Vitrium OAuth Guide for External Service Authentication, Authorization & SSO

For Vitrium Security v8.3.0+ and External Service v1.0

Updated: August 24, 2021
Document Revision: 1.00

Table of Contents

Document Revisions **3**

OAuth 4

Web Viewer by OAuth 5

Web Viewer by OAuth Workflow 5

Vitrium External Settings Configuration 7

SSO Type 8

SSO Required 8

Authentication URL 8

Token URL 9

User Info URL 10

Hide logout button in WV 10

After logout URL 10

Skip Failed Page. 10

Failed Page Button Text 11

Disable Session Interval Validation 11

Session Validation Interval 11

Disable Document Ticket Interval Validation 11

Document Ticket Validation Interval 11

Force Authentication by Token 12

Remember Me 12

Consumer Key 12

Consumer Secret 12

Scope 12

Response Type 12

OAuth Result Type 12

Identity Field Name 12

Access Token in Header 13

Prompt Login 13

Authentication from Query Parameter 13

Trust Invalid Certificates 13

Sample Configuration - Salesforce 14

Document Revisions

Revision Number	Reason	Date
1.00	Created	August 24, 2021.

OAuth

OAuth 2.0 is an open standard for authorization which rose to prominence through its adoption by companies such as Google, Facebook, Microsoft, and Twitter. OAuth 2.0 replaces and is not backwards compatible with the older OAuth 1.0.

Vitrium Security supports OAuth 2.0 for Web Viewer (browser-based) access only – OAuth 2.0 is available for use with downloaded protected PDF content that would be opened in Adobe.

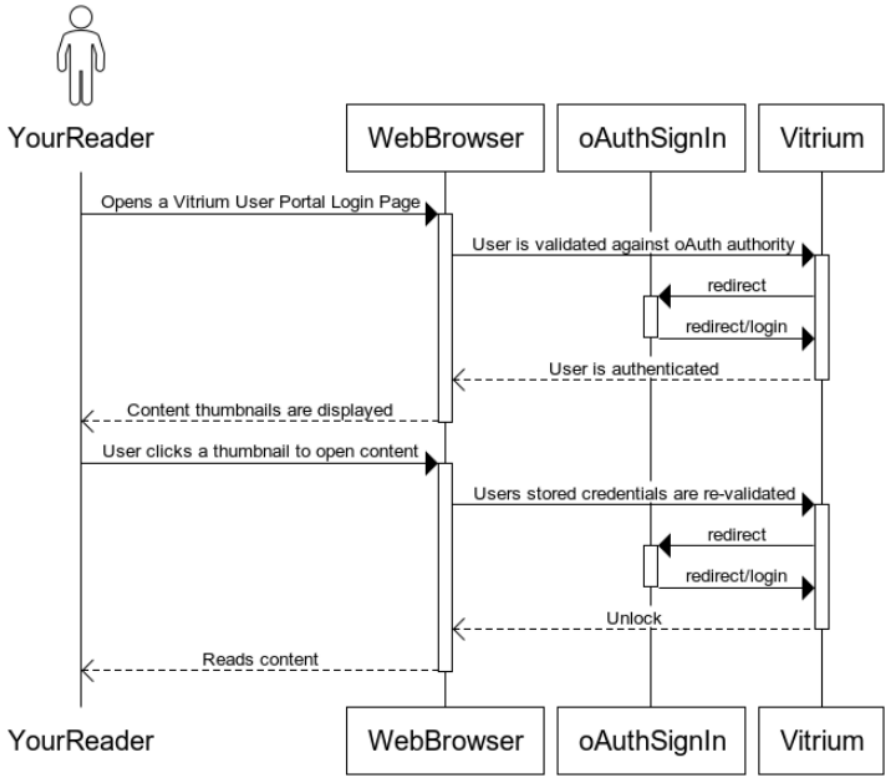
It is important to note that OAuth 2.0 itself is normally a protocol for authorization, not authentication. However, in the context of Vitrium Security, OAuth 2.0 is used in combination with OpenID Connect to perform Reader authentication and enable Single Sign-On (SSO) functionality. OpenID Connect is an extension of the OAuth 2.0 standard that adds pseudo-authentication to OAuth 2.0 by defining a way to retrieve information about an end user. Authorized the authenticated Reader's access to protected content remains the responsibility of Vitrium.

An in-depth discussion of OAuth 2.0 and OpenID Connect is beyond the scope of this document, but it is easy to find the specifications, implementation samples, discussion forums and other information about these widely used protocols.

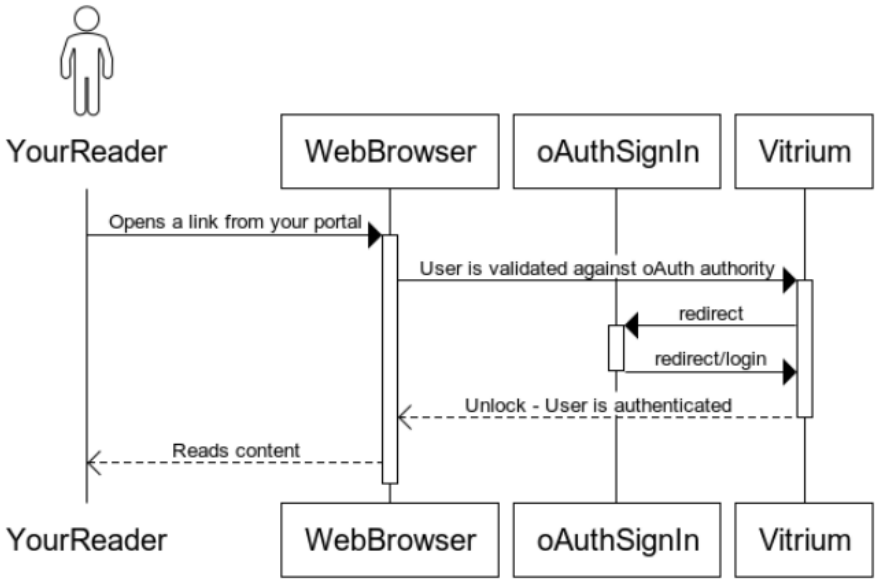
Web Viewer by OAuth

Web Viewer by OAuth Workflow

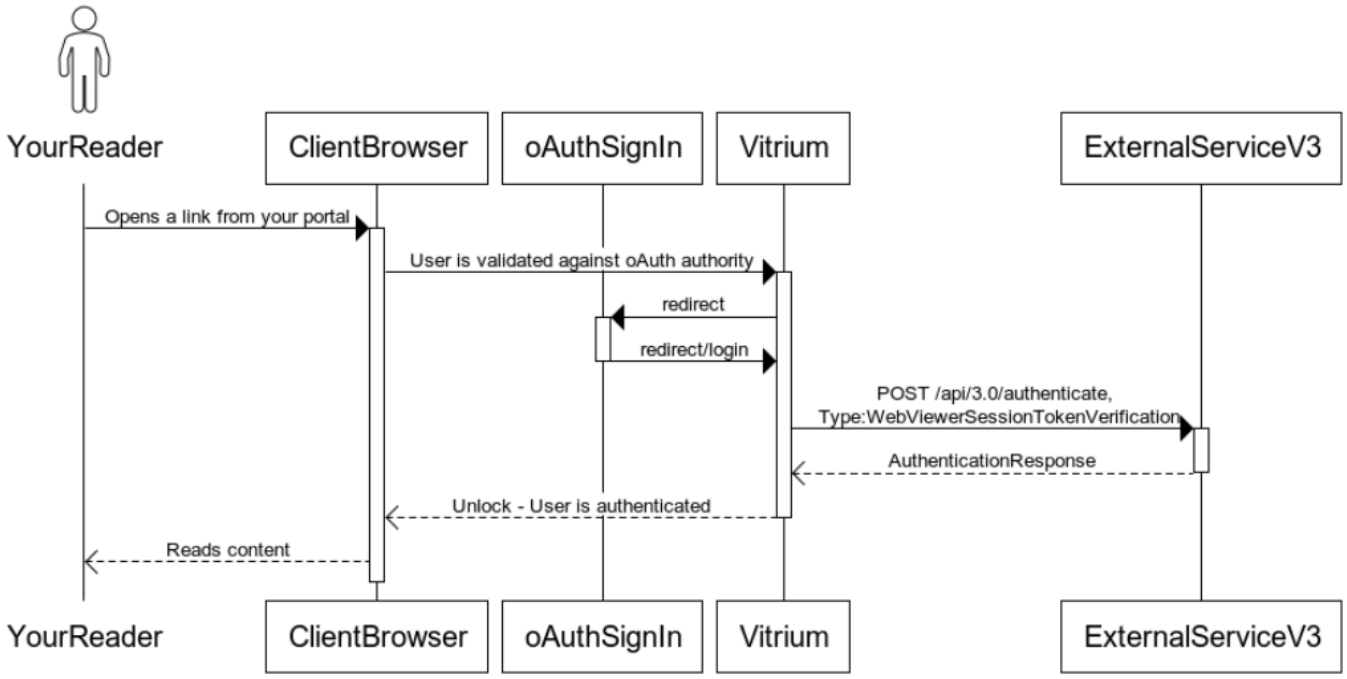
Reader Unlocks a Web Link in User Portal with OAuth2



Reader Unlocks a Web Link with OAuth2

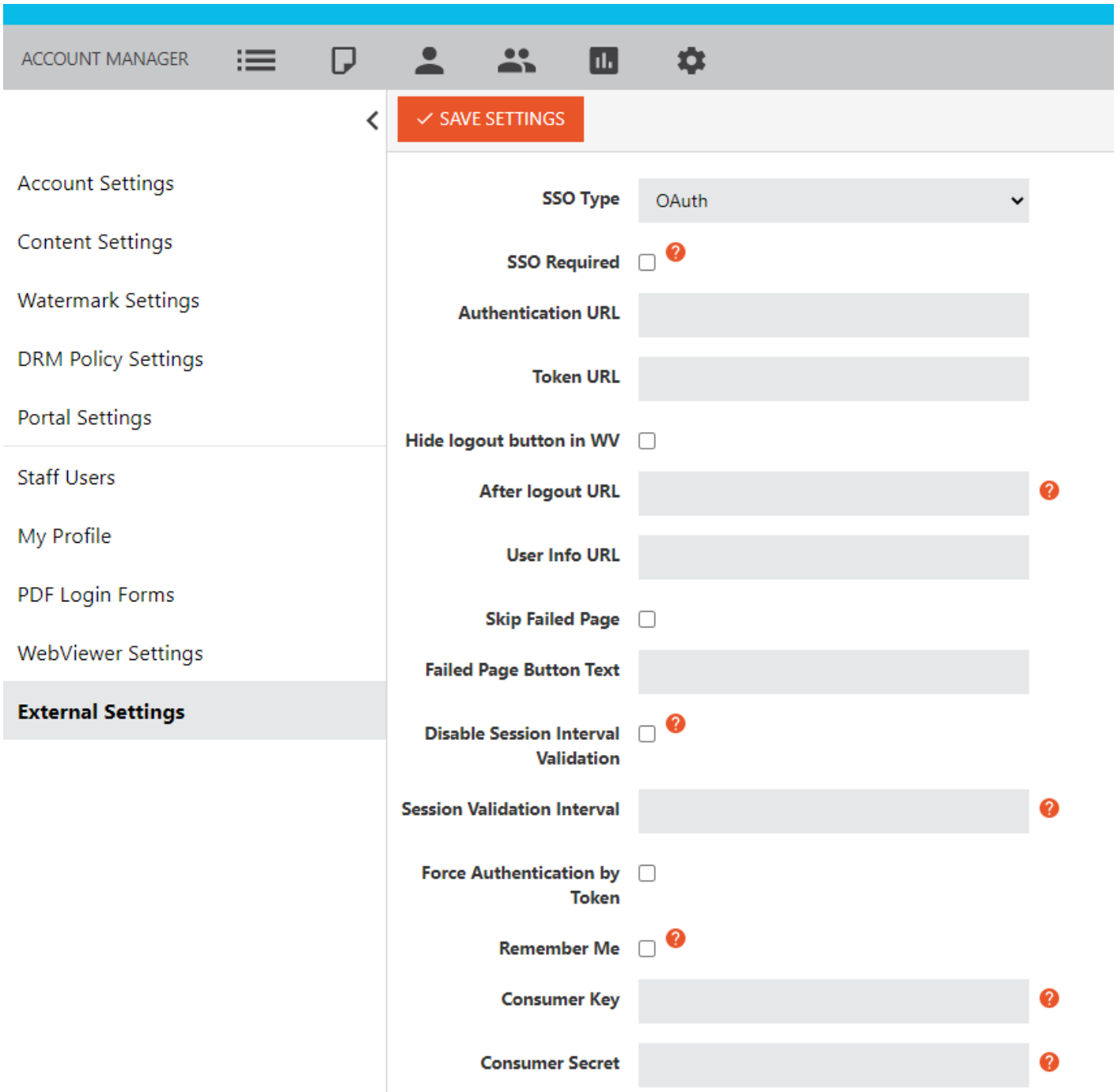


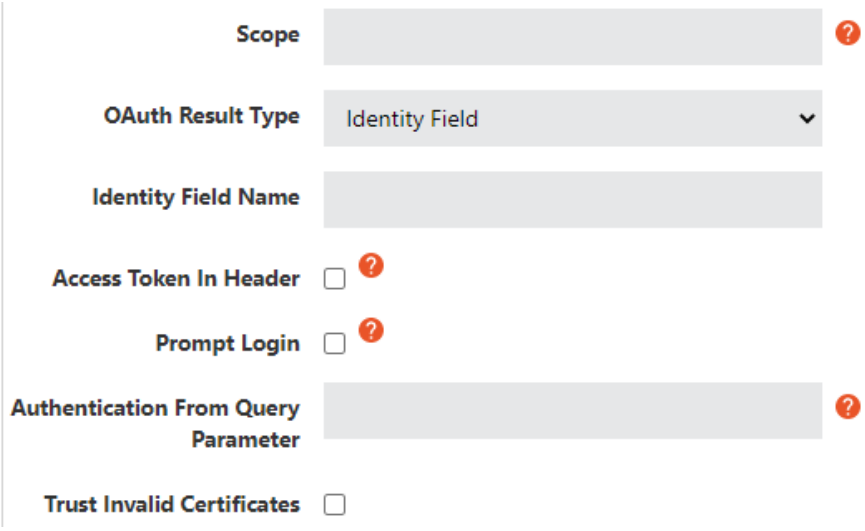
Reader Unlocks a Web Link with OAuth2 (With External Service V3)



Vitrium External Settings Configuration

To take advantage of Web Viewer SSO by OAuth functionality you first need to configure some settings in the Vitrium Admin UI:





SSO Type

Sets the type of SSO used by Vitrium. Options include "None" (no SSO), "Default" (custom SSO using External Service), and "OAuth" (SSO using an external OAuth identity provider). Unless otherwise explicitly specified, this section discusses **External Service SSO implementation** – select "**Default**" in this pull-down.

SSO Required

Checking this box forces Vitrium to use OAuth 2.0 to authenticate a Reader. If this box is left unchecked and OAuth 2.0 authentication is impossible or fails, the Reader is redirected to the Web Viewer login form and has the option to enter a username and password to proceed with regular Vitrium authentication. With the box checked, failed OAuth 2.0 authentication results in an exception and the unauthenticated Reader has no recourse to access the protected content.

If **SSO Required** is checked, SSO failure results in an error page without any recourse for the end user; External Service SSO is the only way that access to protected content can be granted.

If this box is unchecked and Authentication URL is empty, user will be redirected to the default Vitrium login page on SSO failure. Here they will be able to enter their username and password, and these credentials will then be passed to your External Service as part of a manual unlock authorization request.

If enabled, and SSO Authentication fails, Vitrium redirects the end user back to your Authentication URL. It is strongly recommended to provide **Authentication URL** in this case. If this setting is disabled and SSO Authentication fails, we instead redirect the user to the Vitrium Web Viewer login form to allow them to manually login into the content using their Vitrium credentials. NOTE: If not enabling this option, review with a Vitrium Professional Services.

Authentication URL

The URL of SSO service for redirection of unauthenticated user or failed SSO verification.

The key setting here is **Authentication URL**. This is the redirect destination in case of SSO failure. You would normally populate this field with your content management system login page URL, but if you prefer you could instead use a custom error page suggesting an appropriate course of action or any other destination. As such, this URL is a bit of a

misnomer; note that in "OAuth" SSO type, this is normally the URL of the `/authorize` OAuth identity provider endpoint, and that is the origin of the name.

If Authentication URL is populated, it is always used as the redirection target in case of SSO failure, but the field is not required. If you leave it empty, behaviour will be determined by the other related settings

This is the authorization endpoint of your OAuth 2.0 provider. A request to access protected content redirects to this URL to obtain an *authorization grant*. Vitrium includes the following information in the redirection request query string:

```
?client_id={Consumer Key}&redirect_uri={(current Web Viewer URL)/OAuthSignIn?state=(current Web Viewer content code)}&response_type=code&scope={Scope}&state={ current Web Viewer content code}
```

For example, with the configuration in the example screenshot above, a request to https://view.vitrium.com/AaBbCc, where AaBbCc is the Web Viewer content code, would result in redirection to the following URL:

```
https://example.com/oauth2/authorize?client_id=clientid&redirect_uri=https%3A%2F%2Fview.vitrium.com%2FOAuthSignIn%3Fstate%3DAaBbCc&response_type=code&scope=email&state=AaBbCc
```

This URL is responsible for authenticating your end user, as outlined in the OAuth 2.0 specification. The authentication may involve requesting username/password, or perhaps seamlessly using an existing session or cookie; the details are irrelevant. After the end user is authenticated, this endpoint should redirect back to the URL supplied to it in `redirect_uri` and provide two query string parameters: code, which should contain an authorization grant for further communication (this will be supplied by Vitrium in a subsequent call to the Token URL); and state, which should contain the same value as the state parameter passed in. In this example, the redirect may be to a URL such as:

https://view.vitrium.com/OAuthSignIn?code=Mj5fERFjgLJyldygll77Lgyllg8lg3&state=AaBbCc.

Token URL

This is the endpoint which issues an OAuth 2.0 access token. Once an end user is successfully authenticated and the Authentication URL redirects to the OAuthSignIn Vitrium endpoint as described above, Vitrium then issues a POST request to this endpoint with the following query string parameters:

```
grant_type=authorization_code&code={code from Authentication URL redirect}&redirect_uri=={(current Web Viewer URL)/OAuthSignIn?state=(current Web Viewer content code)}&client_id={Consumer Key}&client_secret={Consumer Secret}
```

In our running example, the POST might be to a URL such as:

```
https://example.com/oauth2/token?grant_type=authorization_code&code=Mj5fERFjgLJyldygll77Lgyllg8lg3&redirect_uri=https%3A%2F%2Fview.vitrium.com%2FOAuthSignIn%3Fstate%3DAaBbCc&client_id=clientid&client_secret=clientsecret
```

The response to this POST request should include an `access_token` in its body. Vitrium will first try to parse the body of the response into a JSON object and look for `access_token` there; failing that, it will parse the response as a query-parameter-format string and attempt to extract `access_token` from it.

User Info URL

This is the OpenID Connect UserInfo endpoint, used by Vitrium to obtain data which can be used to retrieve a Reader.

This value is logged in Vitrium as the UserName (email address) for activity logs and analytical reports. It's used to retrieve the "reader" to authorize the user to the document. The authorization is done in Vitrium using the DRM/permission settings.

Vitrium will issue a GET request to this URL provide the access token returned by the Token URL endpoint described above. The token will be supplied in the request header or in the query string according to the Access Token In Header setting.

Hide logout button in WV

Checking this checkbox hides the Logout button in Vitrium's Web Viewer. If you have a business case to disallow users from having access to the log out button in the Web Viewer.

This option is useful if you do not want users to be able to log out of Vitrium.

During Logout, the web viewer clears Vitrium's own session cookie and your SSO cookie (Referenced in the "Cookie Name" textbox), if it is used. If the SSO Cookie is also used by your web portal for other purposes, then clearing the SSO cookie could also log your user out of your portal. The Hide logout checkbox in WV should be used if this behavior is undesirable.

After logout URL

Optional URL to which the user should be redirected after clicking the Logout button in Web Viewer. This option is only available if the logout button is not being hidden by the "Hide Logout Button in WV" option.

This option should be used if your External Service does not validate username/password, or if you wish for them to use your own portal's login page.

If not set, the default behaviour is to redirect the user to the Web Viewer login form where the user could login manually.

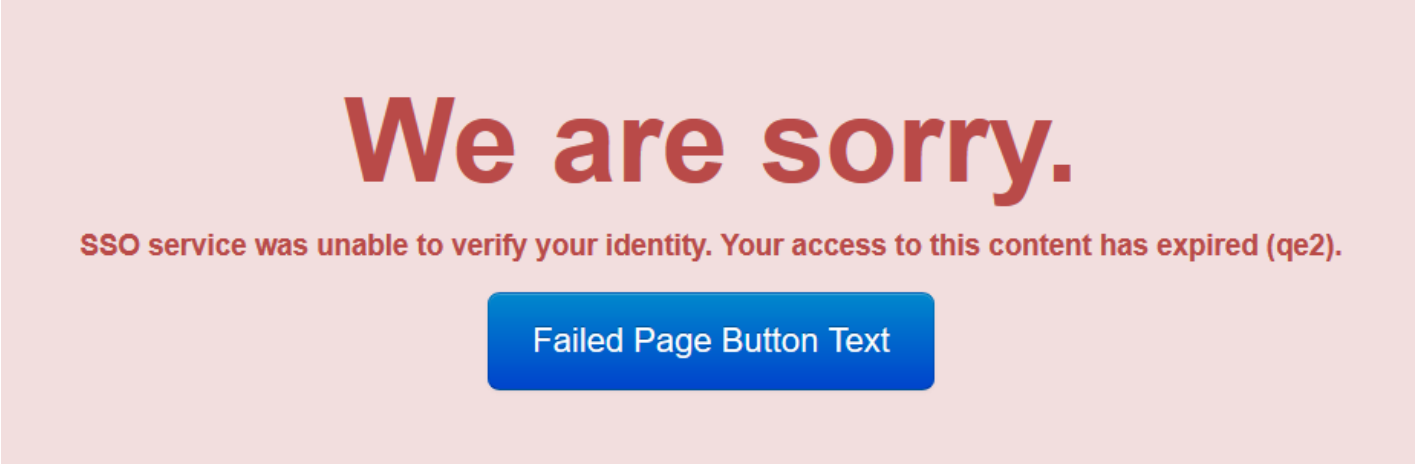
Skip Failed Page.

If set to true (1) and the SSO Authentication fails, the end user is presented with a failed page explaining that the SSO Authentication failed. If the **Skip Failed Page** box is checked, Vitrium will skip displaying this page and instead redirect the user immediately to Authentication URL or the manual login page. If Skip Failed Page is unchecked and the page is being displayed, you have the ability to customize the text of the redirect button using the **Failed Page Button Text** field. The end user is instead redirected immediately either to your Authentication URL or Vitrium Web Viewer login screen.

Failed Page Button Text

This setting is only relevant if **Skip Failed Page** is set to false (0) and can be used to customize the text of the button redirecting the user.

The final piece here is the SSO Fail Page. Unless an error page is being displayed under the conditions mentioned above, Vitrium will display a page informing the user about SSO failure. For example, if your External Service returns a policy with an Expiry date in the past, the user might be presented with a page like this:



Disable Session Interval Validation

As long as Web Viewer content is on screen, Vitrium periodically verifies that the logged in user still has access to the content. This ensures that your content is protected if you deactivate the user's access, and the user is currently viewing the content. When access is deactivated, the validation process fails which locks the user out of the web viewer, erases content from offline storage, and notifies the user that he/she does not have access to the content.

This setting enables or disables this validation check. It disables session token expiration. It ensures that SSO backend service is contacted only once per session lifetime.

Session Validation Interval

This setting configures the interval in which the validation check above is performed. It's the interval before the token expires. The default is 90 minutes. When a session token expires Vitrium talks to backend to validate session again.

Disable Document Ticket Interval Validation

It disables re-verifying user access to content.

Document Ticket Validation Interval

It is the interval (in minutes) for re-verifying user access to content. The default is 5 minutes. Increasing this interval will ease load on your external service but impact its responsiveness and delay revoking content access.

Force Authentication by Token

When SSO is enabled, and if SSO failed, the user could still fall-back to logging in using their username/password. The setting turns off the ability for a user to do so.

This setting should be turned on for integrations that does not expose a way to validate by username/password.

Remember Me

If enabled, it allows offline access and potentially reduce the number of log in. if enabled a web viewer session is set in a cookie with expiry of 1 year from now FOR THE CURRENT CONTENT. If not enabled, then they will need to authenticate each time.

Consumer Key

Consumer Key (AKA Client ID). Must be obtained from the OAuth2 provider. The value is passed to the **Authentication URL and Token URL**. It is a public identifier for application.

Consumer Secret

Consumer Secret (AKA Client Secret). Must be obtained from the OAuth2 provider. The value is passed to the Token URL. It is a secret known only to the application and the authorization server. It must be sufficiently random to not be guessable.

Scope

OAuth2 Access Token Scope, that is **openid.** It provides a way to limit the amount of access that is granted to an access token. It is a list of permission.

Response Type

This indicates what values are returned by the OAuth authorize/ endpoint. The allowed values are "code", "token" (access token), and "id token", but the values can be combined. Vitrium currently only uses the "code", which is our default value. So technically we can allow "code", "code token", "code id_token", or "code id_token token", but in all cases we only use code and ignore any tokens.

OAuth Result Type

Identity Field: If set to "Identity Field", it will look for a value identified by “Identity Field Name” in the UserInfoUrl endpoint response, and try to match the value to an existing username. For example, setting OAuth Result Type to “Identity Field” and Identity Field Name to “email” will inspect the UserInfoUrl response and look for a field named “email”, then take the value of the “email” field and look for a username matching this value. The most common Identity Field Name values in this scenario are “email” and “preferred_username”.

Identity Field Name

Enabled if OAuth Result Type = "Identify Field". We look for this data element in the User Info URL response. This becomes the username that is then logged in Vitrium. That is "email" or "preferred username"

This URL might be useful information for Salesforce: http://releasenotes.docs.salesforce.com/en-us/spring14/release-notes/rn_forcecom_security_userinfo_endpoint.htm

The official openid spec is here: https://openid.net/specs/openid-connect-core-1_0.html#UserInfo (search for section 5.3.2)

Access Token in Header

Determines how the access token is provided to the User Info URL. If checked, the token is supplied in a request header in the format “Authorization: Bearer {access token}”. If unchecked, the token is supplied in the query string as a parameter with the name “access token”. Used by some OAuth2 environments. When enabled, OAuth2 authorization access token will be passed in the header. If disabled, the URL parameter will be used.

Prompt Login

If enabled, it forces the user to sign-in again before it will show the authorization prompt. We append "&auth_type=reauthenticate&prompt=login" to the AuthenticationUrl.

Authentication from Query Parameter

If **Authentication from Query Parameter** is populated, then a query string parameter with that name is appended to Authentication URL, and its value is set to the current Web Viewer content URL with any query parameters stripped off. When specified, it adds to AuthenticationUrl parameter from which URL it has been redirected. For example, Authentication URL https://myservice.com/login, Authentication from Query Parameter “redirect_from”, and failed SSO attempt to open content at https://view.vitrium.com/content?token=ssosecret would result in a redirection target of https://myservice.com/login?redirect_from=https%3A%2F%2Fview.vitrium.com%2Fcontent%3Ftoken%3Dssosecret

Trust Invalid Certificates

Checking this checkbox instructs Vitrium to ignore invalid SSL certificate during OAuth sign-in. This should only be used on development systems.

Sample Configuration - Salesforce

Setting	OAuth2 Example Salesforce
SSO Type	OAuth
SSO Required	1 (Enabled)
Authentication URL	https://my.salesforce.com/vitrium/services/oauth2/authorize
Token URL	https://my.salesforce.com/vitrium/services/oauth2/token
Hide Logout Button in Webviewer (WV)	0 (Disabled)
After Logout URL	https://my.domain.com/login/
User Info URL	https://my.salesforce.com/vitrium/services/oauth2/userinfo
Skip Failed Page	0 (Disabled)
Failed Page Button Text	Continue
Remember Me	1 (Enabled)
Consumer Key	4MFVRG3982o5BBDd6wyHiPZVhY5_HwcIFc30sms9GAEhC8lWtFCEexWB7OmmH9grRFZZc0C.O72vnfcdAUSkfjoganw
Consumer Secret	8647375285085209615
Scope	Openid
Response Type	NULL
OAuth Result Type	Identity Field
Identity Field Name	[Email | Preferred_username]
Access Token in Header	0 (Disabled)
Prompt Login	0 (Disabled)
Authentication From Query Parameter	NULL
Trust Invalid Certificate	0 (Disabled)