Firefly Single Sign On (SSO) API

The Firefly Learning Platform integrates closely with existing school systems like the MIS, Active Directory and Google Apps so that all teachers and students at schools using the Firefly learning platform have an automatically managed account on the learning platform.

Firefly also provides an API to third party developers which allows third party products to perform single sign on for all users able to access Firefly already. This allows content providers, and creators of other third party web based products to allow access to students and teachers from schools using the Firefly learning platform without having to separately manage accounts or login, or themselves integrate with the MIS, AD or other services.

This in turn makes the third party products significantly more attractive to Firefly schools who don't have to manage accounts in multiple systems, and users who do not have to keep logging on to use separate services, often with different usernames and passwords they quickly forget. Third party systems are never provided with the user's password which helps improve security.

This document describes how to implement single sign on with Firefly using the SSO API. Please note that you will need to register your app or service initially with Firefly Solutions to be able to perform single sign on. Please contact partnerships@fireflylearning.com to do so.

To find out more about the Firefly Learning Platform, please see our website: http://fireflylearning.com.

Technical Overview

Firefly's SSO API has 3 main stages:

1. **Redirect to Firefly**: The third party service redirects the user to a specially constructed URL. The user will be asked to authenticate with Firefly if they are not already authenticated, and to approve their account details being passed to the third party.
2. **Redirect back to 3rd party**: After logon has been performed (if required), Firefly redirects the user back to the third party service via a requested URL, providing a secret token in the query string of the URL.
3. **Request user details from Firefly**: The third party service uses the token to make a server side HTTPS request to Firefly to gather information about the logged on user, including their name and e-mail address. This information can then be used to log the user onto the third party service and/or provision a new account as necessary.

These three stages are described in more detail below.

Although this flow is very similar to the authentication flow within OAuth 2.0, it is deliberately kept simpler than the OAuth 2.0 protocol. Unfortunately, this means it will not be possible to use existing OAuth 2.0 libraries. The SSO flow requires a server side component, this could be done from any programming language able to make and process outgoing HTTPS requests (PHP, C#, Ruby, Perl, etc).

Stage 1: Redirect to Firefly

To redirect the user to Firefly, you will need to know the hostname of the school's Firefly Server. This is unique to the school. An example for (the non-existant!) Maple Hill School might be

```
https://vle.maplehill.com
```

The URL to redirect to to begin the SSO process is

```
/login/api/webgettoken
```

So for Maple Hill School, the URL would be

```
https://vle.maplehill.com/login/api/webgettoken
```

There are also several required parameters that must be presented on the query string. They are as follows:

Parameter	Required	Description	Example
`app`	Yes	This is your app identifier, a fixed string provided to you by Firefly, unique to your application or service.	myapp
`successURL`	Yes	This is the URL you would like Firefly to redirect the user to on successful authentication. The hostname needs to be registered with Firefly and will be matched against your app code (above) for security purposes.	https://myapp.com/login/firefly/success
`failURL`	No	As above but used if your app identifier is not recognised or the user refuses to allow their details to be passed on.	https://myapp.com/login/firefly/fail

The parameters should be correctly encoded when added to the query string. An example full redirect URL for the above example data appears below:

```
https://vle.maplehill.com/login/api/webgettoken?app=myapp&succ
essURL=https%3A%2F%2Fmyapp.com%2Flogin%2Ffirefly%2Fsuccess
&failURL=https%3A%2F%2Fmyapp.com%2Flogin%2Ffirefly%2Ffail
```

Once it has constructed the correct redirect URL, the third party service should redirect the user's browser to the redirect URL, ideally using an HTTP status 302 (temporary) redirect. Firefly will then present a logon screen or an authorisation screen if necessary. If the user has already logged in and your service has already been authorised, Firefly will immediately redirect back as below.

Stage 2: Redirect back to 3rd party service

If Firefly successfully matches the app identifier and successURL, and the user is already logged onto Firefly or successfully performs a logon, Firefly will redirect the user back to the successURL provided by the third party service, with an additional URL parameter, ffauth_secret. So using the example information above, Firefly will redirect back to:

```
https://myapp.com/login/firefly/success?ffauth_secret=ABCXYZ
```

The ffauth_secret value is a string, and internally you should allow it to be at least 256 characters long. Please note, it may not always be 256 characters long.

Please note: if Firefly consistently redirects users to your failURL, this may mean your app is not correctly registered. Please contact Firefly partner support to check the status of your app registration.

Stage 3: Request information from Firefly server

Using the token provided on the query string in ffauth_secret, your app or service needs to make a server side HTTPS GET request back to the Firefly service to retrieve the user's information. As in step 1, you will need to know the hostname of the school's Firefly server. The URL endpoint to request the user's information is:

```
/login/api/sso
```

So in the Maple Hill example, this would be

```
https://vle.maplehill.com/login/api/sso
```

There are also two required parameters that must be presented on the query string. They are as follows:

Parameter	Required	Description	Example
`ffauth_device_id`	Yes	This should match your app identifier, as included as the "app" parameter to the initial redirect URL. It identifies your app.	myapp
`ffauth_secret`	Yes	This should match the token provided to your successURL in the ffauth_secret parameter.	AB243223ae3CXYZ

Please note that the ffauth_secret expires after 5 minutes, so please make sure you make this request in a timely manner after receiving the redirect back from Firefly. You should use the information that is retrieved to create a session in your service, and not call the SSO URL multiple times for a given user as it will time out quickly.

An example request URL using the dummy data above would be

```
https://vle.maplehill.com/login/api/sso?ffauth_device_id=myapp
&ffauth_secret=AB243223ae3CXYZ
```

Please note: this HTTPS request to retrieve user information should be done server side. Cross domain scripting policies mean that it is not possible to perform this request from JavaScript in the user's browser. You will need to use a server side technology like PHP, C#, Ruby on Rails or Perl. This also prevents the end user from tampering with the request or Firefly's response in any way.

If a valid ffauth_device_id and ffauth_secret are provided, the Firefly server will return an HTTP status code of 200 (OK) and provide a short XML document with user information in the response body. An example appears below:

```
<sso>
```

```
<user identifier="asdjADS989yhasd" username="johnsmith"
name="John Smith" email="john.smith@maplehill.com"
canSetTask="yes" />

</sso>
```

The identifier is a string which is a unique identifier for the user in the Firefly installation at that school (it may be, but is not guaranteed to be, globally unique). The username is a string of the username the user used to log on, although it is not guaranteed that this is unique even within a single school's Firefly, as multiple logon providers are supported. The name provides the user's name as a string and the email is the user's e-mail address as a string. Please note that Firefly may not have validated the user's e-mail address so you should not assume that the e-mail belongs to the user for authentication purposes. Finally, canSetTask is a string (equal to either yes or no) and indicates if the user has permission to set a task in Firefly. This can be used to determine if the user is a teacher or not.

If Firefly returns HTTP status 401 (Not Authorized), this indicates that either the ffauth_device_id or ffauth_secret parameters (or both) are incorrect. Please note that the ffauth_secret is only valid for 5 minutes after it has been generated. It is only designed to be used once. The third party application should then construct and maintain its own session information for the user, provisioning an account if necessary.

Getting Started

In order to start testing Firefly SSO, please contact partnerships@fireflylearning.com. We can provide you with access to a test Firefly system, and can register your app and provide you with an app identifier that you will need to perform the integration.