

Eval25 Single Sign-On with SOAP Framework

Overview

An important component of the Eval25 system is “single sign-on” functionality for your student, faculty, and staff users accessing the system. A prerequisite for this feature is your use of a web-based system that your users are already securely logging into. Examples of this include self-service functions within your Student Information System (SIS) and on-campus portals.

Assuming that you already have users logging into your system, CollegeNET® developers can build and leverage a trust relationship between your system and ours to make the user experience more pleasant. Your users can use the same usernames and passwords they’re already using, and they don’t need to log in a second time to access the Eval25 system.

This document describes the architecture for this single sign-on integration. If you do not have an on-campus portal or web access to your SIS, we can provide other means of authenticating your users. Email support@collegenet.com to make this request.

In This Article:

- [Overview](#)
- [High-level responsibilities](#)
 - [Your responsibilities](#)
 - [CollegeNET responsibilities](#)
- [General Assumptions](#)
- [System Interfaces](#)
 - [Authentication Transfer \(your system to the Eval25 system\)](#)
- [Logging Out](#)
- [Using the SOAP service](#)
 - [Summary of the Session service point](#)
- [Appendix A: Sample Java Code](#)
- [Appendix B: Sample Perl code](#)
- [Appendix C: Sequence Diagram](#)

High-level responsibilities

Your responsibilities

- Students, faculty, and staff must authenticate against an existing portal or SIS at your institution.
- Your technical staff must create a small authentication program responsible for establishing a temporary session with the Eval25 system and redirecting the user to the system.

- Your technical staff must create a button or link in the appropriate spot within your portal or SIS web that will initiate the authentication transfer.

CollegeNET responsibilities

CollegeNET is responsible for providing a SOAP interface for establishing authentication credentials. (SOAP, the Simple Object Access Protocol, is a method for allowing systems to talk to each other in a standardized fashion.) Authentication credentials will be trusted as long as they are established solely by your system through the “backchannel” provided by the Eval25 System.

General Assumptions

- At this time, the single sign-on service only applies to the Eval25 system.
- All interfaces between your system and the Eval25 system will use the SOAP protocol.
- The interfaces will be established over HTTPS, providing encryption of sensitive user data.
- The interfaces will be password protected and obfuscated for further protection.
- Your system will have only one username and password for accessing the Eval25 SOAP services, and that username and password will be easily configurable to facilitate rapid and periodic credential updates.
- The usernames and passwords of your users will never be sent to CollegeNET. Instead, you will send us each user’s unique student, faculty, or staff ID. We’ll use this ID to determine the user’s identity and permissions.

System Interfaces

Authentication Transfer (your system to the Eval25 system)

To facilitate the transfer of authentication from your system to ours, we employ a session token attached to an applyweb.com URL. This session token is only generated when your backend system talks to the Eval25 system.

This is the sequence flow for this interaction:

1. A user clicks the “Enter Course Evaluation System” link on your website.
2. Your website receives the browser request and initiates a “back-channel” request to the Eval25 system. This request:
 - Takes place via SOAP over HTTPS
 - Is authenticated with a username and password known only to your system and the Eval25 system
 - Includes information about the user being authenticated, most likely a student or faculty ID number (*ID_NUM*)
3. The Eval25 system creates a temporary authentication record for the user identified via the *ID_NUM*. This record includes a securely-generated unique token that expires after a 30-minute time out. Tokens aren’t reused, even for the same user returning at a later time.
4. The SOAP service returns a URL to your system that includes the address of the Eval25 system and the authentication token. This URL points to a location within the www.applyweb.com domain and specifies the HTTPS protocol. Example:

[https://www.applyweb.com/asp/evaluations?sid=g943gd\)9e9432asfnl0l](https://www.applyweb.com/asp/evaluations?sid=g943gd)9e9432asfnl0l)

Note: If an *ID_NUM* doesn't exist within the Eval25 system's available course evaluation data, we still return a URL. In this case, the downstream Eval25 system provides a message to the user similar to, "There are no evaluations for you to fill out at this time."

5. Your website redirects the user's browser to the URL returned by the SOAP service. This completes the transfer of authentication to the Eval25 system. The token provided with the URL is checked for validity and becomes an authentication cookie within the user's browser.

6. This authentication cookie remains active as long as the user stays active. "Active" is currently defined as taking an action within three minutes of the previous action taken.

Logging Out

Logging out doesn't require any specific interactions between your system and ours. Typically, Eval25 system pages are loaded into a new window, and returning to your site is accomplished by closing this window.

When the window is closed:

- The authentication tokens expire and are deleted on our side.
- Authentication cookies within the user's browser expire and become useless.
- Getting back to the Eval25 system requires clicking on a link or button on your website which restarts the authentication handoff process.

Using the SOAP service

Sample code for using the SOAP service with Java is in Appendix A, and sample code with Perl in Appendix B. This is a summary of the Session service point:

Summary of the Session service point

Type	<i>wrapped</i>
Host	<i>pubsoap.applyweb.com</i>
Port	<i>443 (https)</i>
Service	<i>/evaluations/Session</i>
Method	<i>createCourseEvaluationSession(String id, String salt)</i>
Return	<i>String</i>
Auth	<i>HTTP Basic + SSL</i>

id is the user's unique student or faculty ID_NUM.

salt is the User Agent string of the user's browser. This helps us ensure that the browser making the request to your system is the same one that ends up on our system.

Appendix A: Sample Java Code

Below is sample Java code to invoke the Session service for the Eval25 system at ApplyWeb. This service sets up

the evaluation session for a given user (identified by userid) and returns a URL string. The user's browser is then redirected to this string, which takes him/her into the Eval25 system.

Note that the salt is required and must be the User-Agent string exactly as reported by the user's web browser. Also, this code isn't doing any error handling or other best practices; it only provides basic connectivity.

```
import java.net.URL;

import javax.servlet.http.HttpServletRequest;
import javax.xml.namespace.QName;

import org.apache.axis.client.Call;
import org.apache.axis.client.Service;

public class Demo {

    /**
     * Invokes the session Applyweb SOAP Service.
     *
     * @param userId Your user's unique id
     * @return An Applyweb URL we should redirect
     * the client's browser to.
     */

    public String getApplywebSession(

        HttpServletRequest request,
        String userId)

        throws Exception {

        String username = "..."; // To auth your system to Applyweb
        String password = "..."; // To auth your system to Applyweb
        String target =

            "https://pubsoap.applyweb.com/evaluations/Session";

        String qname = "http://com.collegenet/";

        String method = "createCourseEvaluationSession";

        Call call = (Call) new Service().createCall();
        call.setUsername(username);
        call.setPassword(password);
        call.setTargetEndpointAddress(new URL(target));
        call.setOperationName(new QName(qname, method));

        String salt = request.getHeader("User-Agent");

        return (String) call.invoke(new Object[] { userId, salt });
    }
}
```

Appendix B: Sample Perl code

What follows is sample Perl code to invoke the Session service for the Eval25 system at ApplyWeb. This service sets

up the evaluation session for a given user (identified by userid) and returns a URL string. The user's browser is then redirected to this string, which takes him/her into the Eval25 system.

Note that the salt is required and must be the User-Agent string exactly as reported by the user's web browser. Also, this code isn't doing any error handling or other best practices; it only provides basic connectivity.

```
#!/usr/bin/perl use strict;
use SOAP::Lite;

my $userId = "..."; # Your user's unique id
my $redirectURL = getRedirectURL($userId);

# Redirect the browser to $redirectURL...

sub getRedirectURL {

    my $id = SOAP::Data->type(string => shift);
    my $username = "..."; # To auth your system to Applyweb
    my $password = "..."; # To auth your system to Applyweb
    my $salt = $ENV{'HTTP_USER_AGENT'};

    my $proxy = "https://";

    $proxy .= $username . ":" . $password;

    $proxy .= "@pubsoap.applyweb.com/evaluations/Session";

    my $response = SOAP::Lite

        -> uri('http://com.collegenet/')
        -> proxy($proxy)
        -> createCourseEvaluationSession( $id, $salt )
        -> result;

    return $response;
}
```

Appendix C: Sequence Diagram

