TRAITWARE™

Traitware Authentication Service Integration Document

February 2015

V1.1

Acuity SYSTEMS

“Secure and simplify your digital life.”™

Integrating Traitware Authentication

This document covers the steps to integrate Traitware Authentication into your existing web applications. To do so you will need to become a little bit familiar with OAuth 2.0 if you are not already. Fortunately there are several OAuth 2.0 open source libraries written in various languages depending on your server-side requirements.

There are three main steps to integrate with Traitware:

1. Add an application to your Traitware Customer Console. This is where you register the application you are going to protect.

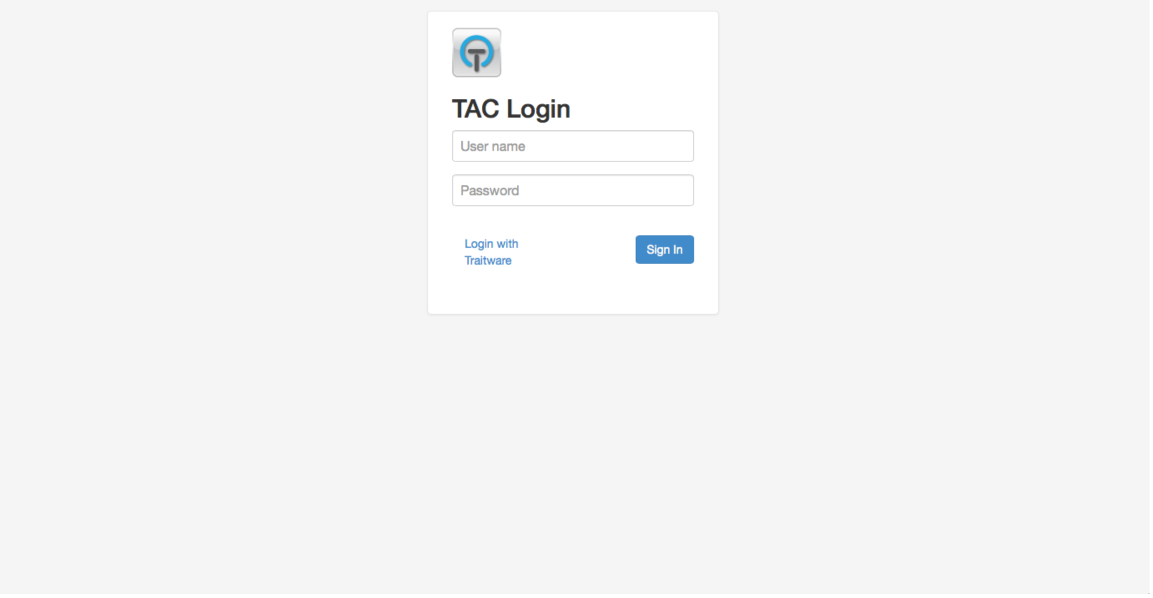
2. Add your users to the Traitware system and set their permissions. This can be done through our APIs or through the Traitware Customer Console web site.

3. Create the necessary modifications to your webserver and backend to handle the OAuth 2.0 authentication flow.

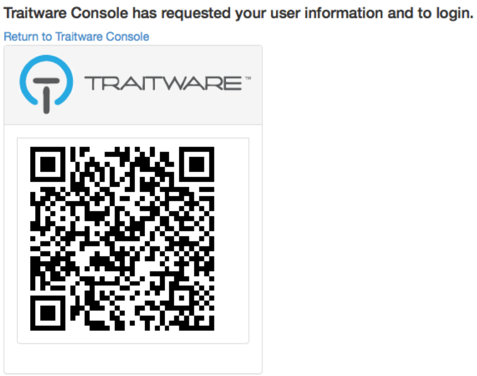
Add an Application to the Traitware Customer Console

You are required to use the Traitware app to login to your Traitware Customer Console (TCC). When you first establish a customer account with Traitware you should receive an activation code to activate the Traitware app. Once activated you will be able to access your TCC. In the future you can choose to give TCC access to other administrators when you create their accounts.

1. Go to **tcc.traitware.com** and click 'Login with Traitware"



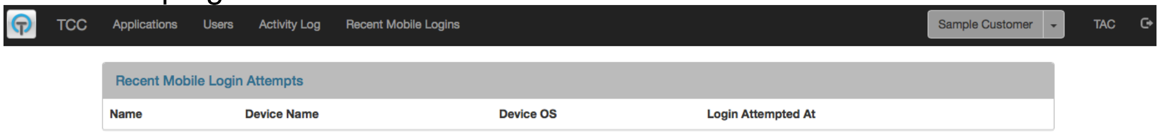
2. Login to the TCC by scanning the QR code using your activated Traitware app.



Traitware Console has requested your user information and to login.

Return to Traitware Console

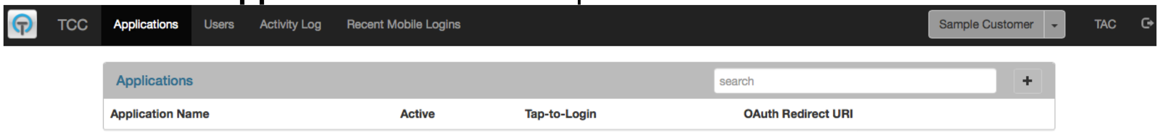
3. Once logged in you should see a screen like this. Your customer name should be in the top right.



Recent Mobile Login Attempts

Name	Device Name	Device OS	Login Attempted At

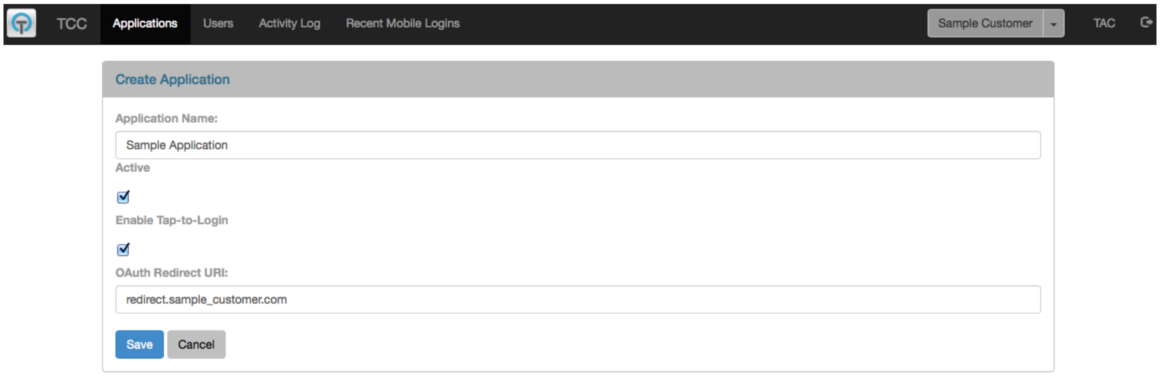
4. Click on the **Application** tab on the top bar and click on the “+”.



Applications

Application Name	Active	Tap-to-Login	OAuth Redirect URI

5. Create your application and click ‘Save’.



Create Application

Application Name: Sample Application

Active ☑

Enable Tap-to-Login ☑

OAuth Redirect URI: redirect.sample_customer.com

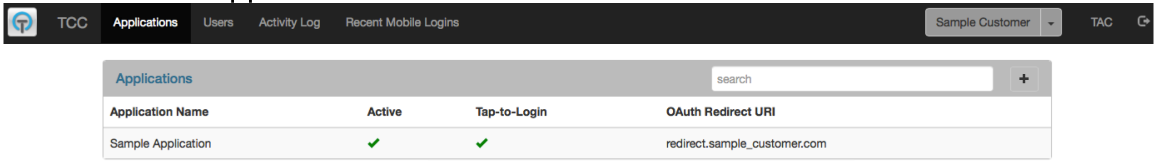
To create your application you must provide two items:

1. The name of the application (Application Name) you are going to be using with Traitware. This can be anything but should indicate the application.
2. An OAuth Redirect URI. This is the URI Traitware will redirect to after receiving an authentication attempt.

You also have two additional options that can be changed at any time:

1. Toggle the Active/Inactive state of your application. Making an application inactive will block any attempts to access it via Traitware.
2. Enable Tap-to-Login. You can choose whether or not to make this access method available to your users. When inactive, you will not see a field to enter an email address when trying to log in. The default for this feature is ON.

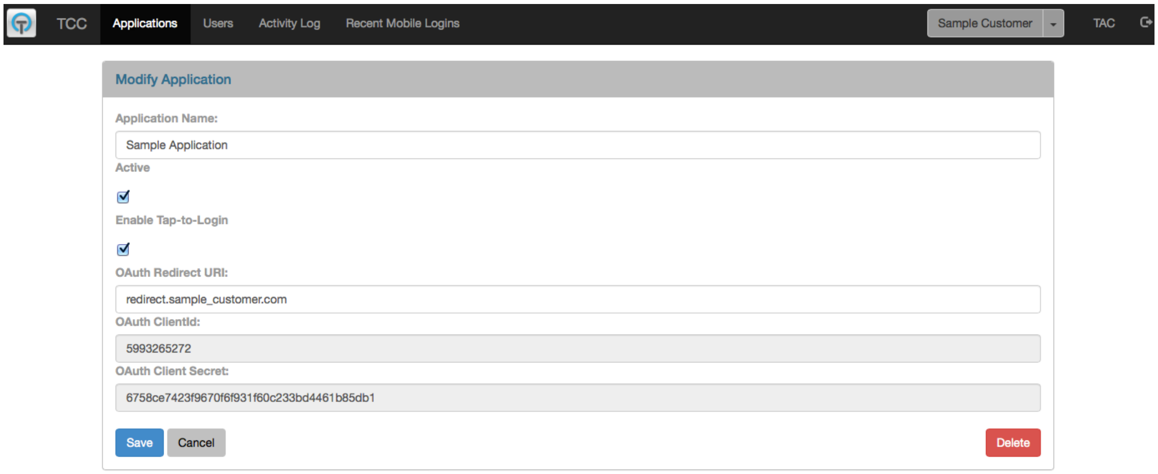
6. You will see your application listed under your application list. You can always add additional applications.



Applications

Application Name	Active	Tap-to-Login	OAuth Redirect URI
Sample Application	✓	✓	redirect.sample_customer.com

7. Click on your application to find your **OAuth ClientId** and **OAuth Client Secret**. These values are used for the OAuth 2.0 authentication flow.



Modify Application

Application Name:
Sample Application

Active
☑

Enable Tap-to-Login
☑

OAuth Redirect URI:
redirect.sample_customer.com

OAuth ClientId:
5993265272

OAuth Client Secret:
6758ce7423f9670f6f931f60c233bd4461b85db1

Add Users To Traitware

The Traitware service offers several APIs to register users. You can also add them manually or through a bulk .csv import. Please see the Traitware Customer API Document for a thorough overview of the available APIs.

When you register a user with Traitware you will receive a `traitwareUserId` in the response. This value should be associated with the user in your database that you are registering. You will need to reference this value during an authentication attempt. This value is mirrored on your server and the Traitware server as a common reference point for each user.

Once a user is registered you will need to send Activation Codes to your users to activate their Traitware app. When you do they will be sent an Activation Code via SMS to the number registered with their account. The Activation Code can be sent any time after a user has been registered and can be triggered via an API call or manually in the TCC under each user account.

The user can enter the Activation Code into the downloaded Traitware app to activate their account on that device. **Please note that the Activation Code has an expiration time of 48 hours. You will have to issue a new code via the Traitware API if the app is not activated in that time period.

The Traitware app can currently be downloaded from the Apple iTunes Store and Google Play. Search for “Traitware Authentication” at either site.

Create the OAuth 2.0 Integration

Integrating the OAuth 2.0 flow into your existing web application will provide the option to direct your web application to the Traitware authentication server for login authentications. It will also allow your server to talk directly to the Traitware server to verify an authentication attempt.

These are the general steps required for this integration. The sections that follow will contain additional details

1. Add a snippet of JavaScript to the HTML on your webpage. This is to add the ‘Login with Traitware’ button that will redirect to the Traitware authentication site when clicked. Your web server needs to generate and store a STATE value, which is like a session identifier, to protect against Cross Site Request Forgery. This value is used by the JavaScript snippet when directed to the Traitware authentication site.
2. Create a redirect endpoint URI on your webserver for the OAuth 2.0 flow. This would be something like www.yoursite.com/auth.

3. Consume the parameters passed back to the redirect URI (Authorization Code, State Value, Traitware User Id) during an authentication attempt. You should verify the STATE value here to make sure it is valid.
4. Implement a call to the Traitware server to exchange the received Authorization Code for a Traitware Authentication Token. If a Token is successfully received, the user has been authenticated.
5. Upon receipt of the Authentication Token, advance your user to the authorized resource they are attempting to access.

Step 1: Add JavaScript to your Login Page and Generate STATE Value

The following needs to be added to the HTML in the login page where the "Login with TraitWare" button will appear:

```
<div id="TraitwareLoginButton"></div>
```

This needs to be added to the bottom of that page:

```
<script type="text/javascript">
TW = Object();
TW.clientId = "CLIENT_ID";
TW.state = "STATE";
</script>
<script src="https://customer-
api.traitware.com/twlogin.js"></script>
```

Be sure to substitute CLIENT_ID and STATE with your own values during page render. CLIENT_ID is your application's OAuth ClientID available in the Traitware Customer Console. The STATE value needs to be generated by your web server as a protection against Cross Site Request Forgery. We suggest a random UUID of sufficient length to not be easily compromised.

For example, a JavaScript UUID generator for STATE value can be found here: https://github.com/broofa/node-uuid

Step 2: Create a Redirect URI on your Webserver

Create a web address that the Traitware server will call to pass an Authorization Code, a State value, and a Traitware User ID during an authentication attempt. This could be something like www.yoursite.com/auth. Remember to add the Redirect URI to your application in the TCC.

Step 3: Consume Authentication Parameters Passed to the Redirect URI

A request to your Redirect URI would look something like this:

```
www.yoursite.com/auth?code=dc436f9h4648…&state=57eb45ac8…&traitwa
reUserId=8de56c4a-55b8-44df…
```

With parameters:

```
?code=                # the AUTHORIZATION_CODE
&state=               # the CSRF value
&traitwareUserId=     # the identifier linked to the authenticating user
```

AUTHORIZATION_CODE – The authorization code is used to query the Traitware server for an Authentication Token. If you receive an error, the person was not authenticated and no further action should be taken.

STATE (CSRF value) - Your web server should verify that the STATE value received here is valid. If it is not an active STATE value the session may have been compromised and no further action should be taken.

TRAITWARE USER ID – This is the user identifier you receive when you register a user with the Traitware service. This should be linked to your user's account in your database. When you receive a request to your Redirect URI, this value is passed to let you know which user is attempting to authenticate to your site.

Step 4: Call Traitware Service For Authentication Token

Once you have received the Authorization Code and have verified the STATE you should request an Authentication Token from Traitware. You can do this via a JSON request or as an x-www-form-urlencoded request. This request is **never passed through the browser**, but through a direct server-to-server query.

A request for an Authentication Token would follow one of these formats:

JSON Format

```
[Content-Type: application/json]
Request: POST https://customer-api.traitware.com/oauth2/token
{
    "client_id":  { type: String, required: true },
           # from the TCC for this application
    "client_secret": { type: String, required: true },
           # from the TCC for this application
    "code": { type: String, required: true },
           # AUTHORIZATION_CODE
    "grant_type": "authorization_code"
}
```

x-www-form-urlencoded Format

```
[Content-Type: application/x-www-form-urlencoded]
Request: POST /token
QueryString Params:
    ?client_id=CLIENT_ID
           # from the TCC for this application
    &client_secret=CLIENT_SECRET
           # from the TCC for this application
    &code=AUTHORIZATION_CODE
    &grant_type=authorization_code
}
```

Remember that the `client_secret` can be found under your application in the TCC. This parameter allows your server to contact the Traitware server directly. **This value is very important and needs to remain secret.

Here is an example of a successful response:

```
Successful Response: (Content-Type: application/json)
{
    "access_token": { type: String, required: true },
    "timeout": { type: Integer, required: true },
    "state": { type: String, required: true }
}
HTTP Status 200
```

If you receive an `access_token` in the response, the user has been successfully authenticated.

Step 5: Grant User Access to Authorized Resource

Once you have received an `access_token` you know the user has been authenticated with Traitware. You can now direct their web browser to the page they are requesting access to.

You should use the `traitwareUserId` received at the Redirect URI to bind your user to the authentication session. In other words, you should grant access to the resource requested by the user associated with that particular `traitwareUserID` in your database. This assumes the user has authorization for that resource.

API Documentation

Please contact a representative from Traitware for the API documentation.

support@traitware.com
530.802.1615