



API Email Original

DOC-MTAPI1224

Indice

1. Introducción.....	3
2. Autenticación.....	3
3. Servicios Web.....	4
A) Servicio Crear nuevo sello (web:ServicioEO_GetSello)	4
I) Parámetros de entrada.....	4
II) Parámetros de salida (Respuesta).....	5
III) Respuestas relacionadas con el servicio	5
IV) Ejemplo Javascript (GetSello):	6
B) Servicio Validar sello (web:ServicioEO_ValidarSello).....	7
I) Parámetros de entrada.....	7
II) Parámetros de salida (Respuesta).....	8
III) Respuestas relacionadas con el servicio	8
IV) Ejemplo Javascript (ValidarSello):	9
4. Errores Comunes.....	10

1. INTRODUCCIÓN

La API de Email Original permite interactuar con el sistema para realizar las dos tareas principales de Email Original:

- Obtención del sello Email Original para insertar en el cuerpo de los correos electrónicos.
- Validación de correos

Para implementar esas dos funcionalidades, la API de Email Original, proporciona dos servicios web que facilitan el proceso.

Cada servicio requiere ciertos parámetros y devuelve valores en respuesta. Esta documentación cubre los detalles de ambos servicios para facilitar su integración.

2. AUTENTICACIÓN

La API requiere de una autenticación para obtener el sello Email Original, estas credenciales constan de dos valores:

- Código: La dirección de correo electrónico en la cual se utilizará el sello.
- Clave de API: Una clave de seguridad que Email Original genera para sus clientes.

Para poder obtener una clave API, debe de ser cliente registrado de Email Original y, en el apartado de Datos del Cliente, tiene una opción que le muestra su clave de API. Esta clave es única por cada cliente.

3. SERVICIOS WEB

Para invocar los servicios la url necesaria es:

https://www.emailoriginal.com/_servicios/eo_api.asmx

A) Servicio Crear nuevo sello (web:ServicioEO_GetSello)

El método del servicio es:

ServicioEO_GetSello

Este servicio es el que nos permite obtener un sello para insertar en el cuerpo del correo electrónico. El sello se debe insertar al principio del cuerpo del correo. Se deben indicar unos parámetros de entrada relacionados con el correo y el servicio nos devolverá unos valores que se producen al generar el sello de Email Original.

I) Parámetros de entrada

PARAMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCION
origen	string	"PLU" - Constante definida para utilizar desde la API. Incluirla es obligatorio
desde	string	Dirección de correo electrónico origen, se trata de la cuenta de correo desde la que se envía el mismo.
para	string	Incluir en este campo una lista de direcciones de correo separadas por "," que incluya todos los destinatarios: PARA, CC y CCO
asunto	string	El texto que tenga el asunto del correo electrónico
codigo	string	Dirección de correo electrónico origen, se trata de la cuenta de correo desde la que se envía el mismo. Actualmente, se trata del mismo que se incluye en el parámetro "desde".
tipo	string	"9G" - Constante definida para utilizar desde la API. Incluirla es obligatorio
idioma	string	Valores posibles actualmente, para generar el sello: "ES" – Español

		"EN" – Inglés "PT" – Portugués
clave	string	Clave proporcionada a los clientes de Email Original para poder generar sellos. Puede encontrarla en su cuenta.
datos	string	Si se trata de un correo programado el valor de este campo debe ser la fecha programada del envío, con el formato "DD/MM/YYYY HH:MM:SS" siendo esta la hora UTC. Si el envío NO es programado, en este campo se incluirá un string vacío " "

II) Parámetros de salida (Respuesta)

PARAMETROS DE SALIDA (RESPUESTA)		
POSICION	TIPO	DESCRIPCION
0	string	En la primera posición devuelve un valor que puede ser: "0" – La generación del sello fue errónea "1" – La generación del sello fue correcta
1	string	En esta posición nos devuelve un string indicando el posible error detectado al generar el sello, si el proceso fue correcto, vendrá vacío.
2	string	Si se generó un sello correcto, nos devuelve en este string un sello de Email Original en formato HTML listo para ser insertado en el cuerpo del correo.

III) Respuestas relacionadas con el servicio

Se pueden recepcionar respuestas relacionadas con un servicio web, tales como:

- Código de Estado 200 OK: La solicitud fue exitosa.
- Código de Estado 400 Bad Request: Parámetros inválidos.
- Código de Estado 500 Internal Server Error: Error en el servidor.

IV) Ejemplo Javascript (GetSello):

```
const getSealEO = async (
  correoOrigen, // "PLU"
  correoDe, // "remitente@email.com"
  correoPara, // "receptor@email.com"
  correoAsunto, // "Asunto de prueba"
  correoCodigo, // "remitente@email.com" Igual que el correoDe
  correoTipo, // "9G"
  correoldioma, // "ES"
  clave, // "Tu clave API"
  datos // " " string vacío si es un correo normal, o "DD/MM/YYYY HH:MM:SS" si es programado como se indica en la tabla
) => {
  const soapRequest = `<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:ServicioEO_GetSello>
      <web:origen>${correoOrigen}</web:origen>
      <web:desde>${correoDe}</web:desde>
      <web:para>${correoPara}</web:para>
      <web:asunto>${correoAsunto}</web:asunto>
      <web:codigo>${correoCodigo}</web:codigo>
      <web:tipo>${correoTipo}</web:tipo>
      <web:idioma>${correoldioma}</web:idioma>
      <web:clave>${clave}</web:clave>
      <web:datos>${datos}</web:datos>
    </web:ServicioEO_GetSello>
  </soapenv:Body>
</soapenv:Envelope>`;

  const requestOptions = {
    method: 'POST',
    headers: { 'Content-Type': 'text/xml;charset=UTF-8',
      SOAPAction: 'http://tempuri.org/ServicioEO_GetSello'
    },
    body: soapRequest
  };

  try {
    const response = await fetch(urlServicio, requestOptions);
    const data = await response.text();

    const parser = new xml2js.Parser();
    const parsedData = await parser.parseStringPromise(data);

    const responseBody = parsedData['soap:Envelope']['soap:Body'][0];
    const selloResult = responseBody.ServicioEO_GetSelloResponse[0].ServicioEO_GetSelloResult[0].string;
    const sello = selloResult[2];
    return sello;
  } catch (error) {
    console.error('Error al realizar la solicitud SOAP:', error);
    return "";
  }
};
```

B) Servicio Validar sello (web:ServicioEO_ValidarSello)

El método del servicio es:

ServicioEO_ValidarSelelo

Este servicio es el que nos permite validar un correo electrónico, esto se consigue indicando unos parámetros de entrada relacionados con el correo, el servicio nos devuelve unos valores de respuesta que nos indica la validez del correo en función del sello de Email Original.

l) Parámetros de entrada

PARAMETROS DE ENTRADA		
NOMBRE	TIPO	DESCRIPCION
origen	string	"PLU" - Constante definida para utilizar desde la API. Incluirla es obligatorio
desde	string	Dirección de correo electrónico origen, se trata de la cuenta de correo desde la que se envía el mismo.
para	string	Incluir en este campo una lista de direcciones de correo separadas por "," que incluya todos los destinatarios: PARA, CC y CCO.
fecha	string	Fecha del correo electrónico en formato "DD/MM/YYYY HH:MM:SS" siendo esta la hora UTC.
asunto	string	El texto que tenga el asunto del correo electrónico
codigo	string	EL Codigo de SEGURIDAD que aparece en el sello mostrado en el correo electrónico. Este está formado por unas marcas iniciales ##EO## y la final ####, un ejemplo: ##EO##gtgtgtgtgtgtfsPdvfvftgtvtvtgtvtgtvtgtgtgtgtgtgtvtvfvmeXzMCAAA=####
clave	string	" " – String vacío. El servicio es abierto, no es necesario incluir la clave ni se recomienda hacerlo

II) Parámetros de salida (Respuesta)

PARAMETROS DE SALIDA (RESPUESTA)		
POSICION	TIPO	DESCRIPCION
0	string	En la primera posición devuelve un valor que puede ser: "0" – La validación es errónea "1" – La validación es correcta
1	string	En esta posición nos devuelve los siguientes posibles valores: "" – Un string vacío indica que el correo no posee un sello de EMAIL ORIGINAL "ERROR" – Se trata de un sello de Email Original No valido "ERRORSELLOFECHA" – Contiene un sello de Email Original que parece correcto, pero hay un problema temporal entre la emisión del sello y la fecha de recepción. Se considera sospechoso o no seguro. Si la posición 1 era un valor "1" esta posición no se utiliza, se considera que el correo contiene un sello correcto y válido.

III) Respuestas relacionadas con el servicio

Se pueden recepcionar respuestas relacionadas con un servicio web, tales como:

- Código de Estado 200 OK: La solicitud fue exitosa.
- Código de Estado 400 Bad Request: Parámetros inválidos.
- Código de Estado 500 Internal Server Error: Error en el servidor.

IV) Ejemplo Javascript (ValidarSello):

```
const verifySealEO = async (
  correoOrigen, // "PLU"
  correoDe, // "remitente@email.com"
  correoPara, // "receptor@email.com"
  correoFecha, // "DD/MM/YYYY HH:MM:SS"
  correoAsunto, // "Asunto de prueba"
  correoCodigo, // ##EO##gtgtgtgtgtgfsPdvfvftgtvtvtgtvtgtgtgtgtgtvtvfvmeXzMCAAA=##*##
  correoClave, // " " string vacío como se indica en la tabla
) => {
  const soapRequest = `<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:ServicioEO_ValidarSello>
      <web:origen>${correoOrigen}</web:origen>
      <web:desde>${correoDe}</web:desde>
      <web:para>${correoPara}</web:para>
      <web:fecha>${correoFecha}</web:fecha>
      <web:asunto>${correoAsunto}</web:asunto>
      <web:codigo>${correoCodigo}</web:codigo>
      <web:clave>${correoClave}</web:clave>
    </web:ServicioEO_ValidarSello>
  </soapenv:Body>
</soapenv:Envelope>`;

  const requestOptions = {
    method: 'POST',
    headers: {
      'Content-Type': 'text/xml;charset=UTF-8',
      SOAPAction: 'http://tempuri.org/ServicioEO_ValidarSello'
    },
    body: soapRequest
  };

  try {
    const response = await fetch(urlServicio, requestOptions);
    const data = await response.text();

    const parser = new xml2js.Parser();
    const parsedData = await parser.parseStringPromise(data);

    const responseBody = parsedData['soap:Envelope']['soap:Body'][0];
    const verifyResult = responseBody.ServicioEO_ValidarSelloResponse[0].ServicioEO_ValidarSelloResult[0].string;

    return verifyResult;
  } catch (error) {
    console.error('Error al realizar la solicitud SOAP:',
    error);
    return ['0', 'ERROR'];
  }
};
```



4. ERRORES COMUNES

En la utilización de los servicios web, un error muy común que ocurre es el de no indicar los parámetros de forma correcta, asegurarse de que los nombres de los parámetros sean correctos, teniendo en cuenta las mayúsculas y minúsculas de los mismos.

Tener en cuenta que son servicios web y, por lo tanto, tener en cuenta la asincronía de los mismos.

