AgileApps Cloud

The Fast Path to
Agile Process-Driven Applications

Get There Faster

Client Certificate Authentication
(aka “Custom Headers”)
in a private AgileApps Cloud

Contents

Client certificate authentication adds custom headers to each HTTP request, giving users the ability to access the platform without requiring a login step. Instead of logging in via the login screen, using username/password combination, the platform checks for custom headers in the HTTP request to authenticate the user.

Learn more: http://en.wikipedia.org/wiki/Public_key_certificate

1. Data Flow

1. User obtains Digital SSL Certificate and installs it in the (web browser) client.
   Learn more: https://en.wikipedia.org/wiki/X.509#Structure_of_a_certificate
2. When the client makes an HTTP request, the certificate is included.
3. The on-premise installation includes an authentication server in the network architecture. That server validates the certificate and adds a custom header to the HTTP request that contains a key/value pair. (The Apache server can be used for that purpose, as well as other servers.)

If this step fails, the user sees the error returned by the authentication server.

4. The AgileApps Cloud platform reads the information in the header and identifies the user who is making the request. (The information must uniquely identify the user.)

If this step fails in a browser, the user is directed to the login page. If in a program that is using the REST API, a "user not found" error is returned.

2. REST API

Once SAML sign-on has been established, a client program can use the platform's REST APIs to log in.

Learn more: http://agileappscloud.info/wiki/REST_API:samlAssertion_Resource

3. Network Setup

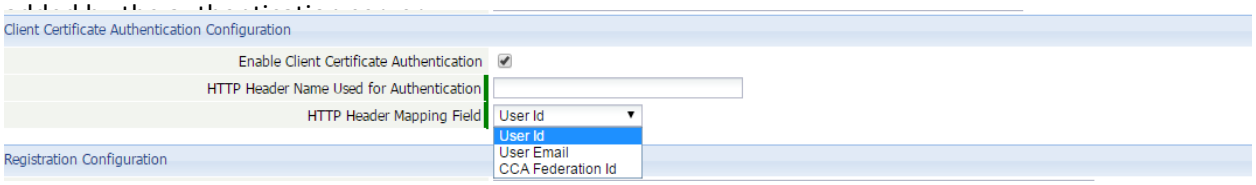
Configure the network architecture so that incoming HTTP requests go to the authentication server, which validates the user's certificate, adds the custom header with the key/value pair expected by the platform, and passes the request on to the application server.

4. Platform Setup

This step identifies the key/value pair that the authentication server adds to the HTTP header.

1. Go to the Admin tenancy. On the Service Configuration page, click the option that enables certificate authentication:

2. When enabled, the platform asks for an additional two parameters that describe the key/value



4.1 Parameter details:

- **HTTP Header Name Used for Authentication**
  The name of the *key* in the HTTP header.
- **HTTP Header Mapping Field**
  Specifies the kind of value:
  - **UserId**
    This is the globally unique user identifier (GUID) defined in the platform when the User record is created. If the authentication server can be configured to provide it, this value provides the best performance and reliability.
  - **User Email**
    When this value is provided, User records are searched for a matching email. The search succeeds when there is one and only one User with that email.

    Note:
    In test and development systems, it is typical to have multiple "personas"—each with a different username and application role, but all with the same email address. In that scenario, the user's email address fails.
  - **CCA Federated Id**
    When certificate authentication is enabled, User records have an additional "CCA Federated Id" field . It might contain a social security number, phone number, or some other unique identifying string. This selection causes the platform to search that field for a User with a matching identifier.

In the platform, the User's value can be specified in the UI or using the REST APIs. The authentication server must then provide the same value for the match to succeed.