

# Table of Contents

|  |           |
|--|-----------|
| <b>1 Google Apps Integration.....</b>                        | <b>1</b>  |
| <b>2 Using Swivel for Google Apps Authentication.....</b>    | <b>2</b>  |
| <b>3 Prerequisites.....</b>                                  | <b>3</b>  |
| <b>4 Google SSQ.....</b>                                     | <b>4</b>  |
| <b>5 Swivel and Google Apps.....</b>                         | <b>5</b>  |
| <b>6 User Experience.....</b>                                | <b>6</b>  |
| <b>7 Install the Swivel Google software.....</b>             | <b>10</b> |
| <b>8 Create private keys and certificates.....</b>           | <b>11</b> |
| 8.1 Creating DSA Private Key.....                            | 11        |
| 8.2 Creating a Certificate.....                              | 11        |
| <b>9 Configure the Google Swivel install.....</b>            | <b>12</b> |
| <b>10 Writing the configuration data.....</b>                | <b>13</b> |
| 10.1 Configuring Swivel for Agent XML Authentication.....    | 13        |
| 10.2 Configuring Swivel for Single Channel Images.....       | 13        |
| 10.3 Configuring Swivel for Dual Channel Authentication..... | 13        |
| <b>11 Configuring Google Apps to use the Swivel IdP.....</b> | <b>14</b> |
| <b>12 Testing.....</b>                                       | <b>15</b> |
| <b>13 Troubleshooting.....</b>                               | <b>16</b> |
| 13.1 Error Messages.....                                     | 16        |
| <b>14 Google Docs Integration.....</b>                       | <b>17</b> |

# 1 Google Apps Integration

## 2 Using Swivel for Google Apps Authentication

GoogleApps is a Software-as-a-Service approach to email, calendars and online document sharing. Swivel can provide Two Factor authentication with SMS, Token, Mobile Phone Client and strong Single Channel Authentication TURing, Pinpad or in the Taskbar using RADIUS.

Organisations can configure their GoogleApps domain to use single-sign-on (SSO), all users in the domain are required to use the Swivel authentication, although with the Authentication Manager it is possible for Swivel to log users in to other applications. This means that rather than supply GoogleApps with a username/password, you configure GoogleApps to refer to an authentication portal to authenticate the user. The portal collects and checks the users credentials and passes back the result of the authentication to GoogleApps.

This document describes how Swivel can be configured to act as the authentication portal for GoogleApps.

## 3 Prerequisites

Swivel authentication platform 3.x

Google account

The authentication page must be placed in a location that can be accessed through the internet, usually by using a NAT to a Swivel virtual or hardware appliance.

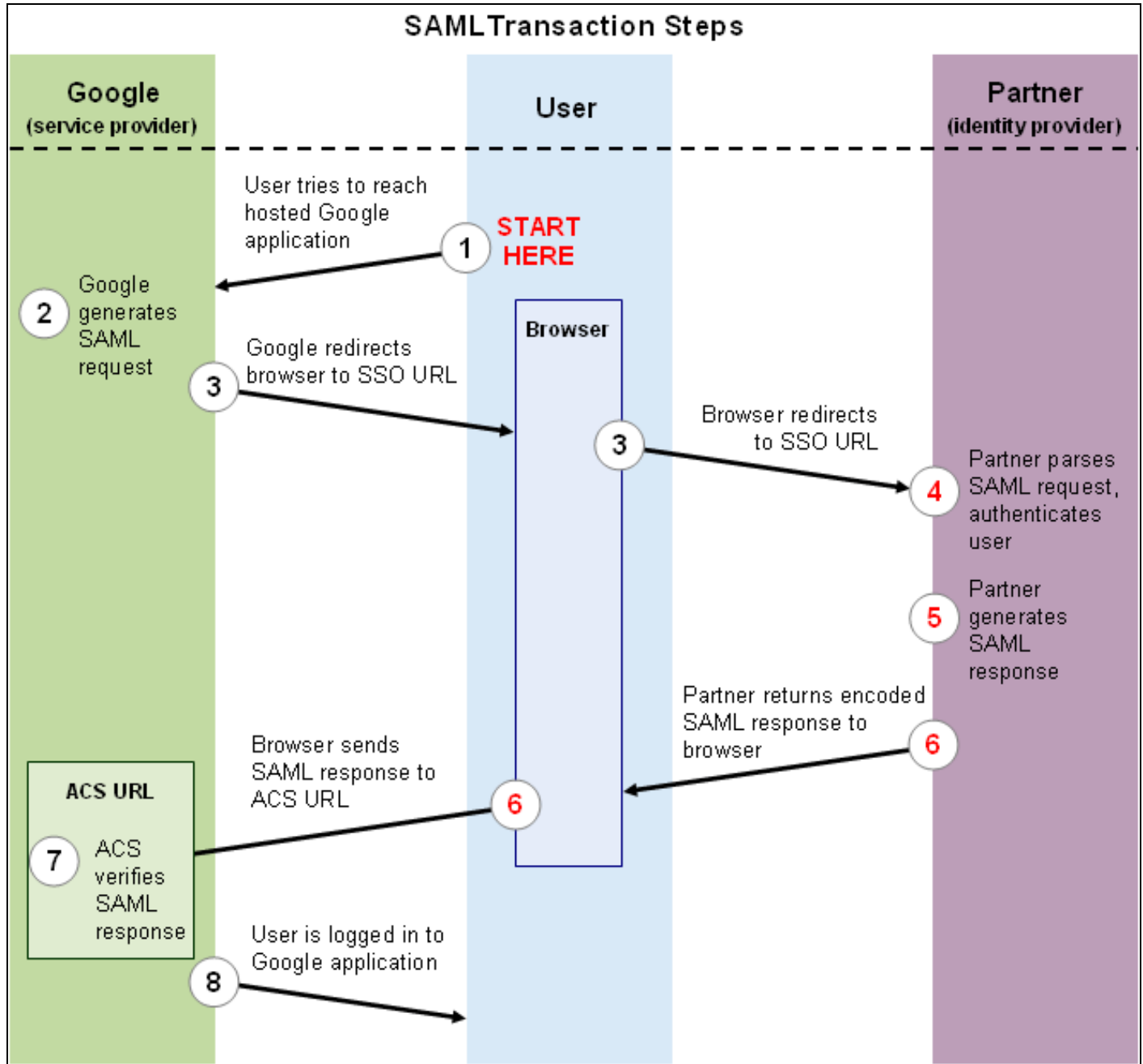
[Swivel Google Authentication Portal](#).

## 4 Google SSO

The diagram below is taken from [Google Apps reference site](#)

When a user attempts to access a Google Apps application Google Apps will look for the presence of a cookie that indicates that the user is an authenticated user. If that cookie is not present the user is redirected to the Partner (Identity Provider) Site.

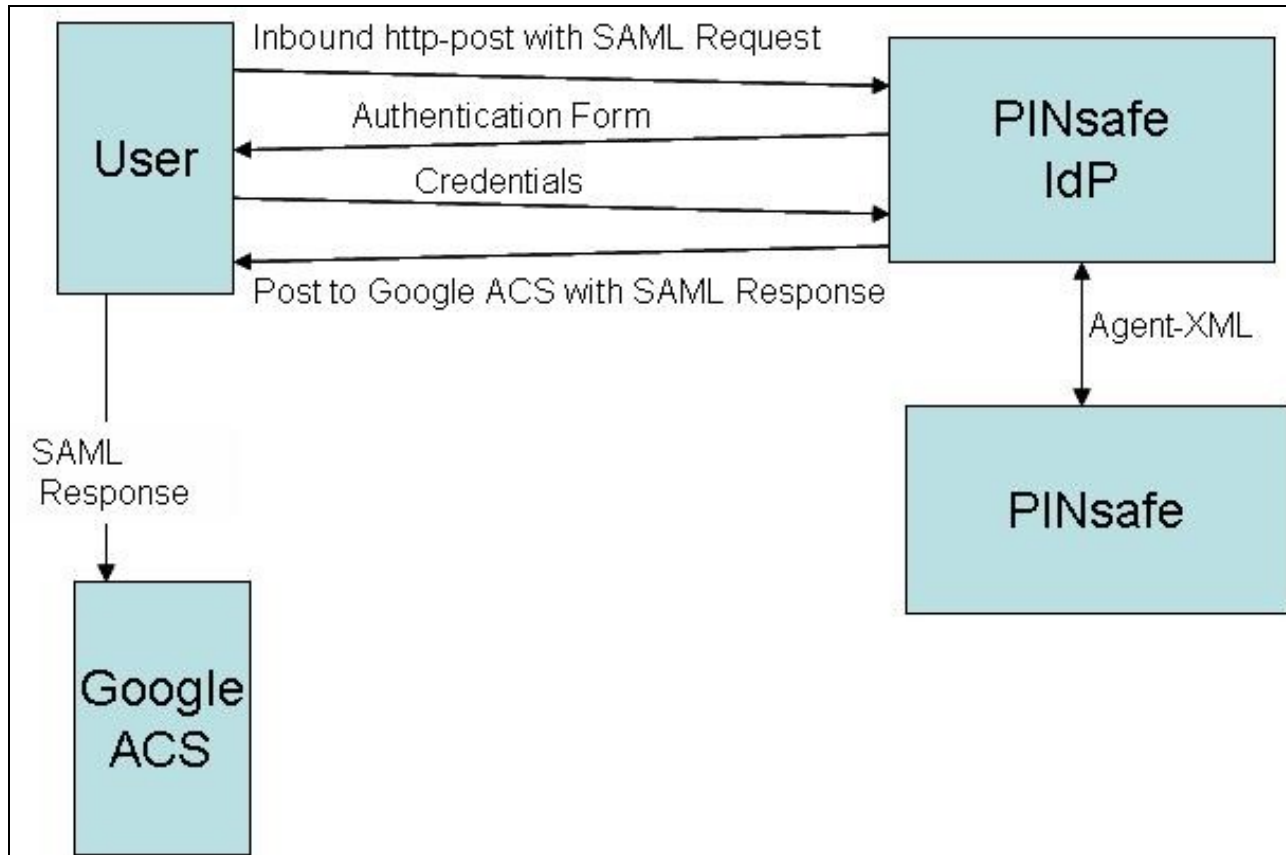
That redirect will include a SAML request. The request includes the url of the Google Apps ACS (Assertion Consumer Service). This is the Google Apps Service that controls access to Google Apps



The Identity Provider (IdP) authenticates the user. If the authentication is successful it creates a SAML response and posts that response to the url of the Google Apps ACS that it was passed in the SAML request. The ACS then allows the user access as appropriate.

## 5 Swivel and Google Apps

Swivel has its own XML-based API that it uses for authentication. There is now an external Swivel application that can interpret the inbound SAML request, carry out a standard Swivel authentication via Agent-XML and then post the associated SAML response.



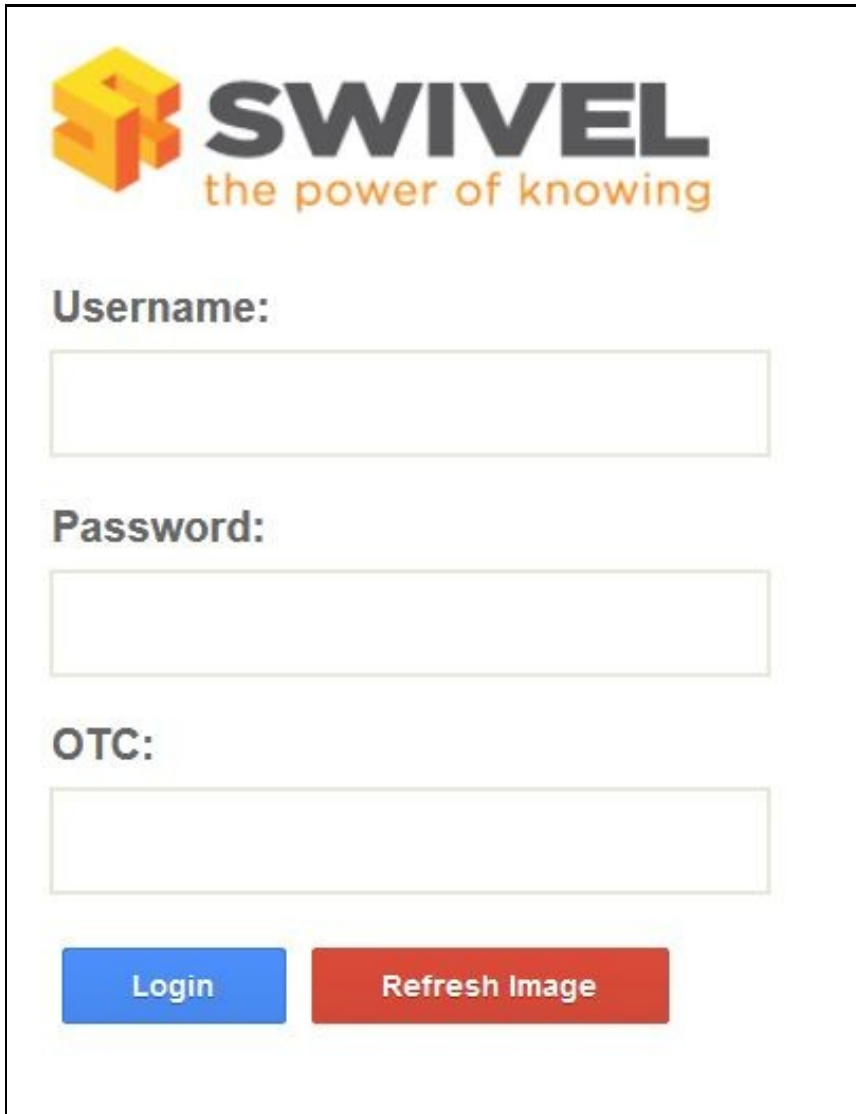
This application needs to be publicly accessible so that users can authenticate to it, it also needs to be configured as an agent on a Swivel server. More detailed configuration information appears later in this document.

## 6 User Experience

The user opens a browser and accesses googleApps e.g. <http://mail.google.com/a/swivelsecure.net> this is then redirected in a new URL includes the encrypted SAML request.

What the user sees is a login page familiar to Swivel users. This page can be modified depending on the form of Swivel authentication required. The user authenticates to this form in the same way as any other Swivel authentication form.

Swivel login page



The image shows a screenshot of the Swivel login page. At the top left is the Swivel logo, which consists of a stylized 'S' made of yellow and orange blocks, followed by the word 'SWIVEL' in large, bold, black letters, and the tagline 'the power of knowing' in smaller, orange letters below it. Below the logo are three input fields: 'Username:', 'Password:', and 'OTC:'. Each field is a simple white rectangle with a thin grey border. At the bottom of the form are two buttons: a blue button labeled 'Login' and a red button labeled 'Refresh Image'.

Dual Channel Authentication



**Username:**

user@domain.com

**Password:**

•••••

**OTC:**

••••|

Login

Refresh Image

Single Channel Authentication





**Username:**

**Password:**

**OTC:**

Login

Refresh Image



After the user has submitted the correct credentials, the browser is redirected to the GoogleApps ACS page and then again to the user's landing page. The user is now authenticated and can access any of their GoogleApps.



Search Mail

Search the web

[Show search options](#)  
[Create a filter](#)

[Compose Mail](#)

**Inbox**

[Starred](#) ★

[Sent Mail](#)

[Drafts](#)

[Follow up](#)

[Misc](#)

[Priority](#)

[4 more](#) ▼

[Contacts](#)

[Tasks](#)

**- Chat**

Search, add or invite

● Test User

Set status here ▼

[Options](#) ▼

[Add contact](#)

**Archive** Report spam Delete Move to ▼ Labels ▼ More actions ▼

[Refresh](#)

Select: All, None, Read, Unread, Starred, Unstarred

- ☆ Gmail Team Get started with Gmail - Gmail is built on the ide
- ☆ Gmail Team Access Gmail on your mobile phone - The days

Select: All, None, Read, Unread, Starred, Unstarred

**Archive** Report spam Delete Move to ▼ Labels ▼ More actions ▼

[Refresh](#)

See whether the

## 7 Install the Swivel Google software

This is usually deployed on the Swivel server, but may be deployed within a Java container such as Apache Tomcat on another server. In HA deployments with multiple Swivel instances, the Software can be deployed in each instance.

Swivel virtual or hardware appliances: Use [WinSCP](#) to copy the AuthenticationPortal-google.war file to /usr/local/tomcat/webapps2

Software installs and older virtual or hardware appliances: copy the AuthenticationPortal-google.war file to the webapps folder of the Apache Tomcat installation.

The google software should create a AuthenticationPortal-google folder.

## 8 Create private keys and certificates

Communication between Google and the Swivel instance is secure through the use of certificates.

### 8.1 Creating DSA Private Key

DSA key generation involves two steps, and can be done through the command line on a Swivel virtual or hardware appliance:

1.

```
openssl dsaparam -out dsaparam.pem 2048
```

2.

```
openssl gendsa -out dsaprivkey.pem dsaparam.pem
```

The first step creates a DSA parameter file, `dsaparam.pem`, which in this case instructs OpenSSL to create a 2048-bit key in Step 2. The `dsaparam.pem` file is not itself a key, and can be discarded after the public and private keys are created. The second step actually creates the private key in the file `dsaprivkey.pem` which should be kept secret.

Export the key into a DER (binary) format. You can do so with the following steps:

1.

```
openssl dsa -in dsaprivkey.pem -outform DER -pubout -out dsapubkey.der
```

2.

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in dsaprivkey.pem -out dsaprivkey.der -nocrypt
```

Step 1 extracts the public key into a DER format. Step 2 converts the private key into the pkcs8 and DER format. Once you've done this, you can use this public (`dsapubkey.der`) and private (`dsaprivkey.der`) key pair.

### 8.2 Creating a Certificate

Once you have your key pair, it's easy to create an X.509 certificate. The certificate holds the corresponding public key, along with some metadata relating to the organization that created the certificate. Follow this step to create a self-signed certificate from either an RSA or DSA private key:

```
openssl req -new -x509 -key dsaprivkey.pem -out dsacert.pem
```

After you answer a number of questions, the certificate will be created and saved as `dsacert.pem`.

The created keys, `dsapubkey.der` and `dsaprivkey.der` need to be copied to the keys folder or wherever specified within `settings.xml`

The **dsacert.pem** certificate needs to be uploaded to the GoogleApps server.

## 9 Configure the Google Swivel install

Edit the AuthenticationPortal-google\WEB-INF\settings.xml file.

**pinsafessl** default: false - To use SSL communications on the pinsafeport set this to TRUE, to use without SSL set this to False.

**pinsafeserver** default: adouglas.swivelsecure.net - The hostname or IP address of the Swivel server.

**pinsafecontext** default: pinsafe - The installation name of the Swivel application.

**pinsafesecret** default: secret - The shared secret configured on the Swivel server.

**pinsafeport** default: 8080 - The communication port for the Swivel server.

**imagesssl** default: false - To use SSL communications on the imageserver port set this to TRUE, to use without SSL set this to False.

**imageserver** default: adouglas.swivelsecure.net - The hostname or IP address used for retrieving images from the Swivel server. This must be contactable from the internet.

**imagecontext** default: pinsafe - The Swivel installation name used for retrieving images from the Swivel server. For virtual or hardware appliances this is usually *proxy*. For Software installations this is usually *pinsafe*.

**imageport** default: 8080 - The port used for retrieving images from the Swivel server. For virtual or hardware appliances this is usually 8443. For a software only install see [Software Only Installation](#).

**selfsigned** default: true - To use SSL communications on the imageserver port with a self signed or invalid certificate set this to TRUE, to use without only the correct SSL certificate set this to False.

**certificateIssuer** default: SwivelSecure

**publicKeyFilePath** default: /keys/pinsafe/robssl/dsapubkey.der

**privateKeyFilePath** default: /keys/pinsafe/robssl/dsaprivkey.der

**certificateFilePath** default: /keys/pinsafe/robssl/dsacert.pem

## 10 Writing the configuration data

From a web browser run the following:

For a virtual or hardware appliance

[https://Swivel\\_google\\_server:8443/AuthenticationPortal-google/configuration.jsp](https://Swivel_google_server:8443/AuthenticationPortal-google/configuration.jsp)

For a software only install see [Software Only Installation](#)

Click on the Generate *Idp Metadata* button.

The *Idp WS-Metadata* button is provided for future enhancements and is not currently used.

This will then generate Metadata files.

Example:

Swivel Virtual Appliance or hardware Appliance:

Metadata successfully written to /usr/local/tomcat/webapps2/AuthenticationPortal-google/generatedIdPMetadata.xml

Software installation:

Metadata successfully written to C:\Program Files (x86)\Apache Software Foundation\Tomcat 6.0\webapps\AuthenticationPortal-google\generatedIdPMetadata.xml

### 10.1 Configuring Swivel for Agent XML Authentication

The IdP is usually deployed on the Swivel hardware or virtual appliance, and a default localhost Agent is usually pre-configured. To make any changes to this see [Agents How to Guide](#)

### 10.2 Configuring Swivel for Single Channel Images

If Swivel Single Channel images are to be used for authentication, then the following guide can be used.

[Single Channel How To Guide](#)

### 10.3 Configuring Swivel for Dual Channel Authentication

If Swivel Dual Channel authentication methods are to be used, refer to the following guide:

[Transport Configuration](#)

## 11 Configuring Google Apps to use the Swivel IdP

To set GoogleApps to use the Swivel IdP you need to configure the service from the Google Apps admin console.

The settings are under: Security, Advanced settings -> Set up single sign-on (SSO).

### Set up single sign-on (SSO)

To set up SSO, please provide the information below. [SSO Reference](#)

Enable Single Sign-on

#### Sign-in page URL \*

URL for signing in to your system and Google Apps

#### Sign-out page URL \*

URL to redirect users to when they sign out

#### Change password URL \*

URL to let users change their password in your system

#### Verification certificate \*

A certificate file has been uploaded-[Replace certificate](#)

The certificate file must contain the public key for Google to verify sign-in requests. [Learn more](#)

You need to enter the public IP address of the Swivel IdP, including the port number and upload the certificate generated in the previous section.

**You will need to include the port numbers of the Idp unless you have configured the virtual or hardware appliance firewall (see [How to run PINsafe on non-default ports](#)) to map port 80 to the port the Idp is listening on**

The **dsacert.pem** certificate needs to be uploaded to the GoogleApps server. If the existing certificate is being replaced and clicking to **Replace certificate** is not working, try in Chrome or Safari web browsers.

## 12 Testing

Browse to the Swivel Google login page to check that it is working:

Swivel virtual or hardware appliance install: [https://swivel\\_appliance:8443/AuthenticationPortal-google/identity\\_provider.jsp](https://swivel_appliance:8443/AuthenticationPortal-google/identity_provider.jsp)

For a software only install see [Software Only Installation](#)

If these work then browse to the google login page, the browser should be directed to a sign-in page. This page is the Swivel IdP. The url is something like:

```
http://<idp IP address>/pinsafeldp.jsp?SAMLRequest=fVLJTSMwEL0j8Q%2BW79kqKiGrCSpFiEosEQ0cuDnOpHXxEjxOA3%2BPm
4KAA72%2BmXnLzMwu3rUiO3AorclpFqeUgBG2kWad06fqOjqnF8XpyQy5Vh2b935jHuGtB%2FQkTBpkYyGnvTPMcpTIDNeAzAu2mt
%2Fdskmcss5Zb4VVICyvcmbjvkV1FKDsOZVSWj0tjU1r7fbRnMtTVivBKwpef62NdnbWiL2sDToufEBSrM0SqdrNqmyKUvP2dn0h
ZLyS%2BISmkOCY7bqQxOym6oqo%2FJhVY0EO9mAuw%2FdOV1bu1YQC6v38iVHILsAt1whUDJHBOeDwYU12GtwK3A7KeDp8TanG%2B
87ZEkyDEP8Q5PwBlfAEeZF7yA24BMukBbjftkY0f1a7PEA%2FNsALX4kZskvquLrbvs4y6vSKik%2ByFwpOywccB%2ByeNeHKNfWae
7%2FV8vibERkE7VjK%2BsNdiBkG65HSVlCVP8%2BShibTw%3D%3D&RelayState=https%3A%2F%2Fwww.google.com%2Fa%2F
swivelsecure.net%2FServiceLogin%3Fservice%3Dmail%26passive%3Dtrue%26rm%3Dfalse%26continue%3Dhttp%253A
%252F%252Fmail.google.com%252Fa%252Fswivelsecure.net%252F%26bsv%3D1eic6yu9oa4y3%26tmpl%3Ddefault%26tmpcache%3D2
```



## 13 Troubleshooting

Check the Swivel logs.

The Tomcat catalina.out file will display error messages relating to creation of the Meta Data.

Virtual or hardware appliance : /var/logs/tomcat/catalina.out

### 13.1 Error Messages

**This account cannot be accessed because the login credentials could not be verified.**

**We are unable to process your request at this time, please try again later.**

The certificates, address or ports may be incorrect.

**Login Failed: Invalid user.**

Verify the username used is present on the Swivel instance. Check the Swivel logs for failed authentications.

## 14 Google Docs Integration

See [Google Apps Integration](#)