ENTAA Developer Guidebook

Matthew Harshbarger
2016/05/18 10:58

Table of Contents

Contents

Purpose of this document

The Client Implementation Guide provides information on the supported ways of using the Enterprise A&A service from a web or desktop application. It gives diagrams and step-by-step instructions to help you plan a successful deployment.

Intended Audience

- **Architects**: To understand the components and interactions required for each scenario.
- **Developers**: To see the set of API calls made and any redirects that are required.
- **Project Managers**: To understand the terminology of an A&A implementation and help decide which is best for the project.
- **Testing Staff**: To understand the set of interactions between systems and plan for testing against those boundaries.

Prerequisites & References

Before using this document to evaluate or plan your A&A implementation, you should read and understand the information in the following documents, which can be found in the SDK package on the ENTAA Forge site:

Title	Description (link)
ENTAA Service	Provides an overview of the service, its components and basic concepts. Good for presentations to new A&A implementers and managers.
Enterprise A&A executive-summary	An executive-summary format. Provides much of the same information as the presentation above,

but in a more narrative format.

Provide high level diagrams showing the various A&A components and how they connect to one another.

Details the set of public functions that are available to users of the A&A service. Includes input and output data structures and any exceptions that may be thrown by each function.

Note: This document has been migrated to the wiki and will be maintained online. This document

will no longer be included in the SDK.

Contains detailed layouts and potential values for the elements and attributes that comprise each type of service message.

Note: This information is now included in the API Specifications, above and the values might not be as current as those found in the API Specifications.

Contains design flows that describe how applications interact with A&A as well as how A&A will process

requests.

Note:
This
information
only
applys
to
v3
of
A&A.
Version
v3
of
A&A
will
no
longer
support
the
reuse
of
an
authentication
tokens
as
has
been
the
case
for
all
previous
versions
of
A&A.

Scenarios

- **Common Steps**: These guidelines are shared by all implementation scenarios below.
- **Common Interface (CI)**: This is generally the simplest scenario, in that only two or three client library methods are required to integrate with A&A. This scenario has the added benefit of allowing users of one site to be logged in automatically at another (Single Sign-On or SSO).
- **Custom Logon**: This scenario makes full use of the client library and allows applications to make authentication and authorization calls directly against the service.
- **Custom Self-Registration**: This scenario allows applications to create new user accounts. These accounts can be re-used by other applications that share the same *repository*.

Common Steps

You should follow these steps no matter which implementation scenario you intend to follow.

1. **Submit an Application Request**
   Using the online form on GForge, you can specify a **short description** for the application (generally 20 words or less), **App ID** you'd like, and other options such as Common Interface (CI), self-registration, etc. The **App ID** is a unique identifier for your application and is generally composed of the requesting agency and an abbreviation or acronym for the calling system (e.g., ITE_BUGS, DOM_BUDG). This ID must be eight characters or less and cannot include spaces or special characters (e.g., asterisk, exclamation point, ampersand)
   Note: The A&A Technical Team can help you decide if CI/SSO features are appropriate for your application.
   Note: With the addition of Active Directory Application Mode (ADAM) to the A&A service, almost every application can now use a single provider for both internal and external users. The A&A Technical Team can help you decide if you have special needs in this area.

2. **Set up Key Users**
   For each application, there are certain key users that help to set up the privileges and implement them in the application code. They are usually development staff, but can also include business owners, project managers and key customer staff.

 When an application is set up, one or more accounts are granted the **A&A Administrator** system privilege. This allows those users to perform any A&A administration activity for the new application. This can include creation and assignment of privileges, creation of accounts, even locking and unlocking of accounts. By granting other users one or more system privilege, the A&A Admin user can delegate these functions to other team members. This is the preferred method of operation, since careless use of the A&A Admin privilege can change information that the system relies on for operation (such as privilege codes, privilege assignments, etc.).
Note: A&A Admin-privileged users are allowed to un-set their own system privileges, including the A&A Admin privilege itself! If this happens to you, call the Service Desk (515-281-5703).

3. **Create Application-Specific Privileges**
   Using the ENTAA Admin website, key users can create a list of privilege codes and corresponding descriptions. These privileges can be assigned to other user accounts (through ENTAA Admin or the Web API) and checked by the application to control access to information and functions. The list of application-specific privileges is delivered by ENTAA as part of the authenticated AAUser object.

Summary of Setup Tasks

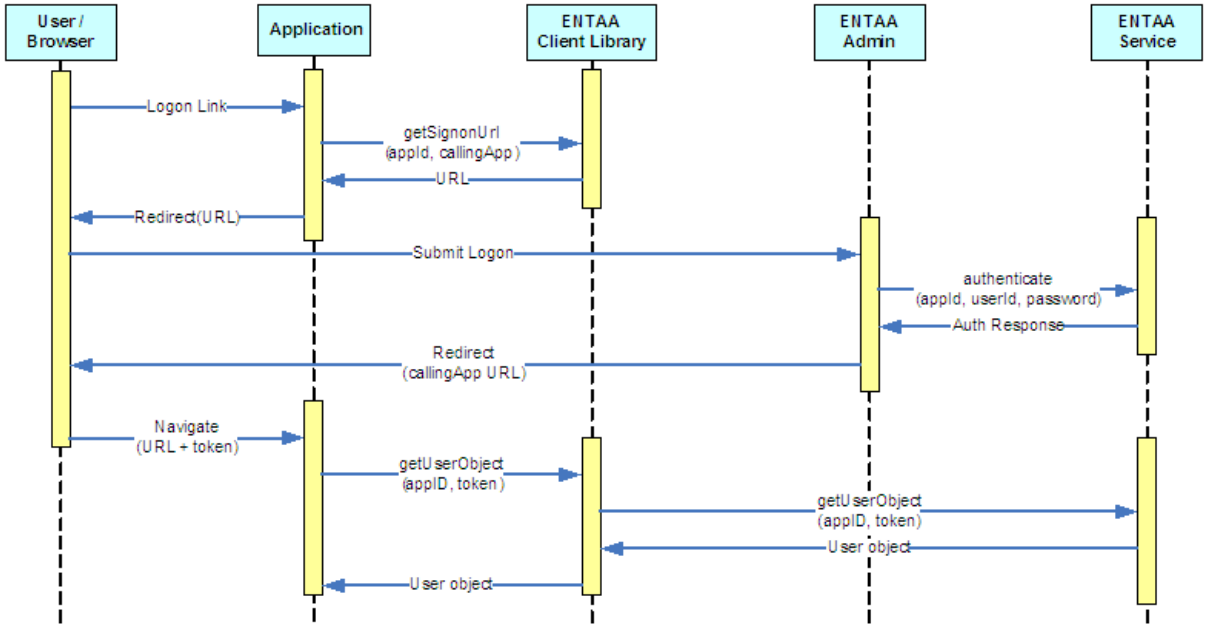
By	Task
A&A Technical Team	Create new App ID. Configure the system in DEV and TEST, and assign one or more A&A Admins for the application.
A&A Admin	Assign App Owner and Privilege Manager privileges to customer's designated accounts.
Privilege Manager	Create the list of application-specific privilege codes.
App Developer	Implement authentication using A&A client

App Developer
- Implement logic checking for specific privilege codes in appropriate sections of application.

Common Interface

In this scenario, the application does not authenticate the user directly with the service. Instead the application passes to user off to the A&A website for authentication. The A&A website validates the user's credentials and passes the user back to an app-defined URL within the application, including a unique session token for that user. The application then validates the token with the A&A service to obtain the user's profile and privilege information.

The main benefit of this scenario is the small effort required for full A&A functionality. There are no application-specific web pages to design or render and only one call to the A&A service itself is required. All account management activities are handled automatically by the A&A website. The calling application is sent back an authenticated user in every case.



Sample Code (C#)

First, you'll need to send the user to the ENTAA Common Interface logon page:

```
// Get AAHost, AppId and callingApp from your specific app/environment
AAService.Host = AAHost;
string redirectURL = AAService.Instance.getSignonUrl(AppId, callingApp);
Response.Redirect(redirectURL, true);
```

When the user comes back, the GET string will contain a token called tokenId. Use this value to validate the user and get their information:

```
try {
    // Get AAHost and AppId from your specific app/environment
```

```
    AAService.Host = AAHost;
    // Scrub tokenId appropriately before using!
    string token = Request.QueryString["tokenId"];
    AAUser user = (AAUser) AAService.Instance.getUserObject(token, AppId)
} catch (Exception ex) {
    // The ENTAA service went away, or the token is bad.
}
```

In order to enable self-service, and as a minimal step to preventing automated or bogus account creations, all Enterprise A&A accounts **require** a valid e-mail address. Validation of the e-mail address is an integral step in the registration process, and cannot be turned off or worked around. This requirement is part of the process based on a consensus of the State's technical and security staff, and customers at the time that self-registration was implemented.

Common Feature Locations – Shared Authentication

	Name	**Notes**	**Location**	**Handled By**
A&A Website	Require logon	Requires the calling app to specify a URL for return when the logon process is complete.		
N/A	Create Account	Creates a new user, available via SA.		
A&A Website	Identity Baseline	Identity proofing (first-time user) is part of the logon redirect.		
A&A Website	Forgot Password	Part of the logon redirect.		
A&A Website	Multiple-site shared logon	Single-sign-on (SSO) exists, then the SA web pages		

will
not
prompt
the
user
for
login.
It
will
return
the
existing
token
back
to
the
calling
application.
This
is
seamless
to
the
user.

ENTAA HTTP Module

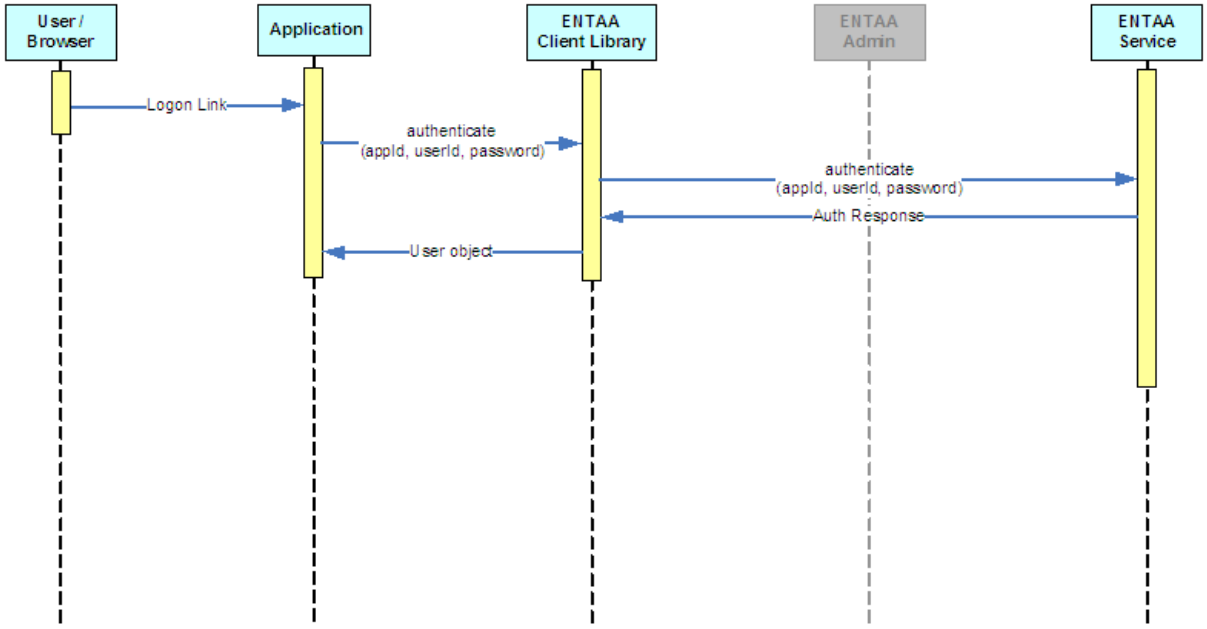
If you are developing .NET C# web applications we have a set of DLLs we are calling "ENTAA HTTP Module" that you can drop into your project and get going faster with A&A. Many of the .NET C# web applications we have produced over the years have used these DLLs but around March of 2016 we updated these in an effort to address session sharing between apps. While working on a solution for this we updated the existing DLLs and implemented a few new features and addressed a few security issues. Apps that used the old DLLs can simply use the new DLLs with only the need to rename a few imported namespaces. The old DLLs also had a dependency on an older version of Log4Net which has been removed in the new DLLs. To find out more about the "ENTAA HTTP Module" click here.

Custom Logon

In this scenario, the calling application creates a form in which it prompts the user for username and password. The application then calls the authenticate method on the A&A Service, passing its AppID and the credentials supplied by the user.

The main benefit of this scenario is the flexibility that the calling app can exercise in the layout and functionality of the logon process. In addition to prompting for A&A-required data, the application can also prompt for app-required information (e.g., license number, date of birth, other) and can even self-register the user automatically if desired.

The main drawback to this method is the fact that the application must plan for and handle various return codes for the authenticate method, including first-time user scenarios, locked accounts, and invalid credentials. The way in which an application chooses to handle (or ignores) these exceptions can create subtle loopholes in its security and undermine the usefulness and trustworthiness of the A&A Service altogether. For this reason, we recommend the Shared approach for all applications that do not have a specific reason for implementing their own logon page.



Interactions between A&A components in Custom Logon scenario

If your application creates accounts using the *createAccountByAdmin()* web service method, you must specify the correct suffix for the account (@IowaID). It is possible to specify a different suffix, but this will not be consistent with accounts created via the Shared Authentication website, and will cause confusion if and when such accounts are re-used by other applications. Application owners that use the Shared Account Repository should *plan* on accounts being re-used. The *createAccountByAdmin()* web service method will be updated to throw an error if a custom suffix is submitted.

Common Feature Locations – Custom Logon

Notification By	
Calling App	Requires the calling app to catch all documented exceptions and have a process for handling each.
Calling App	Create the account (createAccountByAdmin function).
Calling App	Identity Baseline (first-time user)
	AccountNotInitializedException returned by the authenticate method, the

	caller must redirect the user to the A&A Website. There is no API method for programmatically setting or reading the baseline data.
Call Logon App	Logon Password Identity Baseline, above.
Multiple-Factor Authentication (MFA) Not available to Custom Logon applications.	This feature is shared logon (SSO)

A&A Images

We suggest you download the following collection of A&A images for use on your site to ensure consistency between apps using A&A. entaa_buttons.zip