

HelpdeskAPI

Contents

- 1 Introduction
- 2 PRINCIPLES
 - ◆ 2.1 Sending an Admin API request
- 3 Helpdesk API
 - ◆ 3.1 Reset
 - ◆ 3.2 Strings
 - ◆ 3.3 Sync Token
 - ◆ 3.4 Update
 - ◆ 3.5 Purge
- 4 Responses
- 5 Error Messages

Introduction

Swivel has developed an API, the [User Admin API](#), to allow an external application to perform CRUD (Create, Read, Update, Delete) style operations on users. For most installations, these operations are performed by the User Synchronization job but for larger user populations synchronization may be impractical. The User Admin API addresses this need. In conjunction with the User Admin API, Swivel has also developed another API allowing external applications to perform the functions usually performed by a helpdesk operator from the admin console. These are typically day to day functions not viewed as part of the User Admin API such as user unlock, PIN reset etc. This is the **Helpdesk API**

There is additionally read-only functionality provided by the [Reporting API](#) that can be used to read the status (eg idle, locked) of user accounts.

PRINCIPLES

Following the principles behind the existing Agent API used for authorisation, both of the API's are based around XML documents for both request and reply. The basic idea is to build a document containing details of operation(s) to be performed and the user(s) on which they are to be performed, submit it to PINsafe via HTTP and PINsafe will reply with a document detailing which operations succeeded and which, if any, failed. All operations via the API will be logged in the standard PINsafe log. Only authorised PINsafe Agents will be able to submit requests. This is in line with Agents used for authorisation. The API is case sensitive, the correct case shown in the included examples.

In order to use the Admin API, you must create a PINsafe Agent for the computer, or sub-net of computers, that will execute AdminAPI commands. Only the computer(s) defined by this Agent can create, read, update or delete users belonging to this Agent, and the Agent cannot read, update or delete users created by other repositories. The Agent can therefore be regarded as another type of repository. It is possible, but not recommended, to manipulate users created by a normal (e.g. XML or Active Directory) repository, by giving the Agent exactly the same name as the repository. Be aware if you do this, however, that any subsequent User Sync of the repository will potentially overwrite modifications made by the Agent. At present, any PINsafe Agent can act as an AdminAPI repository, but future versions of PINsafe will require you to explicitly enable the "Act as Repository" property of an Agent.

These restrictions do not apply to HelpdeskAPI Agents. Although only Agents can use these commands, it is possible to specify that the HelpdeskAPI command operates on a different repository, or on all repositories.

Sending an Admin API request

An Admin API request must be sent as an HTTP(S) request to the pinsafe application, either as a GET or POST request.

For a GET request, the format is

```
https://[swivelserver]:8080/pinsafe/AdminXML?xml=[request body]
```

For a POST request, the URL is

```
https://[swivelserver]:8080/pinsafe/AdminXML
```

and the XML request should form the content of the POST.

Helpdesk API

An example Helpdesk request is shown below;

```
<?xml version="1.0" ?>
<HelpdeskRequest secret="MyHelpdeskAgent" version="3.4">
  <Reset repository=?repository1?>
    <User name="bob"/>
  </Reset>
</HelpdeskRequest>
```

The above example shows a request to reset bob's credentials and send them out. Obviously bob must have a valid alert transport defined. As with all Agent-XML requests, the HelpdeskRequest element contains a ?secret? attribute.

The Helpdesk API is more restrictive than the Admin API; only the following operations are supported :

- - ◆ Reset ? reset user credentials.
 - ◆ Strings ? send new security strings to the user.
 - ◆ Update ? update user Credentials or Policy.

As with an AdminRequest, all of these operations can take a list of users so multiple users can be supplied in a single request and again there are no specific ordering requirements although as before, obviously, users must have been Created (possibly with an appropriate AdminRequest) before operations can be performed on them.

Also as with the AdminRequest, a HelpdeskRequest is posted to the AdminXML servlet and the response received in the same way. All Helpdesk operations support the (optional) ?repository? attribute, used to specify which repository the user belongs to. If not supplied, the user is assumed to reside in the repository belonging to the calling Agent. A breakdown of the individual operations follows.

Reset

Reset user credentials and send them to the user. Supply only the username(s), nothing else is required or allowed (Note: the user must have a transport to receive the new PIN).

```
<Reset repository=?repository1?>
  <User name="bob"/>
  <User name="joe"/>
  <User name="ann"/>
</Reset>
```

Strings

Send security strings to the user, ie via their assigned transport. Supply only the username, nothing else is required or allowed.

```
<Strings>
  <User name="ann"/>
</Strings>
```

Sync Token

Two consecutive one-time passwords are entered in order to synchronise the token with the Swivel server. Can be used of either HOTP or TOTP tokens, but probably more likely to be required for HOTP (or event-based) tokens.

```
<OathSync>
  <User name="tokenuser"/>
  <OTP1>123456</OPT1>
  <OTP2>456789</OPT2>
</OathSync>
```

If the Synchronisation is successful a pass message is returned.

Possible fail messages are

OATH_SYNC_FAILURE : The token could not be synced using the submitted codes

OATH_TOKEN_NOT_FOUND_FOR_USER : The user does not have a token

OATH_SEED_ERROR : There is an invalid seed associated with the token

Update

Update user details. See [Admin API](#) for valid sub-elements

```
<Update>
  <User name="bob">
    <Credentials password=?mydogsname?/>
    <Policy pinNeverExpires="false"/>
    <Oath SerialNumber="12345"/>
  </User>
</Update>
```

All of the above operations support the optional ?repository? attribute used to specify which repository the user belongs to.

Purge

Purge deleted users from a named repository:

```
<PurgeDeleted repository="repos-name" />
```

The repository attribute is the only content permitted.

The response will contain the number of users purged.

Responses

Responses to helpdesk requests follow the same structure as that for [admin requests](#). They consist of a HelpDeskResponse element containing the actions completed. eg for the above update request the response would be

```
<?xml version="1.0" ?>
<HelpdeskResponse>
  <Update>
    <User name="bob" />
  </Update>
</HelpdeskResponse>
```

If the request fails for any reason then a FAIL is included in the the User element. Any additional information why the command failed maybe in the PINsafe server logs.

```
<?xml version="1.0" ?>
<HelpdeskResponse>
  <Update>
    <User name="bob">FAIL </User>
  </Update>
</HelpdeskResponse>
```

A response may be a mixture of successes and fails, so in the example above where a request was made to reset the credentials of bob, joe and ann may get the response through their alert transport.

```
<?xml version="1.0" ?>
<HelpdeskResponse>
  <Reset>
    <User name="bob"/>
    <User name="joe">FAIL</User>
    <User name="ann"/>
  </Reset>
</HelpdeskResponse>
```

Error Messages

Refer to the [Admin API](#) article for a list of error responses.