AuthenticationAPI

The Swivel Authentication API (Agent-XML) is a means by which external application can make authentication requests to Swivel.

The API is XML-Based and is a subset of the overall Agent-XML API.

Contents

Background

All Agent-XML requests include a shared secret (with the exception of ping). In order for an authentication API to be acted upon by Swivel, the IP addressed and shared secret presented within the request must match that configured on the Swivel server.

All requests also include a version number. This should be set according to the target version of Swivel:

- 3.1 for all versions of Swivel from 3.1 - 3.3.
- 3.4 for versions 3.4 and 3.5.
- 3.6 for versions 3.6 onwards.

(NOTE: the value of version is not actually checked, but the element must be present).

Authentication requests must be sent via an HTTP Post to the Swivel server, to the AgentXML context.

For example http://<ip address>:8080/pinsafe/AgentXML

Note that for appliances this will be https by default. Agent XML requests are sent direct to the Swivel application on port 8080 and not via the proxy.

There is a optional field called requestID. If this is included then it will be echoed back in the corresponding response.

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<RequestID>1000</RequestID>
       .
       .
</SASRequest>

<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
 <RequestID>1000</RequestID>
       .
       .
</SASResponse>
```

Starting a Session

Some Swivel authentication modes are session based.

In this mode an Agent must start a session for a user and then request the security string for that session, either via an Image (Single Channel How To Guide Single Channel, eg TURing) or a message(Dual Channel, eg SMS message).

When an agent starts a session an ID for that session is returned. This session ID can then be used to request the string either via an image

img src = http://pinsafe:8080/pinsafe/SCImage?sessionid=3bfwuefi37tr

or via a message

img src = http://pinsafe:8080/pinsafe/DCMessage?sessionid=3bfwuefi37tr

The a session start message is as follows

The Agent gathers the user's username and sends the following XML Request to the Swivel Server:

The above XML Request will start a session on the Swivel Server, which will respond with the following XML:

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
```

```
<Secret>shared_secret</Secret>
<Action>sessionstart</Action>
<Username>some_user</Username>
</SASRequest>
```

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
<SessionID>c7379ef1b41f90a4900548a75e13f62a</SessionID>
</SASResponse>
```

If for any reason the session start fails a fail message will be return along with the reason for the failure, for example

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>FAIL</Result>
<Reason>AGENT_ERROR_NO_USER_FOUND</Reason>
</SASResponse>
```

A list of possible errors is shown a in the Error Messages Section below.

Authentication

The following example shows how to perform a login using the Agent XML Interface. The login request is used for both Single Channel and Dual Channel logins.

The Agent gathers the user's username,optional password, One-Time-Code and sends the following XML Request to the Swivel Server:

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Action>login</Action>
<Username>some_user</Username>
<Password>password</Password>
<OTC>1234</OTC>
</SASRequest>
```

If the authentication request is successful the server will respond with the following XML:

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
</SASResponse>
```

If there are any problems the Result element will contain a FAIL and if an error has occurred there will be an Error element, for example:

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>FAIL</Result>
<Error> AGENT_ERROR_NO_SECURITY_STRINGS</Error>
</SASResponse>
```

If the Result element?s value is FAIL and there is no Error value the user has simply failed to authenticate successfully, ie supplied incorrect credentials.

Authentication By Attribute

The standard login request must contain the users Swivel Username.

However some services may require another atttribute to the used, eg email address. There is a separate API call to handle this case whereby the agent provides the attribute name and attribute value is part of the authentication request.

```
""Note that the Swivel Authentication Platfrom will need to be configured to allow other attributes to be used in this way""
```

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Action>login</Action>
<Username>some_user@domain.com</Username>
<Attribute>email</Attribute>
<Password>password</Password>
<OTC>1234</OTC>
</SASRequest>
```

Warnings

A successful authentication request can include warnings, these usually refer to the fact that a PIN needs to be change due to a policy that is set or due to the fact that it will soon expire. These warnings can be used by the agent to re-direct the user to a change-PIN page after authentication.

<?xml version="1.0" ?>

```
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
<Warning>AGENT_WARN_CHANGE_PIN</Warning>
</SASResponse>
```

Check Password

from Version 3.10

It is possible to use Agent XML to check a users password. This will either be the user's Swivel password or their repository password depending on the policy set for this agent. The attempt will be logged but an incorrect password will not be treated as a failed authentication attempt

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Action>checkpassword</Action>
<Username>some_user</Username>
<Password>password</Password>
</SASRequest>
```

Reponse is a simple pass or fail.

Change PIN

Change PIN is very similar to a login request with the inclusion of the user?s new password and new PIN, sent has a new One-Time-Code. The Agent gathers the username, existing password, new password, existing One-Time-Code, and new One-Time-Code. To start the change PIN process the following XML is sent to the Swivel Server:

If no password is being used the <Password> and <NewPassword> elements should be empty.

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Action>changepin</Action>
<Username>some_user</Username>
<Password>password</Password>
<NewPassword>newpassword</NewPassword>
<OTC>1234</OTC>
<NewOTC>4321</NewOTC>
</SASRequest>
```

If the request is successful the Swivel Server will change the user's password and PIN, and will respond with a PASS in the Result element:

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
</SASResponse>
```

The change pin can fail for a number of reasons. If the password and one-time code submitted are incorrect the response will just indicate a failure. If the current credentials were correct, then the pin change may fail because the new PIN does not confirm to PIN composition policies, such as not allowing sequences such as 1234.

<?xml version="1.0" ?>

```
<SASResponse>
<Version>3.6</Version>
<Result>FAIL</Result>
<Error>AGENT_ERROR_PIN_COMPOSITION</Error>
</SASResponse>
```

Security Strings

This API is also used by a midlet to request a batch of security strings.

For Versions of Swivel prior to Version 3.8 the format is as follows

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Action>securitystrings</Action>
<Username>some_user</Username>
</SASRequest>
```

For Version 3.8 the format is

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Action>SecurityStrings</Action>
<Id>client-id</Id>
</SASRequest>
```

In this version the mobile client needs to provide a unique user-id in order to receive security strings. The mobile client obtains this client-id by completing the provision process described below.

The response is a set of 99 security strings

Mobile Client Provision (Version 3.8)

In order for a mobile client to download security strings it needs to obtain a unique client-id. This is obtained by submitting a provision code to Swivel. This provision code will be sent to the user, usually via a text message.

There is an API call that will request a provision code to be sent to the user. (Or this can be done via the admin console)

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
```

```
<Action>provisioncode</Action>
<Username>some_user</Username>
</SASRequest>
```

If successful a Pass packet will be returned and a provision code will be sent to the end-user.

If not successful a FAIL will be returned along with any error.

The user will then enter the provision code. The provision code will then be presented to Swivel by the client as

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Action>provision</Action>
<ProvisionCode>code</ProvisionCode>
<Username>username</Username>
</SASRequest>
```

If the code is correct Swivel will return a PASS along with the clients Unique Client ID (UCID)

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
<Id>id</Id>
</SASResponse>
```

If there is a problem, then an error will be returned

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>FAIL</Result>
<Error>AGENT_NO_SESSION</Error>
</SASResponse>
```

Ping

You can use a ping command to test that the Swivel application is available. The response is a pass response.

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Action>ping</Action>
</SASRequest>
```

The response being

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
</SASResponse>
```

Reset

If it is enabled on the Swivel server, a user can request a reset code to be sent to them, then of they enter that reset code, they are sent a new PIN (Note: the user must have a transport to receive the new PIN).

Two API commands support this, <Resetcode> that sends the code to the user and then <Reset> that submits the reset code to Swivel.

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Action>resetcode</Action>
<Username>some_user</Username>
</SASRequest>
```

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
</SASResponse>
```

This sends the reset code to the user. The user enters the code on a form on the agent and submits.

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Action>reset</Action>
<Username>some_user</Username>
<Resetcode>87456hfiu7634</Resetcode>
</SASRequest>
```

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
</SASResponse>
```

If the reset code is entered correctly then the user is sent a new set of credentials.

User Exists

This method can be used as a pre-authentication check to see if an account exist on Swivel.

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Action>exists</Action>
<Username>some_user</Username>
</SASRequest>
```

If the user exists a Pass is returned, if the user does not exist a Fail is sent back.

```
<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
</SASResponse>
```

User Exists By Attribute

A variation of the above method allows the query to pass in an attribute, eg email address.

The response to the query will be the name of the attribute that matched if the user could be found

Or fail if not match (or multiple matches) were found

eg

```
<?xml version="1.0" ?>
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Action>ExistsByAttribute</Action>
<Username>firstname.lastname@domain.com</Username>
</SASRequest>

<?xml version="1.0" ?>
<SASResponse>
<Version>3.6</Version>
<Result>PASS</Result>
<AtttributeName>email</AttributeName>
</SASResponse>
```

Token Synchonisation

This API calls attempts to synchronise a user's token but submitting two consecutive OTPs

```
<SASRequest>
<Version>3.6</Version>
<Secret>shared_secret</Secret>
<Username>tokenuser</Username>
<Action>OathSync</Action>
<OTP1>481262</OTP1>
<OTP2>579024</OTP2>
</SASRequest>
```

If the request is successful a PASS message will be returned. If not the possible failures are SYNC_FAILURE (meaning the OTPs were not valid) or OATH_TOKEN_NOT_FOUND (meaning the user does not have a token.

Token Challenge-Response

This API call validates the users response to an OCRA challenge. (Where the user enters the challenge on the OCRA token keypad)

```
<SASRequest>
<Version>3.1</Version>
<Secret>shared_secret</Secret>
<Username>tokenuser</Username>
<Action>OcraVerify</Action>
<OcraChallenge>8765432</OcraChallenge>
<OcraResponse>12345678</OcraResponse>
</SASRequest>
```

If the respinse is verfied a PASS message will be returned. If not an OCRA_RESPONSE_FAILURE error will be returned

Command Examples

Below is the example of a ping command

http://127.0.0.1:8080/pinsafe/AgentXML?xml=<?xml version="1.0"?><SASRequest><Version>3.1</Version><Action>ping</Action></SASRequest>

Expected output

```
   <?xml version="1.0" ?>
 - <SASResponse>
   <Version>3.6</Version>
   <RequestID />
   <Result>PASS</Result>
   </SASResponse>
```

Below is the example of a session request command http://127.0.0.1:8080/pinsafe/AgentXML?xml=<?xml version="1.0" ?><SASRequest><Version>3.6</Version><Secret>secret</Secret><Action>sessionstart</Action><Username>graham</Username></SASRequest>

Expected output

```
  <?xml version="1.0" ?>
- <SASResponse>
  <Version>3.6</Version>
  <RequestID />
  <Result>PASS</Result>
  <SessionID>f853792503f2e83f3ff55f693f631537</SessionID>
  </SASResponse>
```

Swivel log

```
local:Session started for user: graham.
```

Below is the example of a login command

http://127.0.0.1:8080/pinsafe/AgentXML?xml=<?xml version="1.0" ?><SASRequest><Version>3.6</Version><Secret>secret</Secret><Action>login</Action><Username>graham</Username><Password></Password><OTC>84

Expected output

```
  <?xml version="1.0" ?>
- <SASResponse>
  <Version>3.6</Version>
  <RequestID />
  <Result>PASS</Result>
  <Channel>SINGLE</Channel>
  </SASResponse>
```

Swivel log

```
local:Login successful for user: graham.
```

Below is the example of a changepin command

http://127.0.0.1:8080/pinsafe/AgentXML?xml=<?xml version="1.0"?><SASRequest><Version>3.6</Version><Secret>secret</Secret><Action>changepin</Action><Username>graham</Username><Password></P

Expected output

```
  <?xml version="1.0" ?>
- <SASResponse>
  <Version>3.6</Version>
  <RequestID />
  <Result>PASS</Result>
  <Channel>SINGLE</Channel>
  </SASResponse>
```

Swivel log

```
local:Change PIN successful for user: graham.
```

Below is the example of a resetcode (reset PIN) request command

http://127.0.0.1:8080/pinsafe/AgentXML?xml=<?xml version="1.0"?><SASRequest><Version>3.6</Version><Secret>secret</Secret><Action>resetcode</Action><Username>graham</Username></SASRequest>

Expected output

```
  <?xml version="1.0" ?>
- <SASResponse>
  <Version>3.6</Version>
  <RequestID />
  <Result>PASS</Result>
  </SASResponse>
```

Swivel log

```
local:Self-reset code request successful for user: graham.
```

Below is the example of a reset (reset PIN) command

http://127.0.0.1:8080/pinsafe/AgentXML?xml=<?xml version="1.0"?><SASRequest><Version>3.6</Version><Secret>secret</Secret><Action>reset</Action><Username>graham</Username><Resetcode>697501

Expected output

```
  <?xml version="1.0" ?>
- <SASResponse>
  <Version>3.6</Version>
  <RequestID />
  <Result>PASS</Result>
  </SASResponse>
```

Swivel log

```
local:Self-reset successful for user: graham.
```

Error and Warning Messages

Error	Meaning
AGENT_ERROR_AGENT_ACCESS	The user is not in the correct group to use this agent
AGENT_ERROR_ACTION_TYPE	The XML Request sent by the Agent did not contain an unrecognised Action element.
AGENT_ERROR_GENERAL	An unspecified error occurred.
AGENT_ERROR_NO_ACTION	The XML Request sent by the Agent did not contain an Action element.
AGENT_ERROR_NO_AUTH	Swivel does not know how to authenticate this user
AGENT_CANNOT_CHANGE_REPOSITORY_PASSWORD	A pin change failed as the two different passwords were submitted but the agent was configured to use repository passwords
AGENT_ERROR_NO_CHANGE	A pin change failed as the credentials submitted were the same
AGENT_ERROR_NO_PIN	The user has no PIN set
AGENT_ERROR_AUTH_METHOD_UNSUPPORTED	This agent cannot authenticate a user using this method, eg attempting a single channel authentication on a dual-channel only agent
AGENT_ERROR_NO_OTC	One-time code was missing or malformed
AGENT_ERROR_BAD_OTC	
AGENT_ERROR_SESSION	A Session could not be created by the Swivel server. Please try again at a later time.
AGENT_ERROR_UNAUTHORIZED	The Agent is not authorised to use the Swivel server.
AGENT_ERROR_USERNAME	The Username element in the XML Request sent by the Agent contained invalid characters.
AGENT_ERROR_XML	The XML Request sent by the Agent to the Swivel server was malformed.
AGENT_WARN_CHANGE_PIN	The user is required to change PIN before their next login
AGENT_WARN_PIN_EXPIRY	The users PIN will shortly expire, ie within the period sepcified by the change pin warning on the Swivel server.