



Automating Login to SmarterMail

Help Documentation

Automating Login to SmarterMail

Companies using SmarterMail can easily automate user entry into the mail application by configuring the auto-login functionality. The HTML code shown below demonstrates how you can make a text link on a website (e.g. "Log into your mail") that automatically logs a user in to the SmarterMail site. By putting this hidden form on a simple web page, you can fill in the "Domain", "Email Address", and "Password" information by hard-coding the data or through a scripting language like ASP, ASP.Net, or ColdFusion. This implementation of auto-login works seamlessly across domains, so the two applications do not have to be hosted on the same server.

Some notes about the example code listed below:

We have the form values set to generic text (e.g. "USERNAME_GOES_HERE") to show where you would hard code values that are submitted to the login page. You could also dynamically generate these values using a scripting language like ASP or ColdFusion. A sample ASP script would substitute `var domain = "USERNAME_GOES_HERE";` with `var domain = "<% =email %>"`.

The form action shown, `https://DOMAIN_GOES_HERE`, uses the default location of the Smartermail Web Interface. If you have created a separate web site for Smartermail or if you assign a different IP address for Smartermail within IIS, this action would have to be altered to reflect this change. For example, a modified form action might take the format of `https://mail.smartertools.com`. In addition, this code assumes that your SmarterMail site is secured with SSL/TLS. If it is not, be sure to change the 'var domain' URL from `https` to `http`.

Sample Code for SmarterMail 16.3.6855 and Higher

```
<!DOCTYPE html>
<html> <head> <meta charset="utf-8"> <script> function autoLogin() {
var domain = "https://DOMAIN_GOES_HERE"; var username =
"USERNAME_GOES_HERE"; var password =
"PASSWORD_GOES_HERE"; var xhr = new XMLHttpRequest();
xhr.open('POST', domain + '/api/v1/auth/authenticate-user');
xhr.setRequestHeader('Content-Type', 'application/json;charset=UTF-8');
xhr.onload = function() { if (xhr.status === 200) { var success =
JSON.parse(xhr.responseText); if (!success.success) { var res = "";
if(success.message) res = success.message else res = success.status
document.getElementById("errors").innerText = res; return; }
window.location.href = success.autoLoginUrl; } else {
document.getElementById("errors").innerText = failure.message || failure;
} }; xhr.send(JSON.stringify({ username: username, password: password,
```

```
retrieveAutoLoginToken: true })); } </script> </head> <body
onload="autoLogin()"> <div id="errors"></div> </body> </html>
```

```
Sample Code for SmarterMail 16.0.6345 - 16.3.6843 <html> <head> <meta
http-equiv="Content-Language" content="en-us"> <meta http-
equiv="ContentType" content="text/html; charset=windows-1252 ">
<title>Smartermail Login</title> <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
<style type="text/css"> body { margin: 0; overflow: hidden; } iframe {
width: 100%; height: 100%; border: none; } </style> <script
language="JavaScript"> if (window.navigator.userAgent.indexOf("Edge")
> -1) alert('Unsupported browser'); function WaitToMail() {
setTimeout(GoToMail, 250); }; function GoToMail() { var username =
"USERNAME_GOES_HERE"; var password =
"PASSWORD_GOES_HERE"; var domain =
"https://DOMAIN_GOES_HERE"; var autoLogin = true; var
targetWindow =
document.getElementById('loginWindow').contentWindow;
window.addEventListener("message", function () { window.location.href =
domain + "/interface/root#/login"; }, false); targetWindow.postMessage({
'username': username, 'password': password, 'autoLogin': autoLogin },
"*"); } </script> </head> <body> <iframe id="loginWindow"
src="https://DOMAIN_GOES_HERE"
onload="WaitToMail()"></iframe> </body> </html>
```

The code above would automatically log the user into the SmarterMail interface. To auto-fill the user's login credentials instead, change var autoLogin = true; to var autoLogin = false; .

Also, please note that Microsoft Edge is not a supported browser for automating user login with this script because this version of the auto-login functionality relies on being able to set sessionStorage of the iframe and have that storage carry over in the redirect. In Edge, iframes have their own special storage that doesn't carry over to the actual site. Instead, Edge can have auto-login available through a bookmarklet, using the following code: javascript: window.postMessage({username: "USERNAME_GOES_HERE", password: "PASSWORD_GOES_HERE", autoLogin: true}, "*");