SecurEnvoy
A Shearwater Group plc Company

www.securenvoy.com

RESTful API

SecurAccess RESTful API Guide

SecurAccess RESTful API Guide

Contents

1.1 Solution Summary

SecurEnvoy's SecurAccess MFA solution offers external web-based systems the facility to read, write and control user configuration and authentication using a RESTful API.

1.2 Guide Usage

The information in this guide describes the configuration required to setup access to the REST API, including the many POST commands that can be used against the API.

1.3 Prerequisites

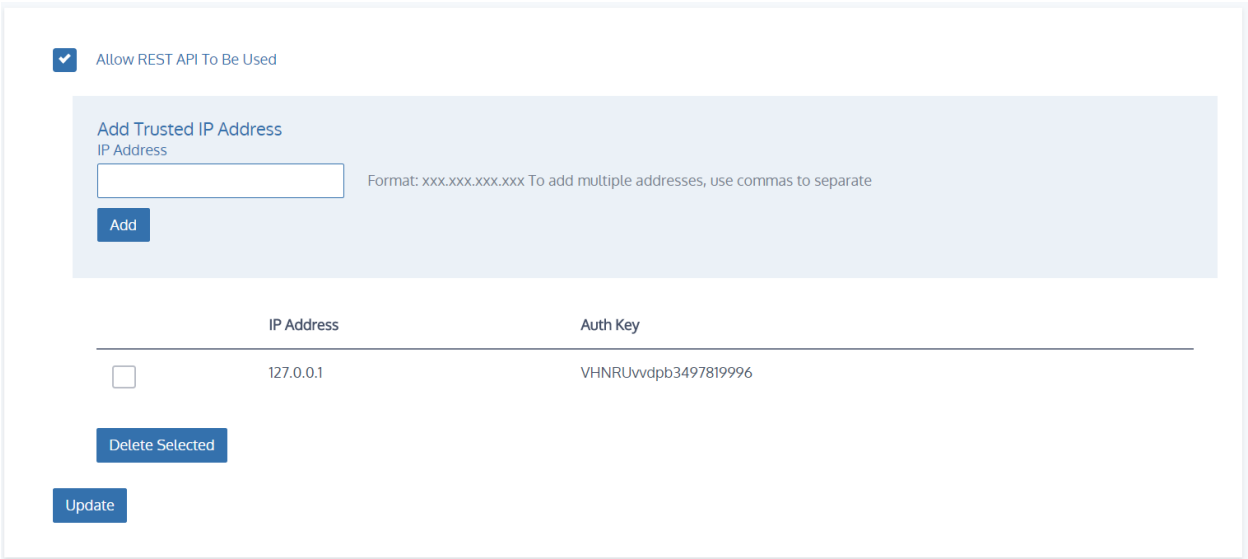
The following conditions are required to set up SecurEnvoy's MFA Solution:

- A SecurAccess MFA server installed, configured and working on a system with:
  - Windows Server 2003 or higher.

  Note: Please see SecurEnvoy's SecurAccess deployment guide on how to setup MFA server solution.
- SecurAccess Version 9.1 or above

1.4 Setup

Each application that will communicate to the SecurAccess server via the REST API interface will require authorisation as a trusted source,
located under “Config/REST API” from the SecurEnvoy Administrator Portal.



AuthKey - This Authkey is generated when each trusted source is added. Each source that connects to the REST API will need use the Authkey in a BASE 64 encoded format.

1.5 Available Resources

URL	HTTP Verb	Functionality
/secrest/api/users/<DomainNumber>/<UserID>	GET	Retrieve details of a specific user
/secrest/api/users	POST	Create a user (Only usable on a SecurEnvoy Internal Managed Users (Microsoft LDS) Domain)
/secrest/api/users/<DomainNumber>/<UserID>	PUT	Update a user
/secrest/api/users/<DomainNumber>/<UserID>	DELETE	Deletes a user (Only usable on a SecurEnvoy Internal Managed Users (Microsoft LDS) Domain)

1.41 Get Users

Description	Retrieve Details of a User
URL	/secrest/api/users/<DomainNumber>/<UserID>
Method	GET
URL Parameters	Required DomainNumber = <string> - The domain number of the domain that the user exists in. UserID = <string> - The UserID of the required user.
Data Parameters	None
Success Response	Code: 200 Content:: {"UserID":"user1","Firstname":"Test","Lastname":"User", … } Returns details of requested user
Error Response	Code: 401 UNAUTHORIZED Content: "ERR, No Auth Key Received" Code: 200 OK Content: "ERR, LDAP Connect Returned An invalid dn syntax has been specified"

Sample Call	var userid = 'jsmith';
	var domainnumber = '1';
	var authkey = 'a1a1a1a1a1a1a1a1a1a1'
	$.ajax({
	url: '/secrest/api/users/' + domainnumber + '/' + userid,
	type: 'GET',
	beforeSend: function (xhr) {
	xhr.setRequestHeader("Accept", "application/json");
	xhr.setRequestHeader("Authorization", 'APIKEY ' + btoa(authkey));
	},
	success: function (result) {
	console.log(result);
	},
	error: function (result) {
	console.log(result)
	}
	});

1.42 Add Users

Description	Create a user (Only usable on a SecurEnvoy Internal Managed Users (Microsoft LDS) Domain)
URL	/secrest/api/users
Method	POST
URL Parameters	None
Data Parameters	Required DomainNumber = <string> - The domain number of the domain that the user exists in. UserID = <string> - The UserID of the required user Optional Various user parameters, see 'Data Parameters' below for full list of available parameters.
Success Response	Code: 200 Content:: {"UserID":"user1","Firstname":"Test","Lastname":"User", … } Returns details of created user
Error Response	Examples: Code: 401 UNAUTHORIZED Content: "ERR, No Auth Key Received" Code: 200 OK Content: "ERR, LDAP Connect Returned The object already exists." Code: 200 OK Content: ""ERROR Mobile Number Required." Other errors will be returned in the format: Code: 200 OK Content: "Err, <Error description here>

Sample Call	var authkey = 'a1a1a1a1a1a1a1a1a1a1'
	var data = {"Firstname":"John","Lastname":"Smith","Mobile":"1234"}
	$.ajax({
	url: '/secrest/api/users',
	type: 'POST',
	beforeSend: function (xhr) {
	xhr.setRequestHeader("Accept", "application/json");
	xhr.setRequestHeader("Authorization", 'APIKEY ' + btoa(authkey));
	}
	data:data,
	success: function (result) {
	console.log(result)
	},
	error: function (result) {
	console.log(result)
	}
	});

1.43 Update User

Description	Update a user
URL	/secrest/api/users/<DomainNumber>/<UserID>
Method	PUT
URL Parameters	None
Data Parameters	Required DomainNumber = <string> - The domain number of the domain that the user exists in. UserID = <string> - The UserID of the required user Optional Various user parameters, see 'Data Parameters' below for full list of available parameters.
Success Response	Code: 200 Content:: {"UserID":"user1","Firstname":"Test","Lastname":"User", … } Returns details of created user
Error Response	Examples: Code: 401 UNAUTHORIZED Content: "ERR, No Auth Key Received" Code: 200 OK Content: ""ERROR Mobile Number Required." Other errors will be returned in the format: Code: 200 OK Content: "Err, <Error description here>

Sample Call	var userid = 'jsmith';
	var domainnumber = '1';
	var authkey = 'a1a1a1a1a1a1a1a1a1a1'
	var data = {"Firstname":"John","Lastname":"Smith","Mobile":"1234"}
	$.ajax({
	url: '/secrest/api/users/' + domainnumber + '/' + userid,
	type: 'POST',
	beforeSend: function (xhr) {
	xhr.setRequestHeader("Accept", "application/json");
	xhr.setRequestHeader("Authorization", 'APIKEY ' + btoa(authkey));
	}
	data:data,
	success: function (result) {
	console.log(result)
	},
	error: function (result) {
	console.log(result)
	}
	});

1.41 Delete User

Description	Deletes a user (Only usable on a SecurEnvoy Internal Managed Users (Microsoft LDS) Domain)
URL	/secrest/api/users/<DomainNumber>/<UserID>
Method	DELETE
URL Parameters	Required DomainNumber = <string> - The domain number of the domain that the user exists in. UserID = <string> - The UserID of the required user
Data Parameters	None
Success Response	Code: 200 Content:: "User deleted successfully"
Error Response	Examples: Code: 401 UNAUTHORIZED Content: "ERR, No Auth Key Received" Other errors will be returned in the format: Code: 200 OK Content: "Err, <Error description here>

Sample Call	var userid = 'jsmith';
	var domainnumber = '1';
	var authkey = 'a1a1a1a1a1a1a1a1a1a1'
	$.ajax({
	url: '/secrest/api/users/' + domainnumber + '/' + userid,
	type: DELETE,
	beforeSend: function (xhr) {
	xhr.setRequestHeader("Accept", "application/json");
	xhr.setRequestHeader("Authorization", 'APIKEY ' + btoa(authkey));
	},
	success: function (result) {
	console.log(result);
	},
	error: function (result) {
	console.log(result)
	}
	});

1.42 Authenticate User (1st Step)

Description	Authenticate the user with userid and password, receive either “success” with qr code data, or “challenge” with a session key
URL	/secenrol/
Method	POST
URL Parameters	None
Data Parameters	action = GETQRONLY userid = <string> - The UserID of the user to enrol PASSWORD = <string> - The password/PIN of the user integrationmode = true

Success Responses	Challenge
	{
	"result":"challenge",
	"session":"<session key>",
	"userid":"<the userid of the user>"
	}
	
	Success (will only happen with soft token push login)
	{
	"result":"success",
	"base64image":"<base 64 encoded image>",
	"seed":"ABCD1234",
	"enrolurl":"<url>",
	"domain":"1"
	}

Cookie Returned	SecurEnvoyPin=<cookie content> (this is only returned with a “success” result)

Error Responses	Access Denied
	{
	"result":"accessdenied"
	}
	
	Unexpected Error
	{
	"result":"error",
	"message":"<error message>"
	}

Sample Call	POST /secenrol/ HTTP/1.1 Host: www.host.com Content-Type: application/x-www-form-urlencoded Cache-Control: no-cache action=GETQRONLY&userid=bob&PASSWORD=secretpassword& PASSCODE=123456&integrationmode=true
Notes	If the authentication is successful (a status of “success”), this request will return a cookie named SecurEnvoyPin. This cookie should be sent with every request after this until enrolment is completed.

1.43 2nd Authentication Step and Get QR Code

Description	Authenticate the user with userid, session key and passcode and get information back to enable enrolment to softtoken
URL	/secenrol/
Method	POST
URL Parameters	None
Data Parameters	action = GETQRONLY userid = <string> - The UserID of the user to enrol SESSION = <string> - The session key returned in the last step PASSCODE = <string> - The one time passcode of the user integrationmode = true

Success Response	{
	"result":"success",
	"base64image":"<base 64 encoded image>",
	"seed":"ABCD1234",
	"enrolurl":"<url>",
	"domain":"1"
	}

Cookie Returned	SecurEnvoyPin=<cookie content>

Error Responses	Access Denied
	{
	"result":"accessdenied"
	}
	
	Unexpected Error
	{
	"result":"error",
	"message":"<error message>"
	}

Sample Call	POST /secenrol/ HTTP/1.1 Host: www.host.com Content-Type: application/x-www-form-urlencoded Cache-Control: no-cache action=GETQRONLY&userid=bob&PASSWORD=secretpassword& PASSCODE=123456&integrationmode=true
Notes	If the authentication is successful, this request will return a cookie named SecurEnvoyPin. This cookie should be sent with every request after this until enrolment is completed.

1.44 Poll Server for Auto Enrolment Completion

Description	Poll the enrolment server to check if the phone has completed auto enrolment
URL	/secenrol/
Method	POST
URL Parameters	None
Data Parameters	action = QUERYSOFTTOKEN seed = <string> - The seed passed back from the previous request integrationmode = true
Success Responses	CONTINUE – this is returned when auto enrol has not yet completed, you should keep polling until OK is returned or the "CheckCode" is manually entered OK - this is returned when auto enrolment has completed successfully
Error Responses	<Error Message>
Sample Call	POST /secenrol/ HTTP/1.1 Host: www.host.com Content-Type: application/x-www-form-urlencoded Cache-Control: no-cache Cookie: SecurEnvoyPIN=<cookie content from previous response> action=QUERYSOFTTOKEN&seed=ABCD1234&integrationmode=true
Notes	You must include the cookie SecurEnvoyPin, returned from the authentication request in this request

1.45 Manually Send Check Code

Description	Manually send check code from phone if auto enrolment is not possible
URL	/secenrol/
Method	POST
URL Parameters	None
Data Parameters	action = SETINFO domain = <string> - The domain number returned from the authentication request tokentype = softtoken SOFTTOKENURL = <string> - The "enrolurl" returned from the authentication request CHECKCODE = <string> - The code displayed on the user's phone if auto enrolment fails integrationmode = true

Success Response	{
	"result":"success"
	}

Error Responses	Access Denied
	{
	"result":"accessdenied"
	}
	
	Unexpected Error
	{
	"result":"error",
	"message":"<error message>"
	}

Sample Call	POST /secenrol/ HTTP/1.1 Host: www.host.com Content-Type: application/x-www-form-urlencoded Cache-Control: no-cache Cookie: SecurEnvoyPIN=<cookie content from previous response> action=SETINFO&domain=1&tokentype=softtoken&CHECKCODE=ABCD1234& SOFTTOKENURL=<enrol url>
Notes	You must include the cookie SecurEnvoyPin, returned from the authentication request in this request

1.6 Authentication

To authenticate against the API, you will need to send an HTTP Authorization Header containing an Auth Key with each request.

The Auth Key is generated from the SecurEnvoy Administrator Portal and is specific to each trusted client based on IP Address.

This is located under “Config/REST API” in the SecurEnvoy Administrator Portal.

Set the Authorization header to a value of APIKEY followed by a space, followed by a Base 64 Encoded string of the relevant Auth Key for your IP Address; obtained from the SecurEnvoy Administrator Portal as above.

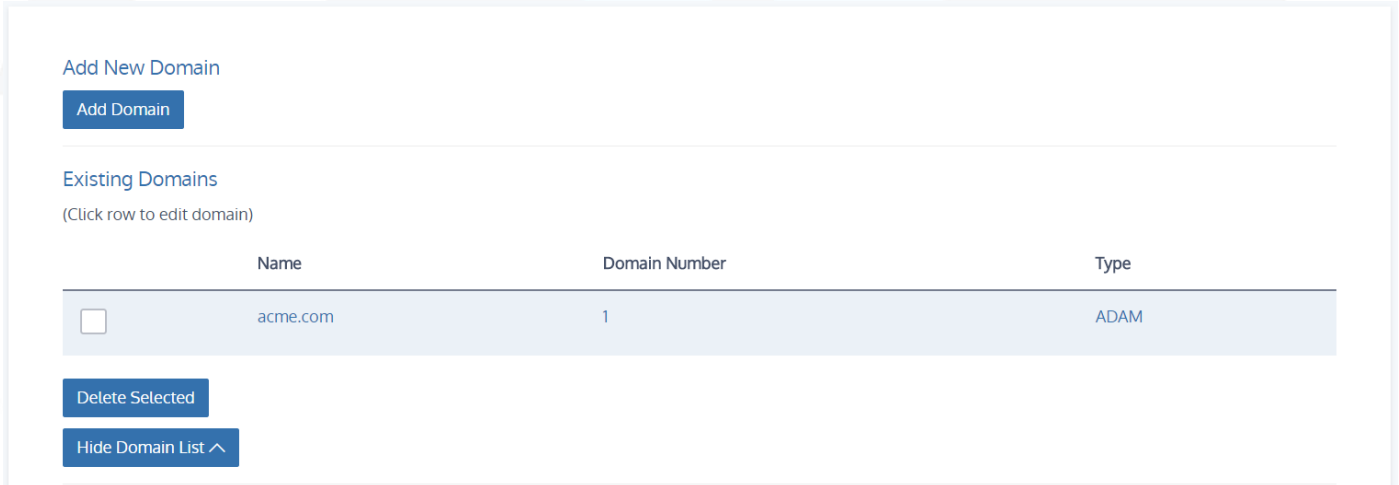
Example
If your Auth Key was a1a1a1a1a1a1a1a1a1a1, your Authorization header would look like the following:
Authorization: APIKEY UVJUVFJraGxrcjcwNjI3OTY0MjY=

The following site can be used to Base 64 Encode the Authkey for use https://www.base64encode.org/

1.7 Required Parameters

DomainNumber - This corresponds to the number associated with your domain when multiple domains have been specified in SecurAccess.

This can be located from the SecurEnvoy Administrator Portal under “Domains”



UserID - This is the unique identifier for each user. This will usually be the username of the user.

1.8 Optional Data Parameters

The optional parameters that can be specified are as follows:

Name	Type	Description
Firstname	String	only be used if the server is setup for SecurEnvoy managed users the users first Name
Lastname	String	only be used if the server is setup for SecurEnvoy managed users the users last Name (given name)
Email	String	The user's SMTP email address
Mobile	String	The user's mobile phone number
PIN	String	Only applies when PIN management is set to SecurEnvoy The PIN of this user (set to stars of the same length when listing user)
toSend	Enum	MOBILE Passcode is sent via SMS to mobile phone EMAIL Passcode is sent via SMTP email Defaults to MOBILE
RandomPINBox	Boolean	Defaults to false, generates a random pin number if set true, overwrites a manually entered PIN number
Seed	Boolean	Soft Token Only, if set to True, creates the first part seed record
ResetFail	Boolean	If set to True resets failed authentication count to 0, only works when called in UPDATE
HelpDeskEnrol	Boolean	If set to True user needs to enrol with secret questions
HiddenMobile	Boolean	If set to True, the mobile number is encrypted and stored in securenvoy's attribute. If set to False, the LDAP attribute "mobile" is used for mobile numbers
MobileEnrol	Boolean	If set to True user needs to enrol their mobile phone number
AllowOffline	Boolean	This applies if Windows Login Agent is deployed or when Windows Integrated Desktop Management is enabled. If set to True this user has a laptop which will require authentication when logging in offline
ResendToken	Boolean	If set to True, resends the passcode after UPDATE is called
ResyncToken	Boolean	Resync Soft Token with codes set in Resync1 & Resync2
SimpleSMS	Boolean	If set to True, SMS messages will not overwrite or flash. Required for some international SMS gateways to prevent dropping the request
SoftTokenEnrol	Boolean	If set to True user needs to enrol their soft token app
TempUser	Boolean	If Set to True User is temporary and will unmanage (or delete if LDAP type is ADAM) after TmpUserDays
Unmanage	Boolean	If set to True, unmanages this user when UPDATE is used and releases a license.
VoiceEnrol	Boolean	If set to True user needs to enrol their phone number
Weekend	Boolean	Only applies to users in Mode DAYCODE If set to True daycode are sent at weekends (Saturday & Sunday)

Name	Type	Description
LastLogIn	String	This is set after using GET or UPDATE and is a text string of the users last login status – should be read only
FailedAttempts	String	The number of failed authentications since the last good one
toAuth	Enum	ONETIME - Preload OTP Each passcode can only be used once THREECODES - Three Preloaded OTP's in each message REALTIME - Passcode Sent in real time after PIN is authenticated SOFTTOKEN - uses soft token apps on smart phones DAYCODE - Each passcode can be reused up to NumDays * 2 VOICECALL - VOIP call, user enters the displayed OTP into the phone STATICPW - Passcode is static and never changes TMPPW - tmp code that switches back to onetime after TmpDays NONE – Use this if a Yubikey is assigned and no other authentication method is wanted. Defaults to ONETIME
AdminLVL	Enum	ADMIN The user has full admin rights HELPDESK The user has helpdesk only admin rights NONE The user has no admin rights UNCHANGED Do not change the existing admin leave Defaults to NONE
UserEnabled	Enum	DISABLED The user is disabled and can't authenticate ENABLED The user is enabled and can authenticate OK ICE The user can only authenticate when ICE is enabled Defaults to ENABLED
Resync1	String	Soft tokens only, the first 6-digit passcode when ResyncToken is True
Resync2	String	Soft tokens only, the first 6-digit passcode when ResyncToken is True
EnrolURL	String	Soft tokens only, this string is set when Seed is set True
TempPass	String	Only applies to users in Mode TMPPW (Temporary Password) The temporary passcode to authenticate with.
StaticPass	String	Only applies to users in Mode STATIC The static passcode to authenticate with
NumDays	Integer	Only applies to users in Mode DAYCODE the number of days between each passcode being sent
TmpDays	Integer	Only applies to users in Mode TMPPW (Temporary Password) The number of days before switching back to One Time Passcode
TmpUserDays	Integer	Only applies to users that have TempUser set to True The number of days before this user is un-managing or deleted if LDAP type is ADAM
YubikeyEnrol		If you want to assign a Yubikey to the user, enter the code generated by the Yubikey here. By default this will be used as an alternative authentication method to the one specified by "toAuth". If you want to use this as the sole method, set "toAuth" equal to "NONE".

1.9 Response Type

The program allows you to request the response information in two formats: JSON or XML. This is determined in the header you use in your web client, e.g.:

```
Accept: application/json
```

Please Reach Out to Your Local SecurEnvoy Team...

UK & IRELAND

The Square, Basing View
Basingstoke, Hampshire
RG21 4EB, UK

Sales

E sales@SecurEnvoy.com
T 44 (0) 845 2600011

Technical Support

E support@SecurEnvoy.com
T 44 (0) 845 2600012

EUROPE

Freibadstraße 30,
81543 München,
Germany

General Information

E info@SecurEnvoy.com
T +49 89 70074522

ASIA-PAC

Level 40 100 Miller Street
North Sydney
NSW 2060

Sales

E info@SecurEnvoy.com
T +612 9911 7778

USA - West Coast

Mission Valley Business Center
8880 Rio San Diego Drive
8th Floor San Diego CA 92108

General Information

E info@SecurEnvoy.com
T (866)777-6211

USA - Mid West

3333 Warrenville Rd
Suite #200
Lisle, IL 60532

General Information

E info@SecurEnvoy.com
T (866)777-6211

USA – East Coast

373 Park Ave South
New York,
NY 10016

General Information

E info@SecurEnvoy.com
T (866)777-6211

SecurEnvoy
A Shearwater Group plc Company

www.securenvoy.com