WebService MailUpSend

Overview

MailUpSend allows you to replicate - outside of the MailUp admin console - many of the common tasks that are typically performed within the MailUp admin console.

The service is available at the following locations (the WebService is accessible through the SOAP protocol):

- https://wsvc.ss.mailup.it/MailupSend.asmx
- ~~http://services.mailupnet.it/mailupsend.asmx~~ (only https is supported)

The corresponding WSDL is available at the following location:

- https://wsvc.ss.mailup.it/MailupSend.asmx?WSDL

MailUpSend vs. SMTP+

When it is better to use the MailUpSend Web service, and when does it make more sense to use SMTP+, MailUp's SMTP relay service? Here is a quick comparison.

MailUpSend	SMTP+
Recommended when... - You need to replicate the most common functionality of the MailUp admin console, outside of MailUp (e.g. creating a message and sending it) - You want to import data and schedule a mailing with a single call: the CSV can contain hundreds of thousands of addresses - You wish to use dynamic tags, dynamic content, and other advanced features - You need to send a message to a specific Group or using specific filters	Recommended when... - You are sending transactional emails - You need to rapidly integrate an external system with MailUp - You want to use MailUp as a high-deliverability SMTP service (e.g. from desktop email clients like Outlook) - You need to send messages that are so personalized that it would be difficult to create them in MailUp, even using dynamic tags and content

> ⓘ **SMTP+ and high volumes**
>
> Please note: if you need to send over 1 million emails per day with SMTP+, please contact us as a special setup may be required.

Authentication

> ⚠ "Login" method was removed from this document and is to be considered **deprecated** starting from MailUp release 7.4. It was replaced by **LoginFromId** method. If you are still using Login, don't worry: Login method is still available, even if not recommended for new integrations, and returns a response that is equal to the one that is returned by LoginFromId. Signature of old Login method is *string Login(string user, string pwd, string url)*, where URL is the console URL without http://

LoginFromId method

This method is available only from MailUp release 7.4 and it's very similar to the previous one (*Login*) but it allows to specify an account ID instead of a console URL. The main benefit lies in the fact that a URL limits the use of the infrastructure, whereas a fixed reference like the console ID makes it easier to move the administration console on a higher-performing hardware, if needed. There would be little or no malfunction, as the customer's applications would keep on working after the operation.

- string LoginFromId(string user, string pwd, int consoleId).
- user: username to access the MailUp console
- pwd: password to access the MailUp console
- consoleId: console identifier. The console ID can be drawn from the username associated to the console, without the initial "m" (e.g. username: m1234 > console ID 1234). In case it is not possible to get the console ID from the username, please contact our support team

> ⓘ Note: the **accessKey** returned by the *LoginFromId* method in the MailUpSend Web Service can also be used for the MailUpReport and MailUpManage Web Services.

ⓘ

SOAP Examples

SOAP Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:ws="http://services.mailupnet.
it/WS">
    <soap:Header/>
    <soap:Body>
       <ws:LoginFromId>
          <!--you can use special username ('a'+consoleID, whose password never expires) that is defined
in Manage/web services page-->
          <ws:user>a1234</ws:user>
          <ws:pwd>password</ws:pwd>
          <ws:consoleId>1234</ws:consoleId>
       </ws:LoginFromId>
    </soap:Body>
 </soap:Envelope>
```

SOAP Response

```
<LoginResult>
        <errorCode>0</errorCode>
        <errorDescription></errorDescription>
        <!--Use accessKey value that is returned by LoginFromId method-->
         <ws:accessKey>HzAgwRRJaAKBtkgNWpkAuURfV4SxMm6T3HJegRuSkUivKJElNNcmSQe8nqGyoM9</ws:accessKey>
</LoginResult>
```

Code Examples

PHP Sample

```
<?php
// The sample code described herein is provided on an "as is" basis, without warranty of any kind.
// MailUp shall not be liable for any direct, indirect or consequential damages or costs of any type
arising out of any action taken by you or others related to the sample code.
class MailUpWsSend {
        protected $WSDLUrl = "http://services.mailupnet.it/MailupSend.asmx?WSDL";
        private $soapClient;
        private $xmlResponse;
        protected $domResult;
        function __construct() {
                $this->soapClient = new SoapClient($this->WSDLUrl, array("trace" => 1, "exceptions" =>
0));
        }

        function __destruct() {
                unset($this->soapClient);
        }

        public function getFunctions() {
                print_r($this->soapClient->__getFunctions());
        }

        public function loginFromId() {
                try {
                        $loginData = array("user" => "user",
                                                "pwd" => "password",
```

```php
                                                "consoleId" => "idconsole");

                        $this->soapClient->loginFromId($loginData);
                        if ($this->readReturnCode("LoginFromId","errorCode") != 0) {
                                echo "<br /><br />Error in LoginFromId: ". $this->readReturnCode
("LoginFromId","errorDescription");
                                die();
                        }
                        else $this->accessKey = $this->readReturnCode("LoginFromId","accessKey");
                        echo "<br>AccesKey: ". $this->accessKey;
                        } catch (SoapFault $soapFault) {
                                var_dump($soapFault);
                        }
        }

        public function logout() {
                try {
                        $this->soapClient->Logout(array("accessKey" => $this->accessKey));
                        if ($this->readReturnCode("Logout","errorCode") != 0)
                                echo "<br /><br />Error in Logout". $this->readReturnCode("Logout","
errorDescription");
                        } catch (SoapFault $soapFault) {
                                var_dump($soapFault);
                        }
        }
        private function readReturnCode($func, $param) {
                $this->xmlResponse = $this->soapClient->__getLastResponse();
                $dom = new DomDocument();
                $dom->loadXML($this->xmlResponse) or die("(1)XML file is not valid!");
                $xmlResult = $dom->getElementsByTagName($func."Result");
                $this->domResult = new DomDocument();
                $this->domResult->LoadXML(html_entity_decode($xmlResult->item(0)->nodeValue)) or die("(2)
XML file is not valid!");
                $rCode = $this->domResult->getElementsByTagName($param);
                return $rCode->item(0)->nodeValue;
        }

        //... other functions...
        // public function functionName(...) {...}
}
?>
<html>
<head></head>
<body>
<?php
        $WsSend = new MailUpWsSend();
        $WsSend->loginFromId();
        // use $WsSend->functionName(...) to call other methods

        $WsSend->logout();
?>
</body>
</html>
```

Logout method

This method has become useless, the accessKey remains active for 60 minutes

- *string* Logout(*string* accessKey)
- *accessKey*: access key obtained by calling the Login method
- *LogoutResult:* string containing the outcome of the operation

SOAP Examples

```
<LogoutResult>
 <errorCode>0</errorCode>
 <errorDescription></errorDescription>
</LogoutResult>
```