

Integración con el Bigmailer.cloud - API

Registros en la Red, S.L.

www.registros.net

www.bigmailer.net

Tabla de contenido

Introducción.....	3
Seguridad.....	4
Integración Web.....	5
SOAP.....	6
Entrada JSON.....	7
Entrada.....	7
Estructura.....	7
Codificación.....	8
El resultado de la llamada.....	9
Contact – Modos de trabajo.....	10
El modo de trabajo.....	11
Integración PHP.....	13
Soporte técnico.....	14

Introducción

Bigmailer.cloud como plataforma de envío masivo correos y comunicación a gran escala con los clientes ha sido preparado para fácil integración con existentes sistemas informáticos incluyendo las paginas web.

Como medio de la integración se esta utilizando dos estandartes comunes: SOAP y llamadas http con el uso de codificación de entrada y salida con JSON.

Seguridad

Para poder acceder al API de la plataforma Bigmailer.cloud es necesario utilizar la clave API. Para contecer tu clave API consulte el apartado *Información principal* en la sección *Mi cuenta* dentro del panel de control de bigmailer.cloud.

Es importante que proteja su clave API. No la facilite a los terceros ni utilice en conexiones no seguras. Utilice la clave solo bajo protocolo https.

Integración Web

La plataforma Bigmailer.cloud permite fácil integración con las paginas web. Esta opción puede ser muy interesante para vincular el formulario de alta en el newsletter (boletín). La clase o web service *Contact* puede trabajar en el modo de trabajo *newsletter* lo permite una integración rápida de boletín. El sistema bigmailer se ocupara del envío de un correos de confirmación de alta almacenando toda la información requerida por la nueva normativa RGPD.

Para garantizar maxima seguridad es altamente importante incluir un sistema CAPTCHA dentro del formulario del alta del newsletter.

SOAP

El sistema bigmailer ofrece un explorador SOAP que contiene la lista de servicios y métodos disponible y toda la información necesaria para la integración.

Para utilizar el explorador, visite el sitio: <https://bigmailer.cloud/soap>

Para agaragar contactos nuevos es altamente aconsejable utilizar el servicio *Contact* que puede encontrar siguiendo el vinculo: <https://bigmailer.cloud/es/soap/index/class/Api-Soap-Integration-Contact>

El servicio *Contact* permite aprovechar todas las ventajas de la herramienta **Integración** que encontrara en su panel de control de bigmailer.cloud en la sección **Gestión**. Las funcionalidades destacadas de la herramienta *Integración* son:

- Selección automática de la base de datos;
- Registro de acciones.

Para conocer modos de trabajo de servicio o clase *Contact* consulte el subtitulo *Contact – Modos de trabajo*.

Entrada JSON

Entrada JSON puede ser una atractiva alternativa para el estándar de SOAP. Este método permite la integración incluso en los sistemas que no están preparados para uso de SOAP.

Para utilizar la entrada, lo único que tienes que hacer es efectuar una llamada http a la dirección URL de web service del sistema bigmailer.cloud.

La URL del web service: <https://bigmailer.cloud/api/json/run>

Entrada

La entrada tomo como datos la información pasada dentro del parámetro *input*. El parámetro puede pasarse tanto con el método POST como GET pero es importante garantizar estructura y codificación adecuada de la información.

Estructura

La información de la entrada debe estructurarse como una *array asociativa* que contiene siguientes elementos importantes como claves:

- *class*
- *method*
- *arguments*

class – Contiene el nombre de la clase que se va a utilizar en la llamada. Los nombres de clases corresponden a los nombres que aparecen en el explorador SOAP. Por ejemplo para utilizar la clase *Contact* hay que pasar como el el nombre completo: `\Api\Soap\Integration>Contact`

method – El nombre de método a cual se hace la llamada. Los nombres corresponden a los nombres que aparecen en el explorador SOAP. Por ejemplo para añadir el contacto según la IP del visitante de la web hay que utilizar el método *addByIP*.

arguments – El parámetro arguments debe contener una array con los parámetros que se pasan al método, de acuerdo con la información que aparece en el explorador SOAP. Por ejemplo para el método *addByIP* los parámetros requeridos son: *key*, *email*, *name*, *ip* y *mode*.

Un ejemplo de la estructura de la entrada codificada en PHP se puede encontrar en el listado 1.

```
$input = array(  
    'class' => '\Api\Soap\Integration\Contact',  
    'method' => 'addByIP',  
    'arguments' => array(  
        '77b1fb0f84c424b2ba6de76b1ffce172766b0834',  
        'test@test.net',  
        'Test',  
        '8.8.8.8',  
        0,  
    ),  
);
```

Listado 1: Estructura de la entrada

Codificación

El parámetro input tiene que estar codificado para poder ser pasado mediante el protocolo http. Para codificar el parámetro hay que efectuar dos operaciones: 1) Serializar a formato JSON y 2) codificación según la normativa RFC 3986.

En el listado 2 se presenta el proceso de codificación PHP utilizando las funciones *json_encode()* y *rawurlencode()*.

```
$input = json_encode($input);  
$input = rawurlencode($input);
```

Listado 2: Codificación de la entrada

El resultado de la llamada

Como el resultado de la llamada el sistema devuelve información codificada como JSON. La información después de deserializar tiene estructura de una array asociativa con al menos dos campos: *code* y *result*.

El campo *code* indica si la llamada se ha efectuado con éxito (código 200) o ha fallado (código 5xx).

Si el código de la respuesta es 200, el campo *result* contiene el resultado de la ejecución del metodo indicado en la entrada.

```
$input = array(
    'class' => '\Api\Soap\Integration\Contact',
    'method' => 'addByIP',
    'arguments' => array(
        '77b1fb0f84c424b2ba6de76b1ffce172766b0834',
        'test@test.net',
        'Test',
        '8.8.8.8',
        0,
    ),
);

$input = json_encode($input);
$input = rawurlencode($input);

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "https://bigmailer.cloud/api/json/run");
curl_setopt($ch, CURLOPT_POST, TRUE);
curl_setopt($ch, CURLOPT_POSTFIELDS, "input={$input}");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($ch);
curl_close ($ch);

$response = json_decode($response, TRUE);

var_dump($response);
```

Listado 3: Implementacion de la llamada JSON

El listado 3 contiene un código PHP completo para efectuar una llamada JSON.

Contact – Modos de trabajo

La clase o web service *Contact* es la mejor opción para insertar de contactos nuevos mediante el API de Bigmailer.cloud. El funcionamiento de este método se puede controlar desde panel de bigmailer accediendo a la herramienta **Integración** dentro de la sección **Gestión**. Además la herramienta facilita un registro detallado de las interacciones.

La clase ofrece tres métodos *state()*, *addByIP()* y *addByCountry()*.

El método *state()* permite obtener información acerca del estado de un contacto. El método toma solo un parámetro tipo *string* que debe de ser una dirección email correcta. Los posibles resultados del método se ha presentado en la tabla 1.

El método *addByIP()* permite asignación automática de la base de datos donde se va a añadir el contacto nuevo. Como referencia para elegir la base de datos se toma información de geolocalización basada en la IP.

El método *addByCountry()* permite asignación automática de la base de datos donde se va a añadir el contacto nuevo. Como referencia para elegir la base de datos se toma el código del país según la normativa ISO 3166-1.

Los parámetros que toman los dos métodos han sido descritas en las tablas 2 y 3.

Valor	Descripción
0	El contacto no existe.
1	El contacto existe pero no tiene asignados los permisos para mailing.
2	El contacto existe pero el envío esta prohibido.
3	El contacto existe y tiene permiso para el mailing.
4	El contacto existe y tiene permiso para el mailing pero no se cumple con la normativa RGPD.
5	El contacto existe y tiene permiso para el mailing pero el control de RGPD has sido desactivado. Se supone que la información requerida por la normativa RGPD se guarda fuera del bigmailer.cloud.

Tabla 1: Resultados del método *state()*

Parámetro	Tipo	Descripción
key	string	API Key
email	string	Dirección de correo.
name	string	Nombre de contacto. Se puede dejar vacío.
ip	string	IP del contacto.
mode	integer	El modo de trabajo.

Tabla 2: Parámetros del método addByIP()

Parámetro	Tipo	Descripción
key	string	API Key
email	string	Dirección de correo.
name	string	Nombre de contacto. Se puede dejar vacío.
country	string	Código del país según la normativa ISO 3166-1
ip	string	IP del contacto. Dependiendo del modo del trabajo puede ser obligatorio.
mode	integer	El modo de trabajo.

Tabla 3: Parámetros del método addByCountry()

El modo de trabajo

El modo de trabajo indica al sistema de que manera se tiene que tratar el contacto nuevo. En la tabla 4 se describe con detalle todos modos de trabajo soportados por métodos *addByIP()* y *addByCountry()*.

Valor	Descripción
0	El contacto se va a importar sin permiso para el envío.
1	El contacto se va a importar en modo newsletter. Sin permiso para el envío pero al contacto se va a enviar un correo con la solicitud de la confirmación de alta. Si el contacto acepta la alta, el sistema va a asignar el permiso para envío y va a guardar toda la información requerida por RGPD.
2	El contacto se va a importar sin permiso pero al contacto se le va a enviar una solicitud de permiso para el envío de mailing. Si el contacto acepta la alta, el sistema va a asignar el permiso para envío y va a guardar toda la información requerida por RGPD.
3	El contacto se va a importar con permiso para envío. Además se va a tomar la IP y la fecha y hora actual para guardar el registros requerido por RGPD. En este modo la IP es requerida también si se utiliza el método <i>addByCountry()</i> .
4	El contacto se va a importar con permisos pero el sistema va a desactivar control de RGPD para esta dirección. En este modo de trabajo se supone que la información requerida por RGPD se guarda fuera del bigmailer.cloud.

Table 4: Modos del trabajo

Integración PHP

En caso de la integración con sistemas escritos en PHP están disponibles las librerías que facilitan de forma importante la programación.

Las librerías junto con los ejemplos se puede descargar desde :

<https://github.com/registros/integration>

El listado 4 presenta la interacción con el uso de las librerías.

```
require_once 'vendor/Registros/loader.php';

use Registros\Bigmailer\Api\Contact;

$contact = new Contact('77b1fb0f84c424b2ba6de76b1ffce172766b0834');
$result = $contact->addByIP('test@test.net', 'Test', '8.8.8.8', Contact::ADD_MODE_WITH_PERMISSION);

var_dump($result);
```

Listado 4: Integración utilizando las librerías

Soporte técnico

En caso de dudas o problemas con la integración, las preguntas se puede enviar por el correo electrónica a la dirección de *soporte@registros.net* .