - usuario teenvio (usuario.nombredeplan)

Clave - clave teenvio

Aquí se ve, por ejemplo, la configuración del SMTP en el cliente de correo electrónico Mail Apple Thunderbird:





Utilizar un cliente de correo electrónico que haga uso del SMTP puede ser útil para pruebas

3.2 Mediante programación para utilizarlo en sus desarrollos

Con los parámetros adecuados, tal y como explicamos en el presente documento, es fácil realizar un envío con nuestro SMTP mediante programación, veamos algún ejemplo:

3.2.1 PHP5

Ejemplo en php5 donde conectamos al servidor SMTP, mostramos su mensaje inicial, saludamos y definimos el saludo personalizado que nos devuelve, luego cerramos la conexión.

```
<?php
$fp = stream_socket_client("ssl://api.teenvio.com:46500", $errno, $errstr, 30);
if (!$fp) {
    echo "$errstr ($errno)<br />\n";
} else {
    echo "Conexión válida, saludo del SMTP:<br/>\n";

    $buffer = fgets($fp, 1024);
    echo $buffer."<br/>\n";

    echo "Contesto al saludo y su respuesta es:<br/>\n";

    fwrite($fp, "EHLO PruebaPHP\n");

    while (($buffer = fgets($fp, 1024)) !== false) {
        echo $buffer."<br/>\n";
        if(substr($buffer,0,4) == "250 ") break;
    }

    echo "Finalizo esta prueba<br/>\n";

    fwrite($fp, "QUIT\n");
    $buffer = fgets($fp, 1024);
    echo $buffer;

    fclose($fp);
}
?>
```

Es un ejemplo muy básico y sin utilidad práctica, pero ilustra claramente cómo debe conectarse y cómo establecer una conversación con el servidor SMTP, tan solo hay que continuar con los comandos oportunos.

3.2.2 Protocolo

Una conversación real que sí provocaría el envío sería la siguiente:

```
001 >> 220 api.teenvio.com ESMTP API teenvio.com Thu, 12 Sep 2013 13:59:15 +0200
002 << EHLO MaquinaPrueba
003 >> 250-api.teenvio.com ESMTP API teenvio.com. Hola MaquinaPrueba!
004 >> 250-Recuerda que debes ser un usuario registrado de teenvio.com. http://teenvio.com/
005 >> 250-Usa este servicio con responsabilidad cumpliendo con las condiciones de uso.
006 >> 250-AUTH LOGIN
007 >> 250-SIZE 512000
008 >> 250-HELP
009 >> 250 AUTH LOGIN
010 << AUTH LOGIN
011 >> 334 VXNlcm5hbWU6
012 << [USUARIO EN BASE64]
013 >> 334 UGFzc3dvcmQ6
014 << [PASSWORD EN BASE64]
015 >> 235 Authentication Succeeded
016 << MAIL FROM:<remitente@dominio.com> SIZE=1307
017 >> 250 OK
018 << RCPT TO:<usuario@dominio.com>
019 >> 250 Accepted
020 << DATA
021 >> 354 Enter message, ending with "." on a line by itself
022 << Message-ID: <5231AC96.9010208@dominio.com>
023 << Date: Thu, 12 Sep 2013 13:59:18 +0200
024 << From: =?ISO-8859-1?Q?=22V=EDctor_J=2E_Chamorro=22?= <victor@dominio.com>
025 << X-Teenvio-SocialLinks: yes
026 << User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:10.0.12) Gecko/20130116 Thunderbird/10
027 << MIME-Version: 1.0
028 << To: usuario@dominio.com
029 << Subject: prueba correo simple
030 << Content-Type: multipart/alternative;
031 << boundary="-----050004090406000109090608"
032 <<
033 << This is a multi-part message in MIME format.
034 << -----050004090406000109090608
035 << Content-Type: text/plain; charset=ISO-8859-1; format=flowed
036 << Content-Transfer-Encoding: 8bit
037 <<
038 << Buenas,
039 <<
040 << Esto es un email *html* para probar el servidor smtp-api de teenvio
041 <<
042 << Un saludo,
043 <<
044 << _Victor J. Chamorro_ - teenvio.com
045 <<
```

```

046 <<
047 << -----050004090406000109090608
048 << Content-Type: text/html; charset=ISO-8859-1
049 << Content-Transfer-Encoding: 7bit
050 <<
051 << <html>
052 <<   <head>
053 <<       <meta http-equiv="content-type" content="text/html;
054 <<         charset=ISO-8859-1">
055 <<   </head>
056 <<   <body bgcolor="#FFFFFF" text="#000000">
057 <<       Buenas,<br>
058 <<       <br>
059 <<       Esto es un email <b>html</b> para probar el servidor smtp-api de
060 <<       teenvio<br>
061 <<       <br>
062 <<       Un saludo,<br>
063 <<       <br>
064 <<       <u>V&iacute;ctor J. Chamorro</u> - teenvio.com<br>
065 <<       <br>
066 <<   </body>
067 << </html>
068 <<
069 << -----050004090406000109090608--
070 << .
071 >> 250 OK id=1166 fa8896c2351f8b7e73ac3a3d209c6ac5
072 << QUIT
073 >> 221 2.0.0 Closing connection.

```

Paso a detallar paso por paso:

- Leyenda:

“>>” significa mensaje entrante, desde el servidor a nuestra máquina.

“<<” significa mensaje saliente, desde nuestra máquina al servidor.

- Líneas de la 001 a la 009:

Es el protocolo de **saludos**, saludamos nosotros y el servidor nos saluda además dándonos una serie de información, como el método de login, el tamaño máximo que permite por mensaje, etc.

- Líneas de la 010 a la 015:

Mecanismo de **login**, los mensajes codificados utilizan BASE64, no es un riesgo de seguridad mientras utilicemos la conexión con SSL o, en conexiones normales, lancemos una conexión segura con STARTTLS previa al login.

- Líneas de la 016 a la 017:

Identificamos el **remite** que utilizaremos, debe estar activo y validado dentro del plan de

teenvio, o fallará.

- Líneas de la 018 a la 019:

Indicamos un **destinatario** (TO:), estas dos líneas deben repetirse por cada uno de los destinatarios a los que deseemos enviar, en el ejemplo solo hay un destinatario.

- Líneas de la 020 a la 071:

Especificamos el **contenido** del email en sí.

- Líneas de la 022 a la 031:

Cabeceras principales del mensaje, aquí establecemos el asunto e indicamos que el mensaje está compuesto por varias partes (`Content-Type: multipart/alternative;`) entre otras cabeceras. Las únicas cabeceras que utilizará teenvio son la de Subject y las cabeceras propias X-Teenvio-*

- Línea 025:

Ejemplo de cabecera con un parámetro para teenvio, en este caso establece que se deben mostrar los iconos con las redes sociales en el pie del envío. (Ver sección 2.3 Particularidades)

- Líneas de la 034 a la 046:

Parte de **texto plano** del email (`Content-Type: text/plain;`), es imprescindible.

- Líneas de la 047 a la 069:

Parte de **texto html** del email (`Content-Type: text/html;`), es imprescindible, será lo que se vea en la mayoría de gestores

- Línea 070:

Indica el **fin del contenido** del mensaje, muy importante. Es una línea nueva con sólo un punto y un salto de línea.

- Línea 071:

El servidor responde que el envío se ha **cursado correctamente**, se empezará a tramitar inmediatamente, informa además, del **id interno del envío lanzado**, junto con un identificador de envío smtp.

- Líneas de la 072 a la 073:

Desconexión, el cliente manda QUIT y el servidor se despide y finaliza la conexión.

Hay que recordar los siguientes puntos:

- El remitente debe existir en el plan de teenvio del cliente logueado.
- El mensaje debe estar compuesto mínimo por dos partes, texto plano y texto html.
- El tamaño máximo del mensaje es de 500kb.
- Si el servidor no responde 250 OK al comando DATA, no se cursará ningún envío
- Los contactos son insertados y agrupados aunque no se finalice un envío, lo que permite realizar importaciones sin enviar nada utilizando el comando DATA únicamente con el punto de fin de contenido (línea 070)
 - Los archivos adjuntos serán ignorados, salvo aquellos que estén referenciados dentro del HTML, como las imágenes incrustadas (cid:), que serán transformadas en imágenes normales alojadas en los servidores de teenvio. (ver estándar [rfc2392](#))
 - Junto con el OK final (línea 071) se manda el id del envío interno recién generado. Este identificador puede usarse para consultar la estadística mediante api o simplemente distinguir el envío generado de los demás.

4. Códigos de respuesta de error

500 5.5.1 Invalid command / 551 Invalid command
Comando no reconocido

503 valid RCPT and MAIL command must precede DATA
Antes de usar el comando DATA debes usar los comandos RCPT TO y MAIL FROM

504 A command parameter is not implemented.
El parámetro del comando no está implementado y no es válido en este servidor

530 Authentication required
Antes de realizar el comando debes loguearte

535 Authentication credentials invalid
Los datos utilizados para el login no son válidos

552 message size exceeds fixed maximum message size (512000) bytes
El tamaño del mensaje excede el máximo permitido

553 Remitente no existente o no validado desde su plan de teenvio
No se acepta el remitente, debe existir en teenvio

500 Internal error - [Mensaje]
Error genérico, suele darse al haber algún problema con el comando DATA, el mensaje que acompaña a este error tiene la descripción exacta. Mediante este error se puede informar de envíos insuficientes, exceso de contactos, etc.

500 Syntax error
Comando no reconocido

5. Guardado de datos

Todas las campañas enviadas desde protocolo SMTP son almacenadas en teenvio para su posterior consulta y manejo por el usuario.

Los contenidos y comportamientos identificables en la plataforma teenvio serán los siguientes:

Contactos:

Todos las cuentas de correo a las cuales se realice el envío desde protocolo SMTP generarán un grupo físico identificable en la subsección “Contactos/Grupos” con un distintivo SMTP para ser fácilmente identificables, donde se procederá a la importación y agrupación de los contactos elegidos para el envío.

Envío:

Todos los envíos los cuales se realicen desde protocolo SMTP generarán un envío físico identificable en la subsección “Estadísticas/Envíos” con un distintivo SMTP para ser fácilmente identificables.

Contenido:

El contenido HTML creado para el envío se guardará en la subsección “Piezas/Listado de Piezas” con un distintivo SMTP para ser fácilmente identificables.