

Captive Portals

How To: Social Media Onboarding using OAUTH

April 2016

Revision A

© 2016 ZIH Corp. All Rights Reserved. Zebra and the Stylized Zebra Head are trademarks of ZIH Corp, registered in many jurisdictions worldwide. All other trademarks are the property of their respective owners.

Table of Contents

Table of Contents.....	4
1. Introduction	5
2. Technology Overview	5
3. Architecture / Design	6
3.1 Overview.....	6
3.2 Login flow	6
3.3 Steps to configure OAuth based captive portal	8
3.3.1 Create a Client ID.....	8
3.4 Captive Portal OAuth Setup (AP or Controller)	16
3.5 Create a dns-whitelist	17
3.6 Create WLAN:	17
4. Management.....	18
4.1 CLI	18
5. Test Framework.....	19
5.1 Unit Testing	19
6. References	19

1. Introduction

Oauth (Open standard for authorization) support for captive portal allows users to sign in onto Guest WLANs using their Google or Facebook credentials using any type of wireless clients (laptops/tablets/phones etc). This will basically ensure that users or customers walking into a retail store will be able to connect to the store's WiFi network using their Google/Facebook accounts. This will help the Stores publicize themselves on the Facebook News Feed of users thereby providing more social followership. It will also help in analytics reporting.

2. Technology Overview

This overview assumes a basic working knowledge of Hotspot/Captive Portal authentication methods and the authentication flow. Similar to captive portal using email/SMS validation or radius based authentication methods, there will be another way to do captive portal i.e using Google or Facebook credentials.

OAuth is an open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications. It works by delegating user authentication to the service that hosts the user account (Facebook/Google), and authorizing a third-party applications to access the user account in a limited way on an HTTP service. The third party application in this scenario will be the WiNG software running on an access point or a controller assigned to do the captive portal authentication for the client. A wireless client trying to log in onto a Guest WLAN using this method will be asked for his Google/Facebook credentials and allowed network access upon successful authentication.

After the user authentication is complete, additional API calls are made to obtain the user's public profile and email address. The WiNG app can only request for certain fields of the user's profile. They are:

- id
- name
- first_name
- last_name
- link to the profile picture
- gender
- locale
- email address

Please note that the user credentials are not stored by the AP or Controller. Also, the WiNG Application does not receive any of the information not shared publicly by the user.

3. Architecture / Design

3.1 Overview

The OAuth support for wireless clients will be tied with the Captive Portal feature on the APs and controllers. Logging in using the client's Facebook/Google account will serve as an additional mechanism for doing the captive-portal authentication.

3.2 Login flow

Once a wireless client associates with an AP on a wlan with captive-portal enforced and OAuth is configured to be included in the login page, the login screen popped up on the client browser will have the Facebook & Google login buttons on it. The user can click on any one of the above (based on whichever account he has or prefers). A popup will then appear asking the user to enter his credentials and hit OK. If the credentials entered are incorrect, the popup complains and asks the user to enter the correct credentials again.

Organization Name

Welcome to Guest User Wireless Service.

Please enter the username and password to sign-in.

Username

Enter Username

Password

Enter Password

Sign In Clear

(Or) Sign in using,

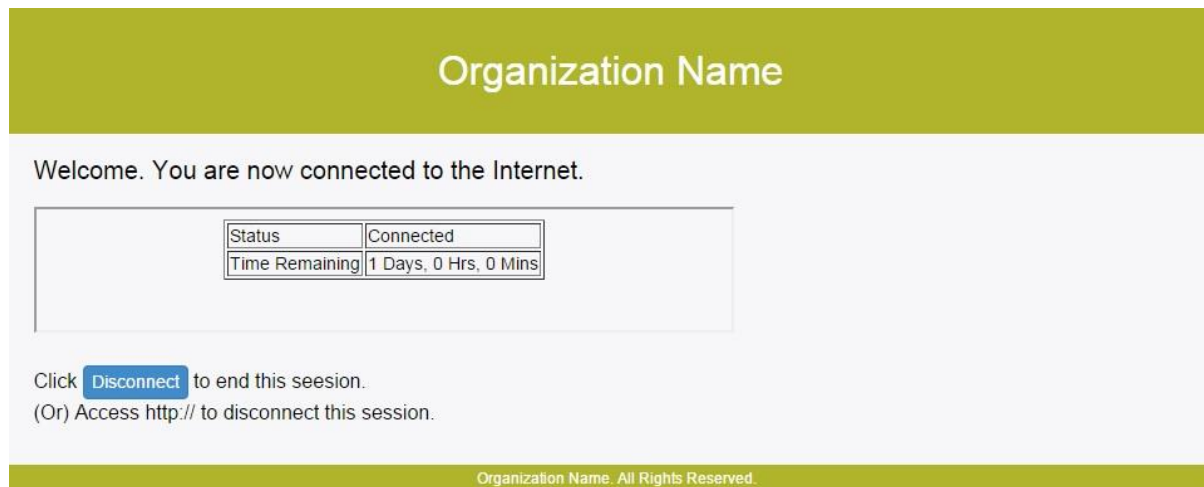
 Sign in with Facebook

 Sign in with Google

Organization Name. All Rights Reserved.

Login screen on the client browser using OAuth based captive-portal authentication

If the credentials are correct, the login is successful and the client is redirected to the welcome screen. The client's captive portal state now changes to SUCCESS (from PENDING) and he is now able to surf the internet.



Welcome screen on the client browser after a successful login.

3.3 Steps to configure OAuth based captive portal

3.3.1 Create a Client ID

You must create a Google/Facebook Client ID for the WiNG application running on the AP/controller to be able to do the Authentication on behalf of the user. This registration process will yield two strings, *Client ID* and *Client Secret*. You will use these strings to authenticate and gain access to Facebook & Google OAuth APIs.

- a. For Google, use the [Google Developer Console](https://console.developers.google.com/)
- b. For Facebook, use the [App Dashboard](https://developers.facebook.com/apps/) (Facebook calls it the App ID)

Google:

For Google, go to the Google Developer Console - <https://code.google.com/apis/console>.

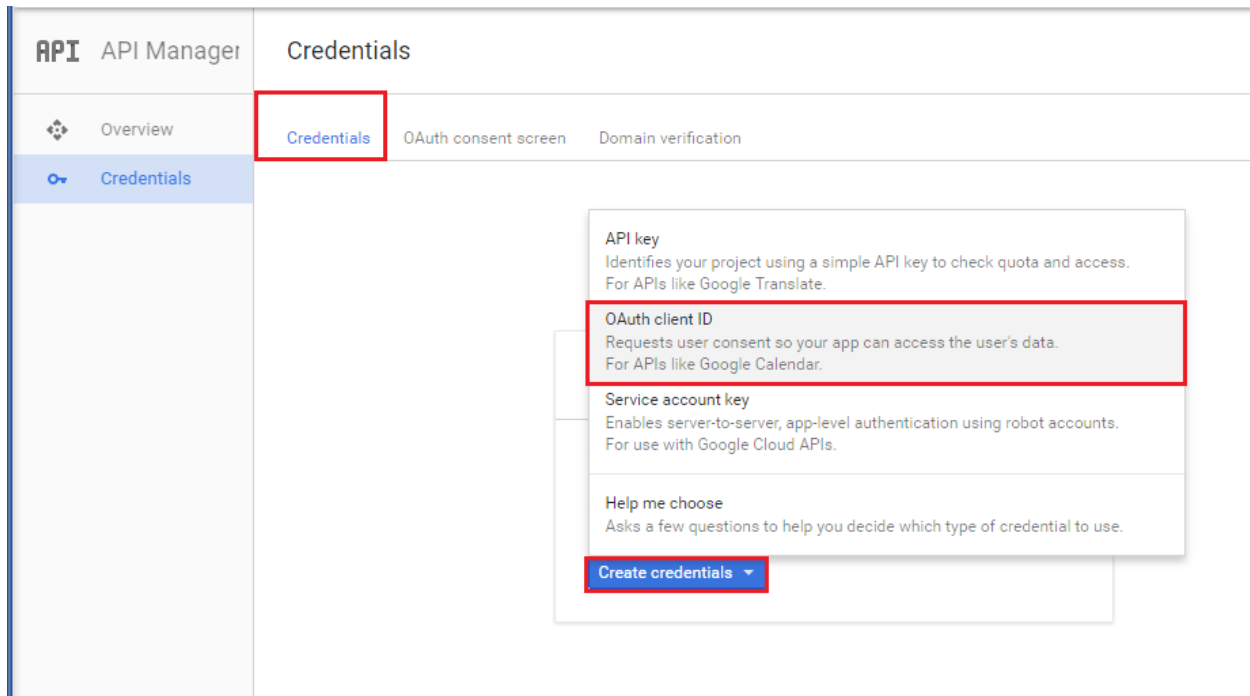
1. Edit the **consent screen**. Give a **product name**.

The screenshot shows the Google Developer Console interface for configuring an OAuth consent screen. The left sidebar has 'API Manager' and 'Credentials' tabs, with 'Credentials' selected. The main area has three sub-tabs: 'Credentials', 'OAuth consent screen' (which is highlighted with a red box), and 'Domain verification'. The 'OAuth consent screen' form includes the following fields:

- Email address**: A text field containing 'guestadmin@zebra.com'.
- Product name shown to users**: A text field containing 'GuestRegistration'.
- Homepage URL (Optional)**: An empty text field.
- Product logo URL (Optional)**: A text field containing 'http://www.example.com/logo.png'.
- Privacy policy URL (Optional)**: An empty text field.
- Terms of service URL (Optional)**: An empty text field.

At the bottom of the form, there is a 'Save' button (highlighted with a red box) and a 'Cancel' button. To the right of the form, there is a diagram showing a laptop and a smartphone with green checkmarks, indicating successful configuration. Below the diagram, there is explanatory text: 'The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.' and 'You must provide an email address and product name for OAuth to work.'

2. To create a new **Client ID** click **Create Credentials** under the **Credentials** tab.



3. Select **Web Applications** and configure Web-Origin and Redirect URL

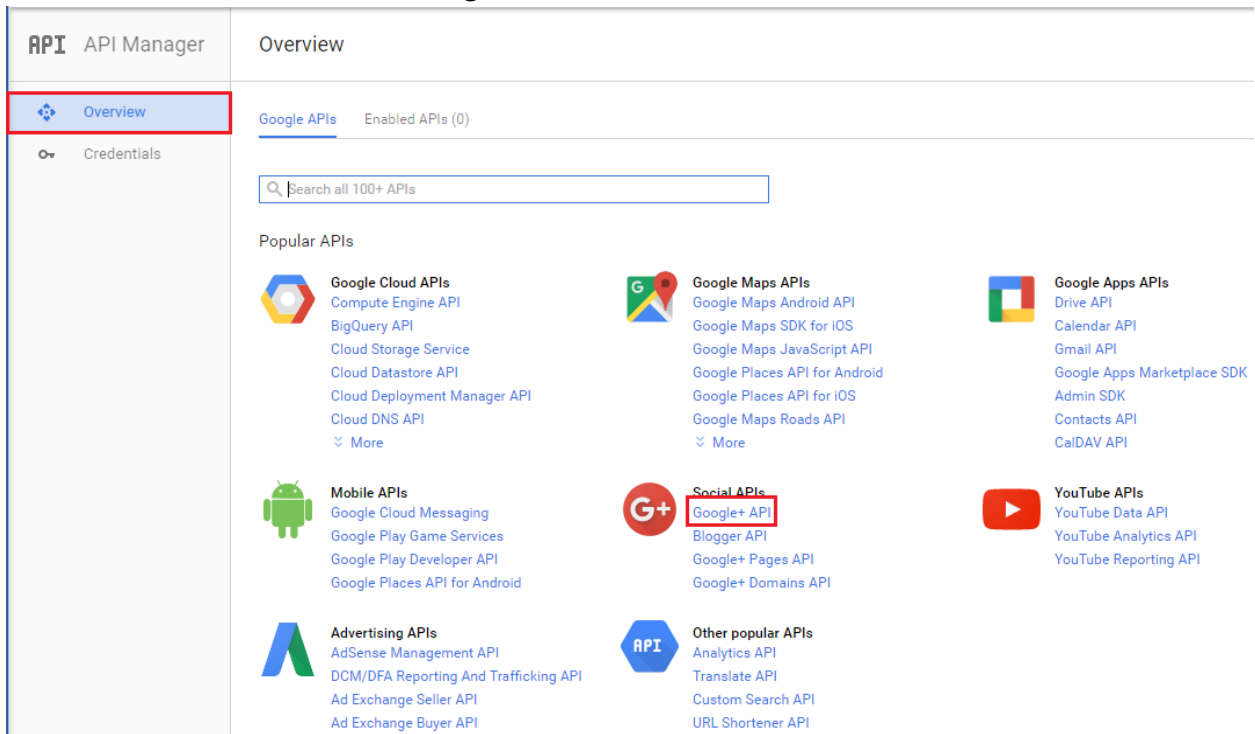
- Web-Origin: The Controller running the captive portal server eg. <http://quest.social.com:880>
- Redirect URL: Eg. www.google.com or any other external URL

The screenshot shows the 'API Manager' interface. On the left, the 'Credentials' tab is selected. The main area is titled 'Create client ID'. Under 'Application type', 'Web application' is selected. The 'Name' field contains 'Web client 1'. Under 'Restrictions', the 'Authorized JavaScript origins' field contains 'http://quest.social.com:880' and the 'Authorized redirect URIs' field contains 'http://www.google.com'. At the bottom, there are 'Create' and 'Cancel' buttons.

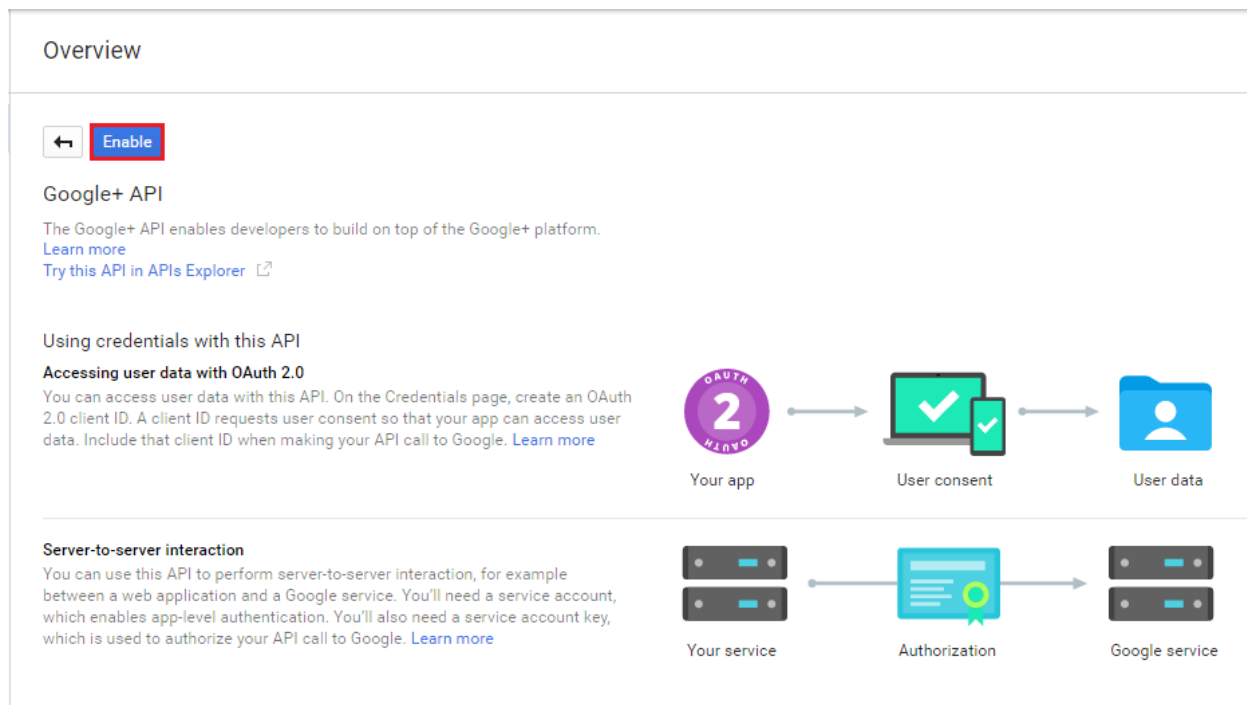
4. Save the **client ID** and **Client Secret** to be configured later on the WiNG5 Captive portal configuration.

The screenshot shows a dialog box titled 'OAuth client'. It contains two text fields. The first field is labeled 'Here is your client ID' and contains the value '53763355036-k6ilfkir8oos1675en9gqufcsblh5j02.apps.googleusercontent.com'. The second field is labeled 'Here is your client secret' and contains the value 'yZr7abed05fFCauAPpQ4rUZQ'. Below the fields is an 'OK' button.

5. Enable Google+ API so that the app will be able to obtain the user's profile information. On the **Overview** tab, click on **Google+ API**.

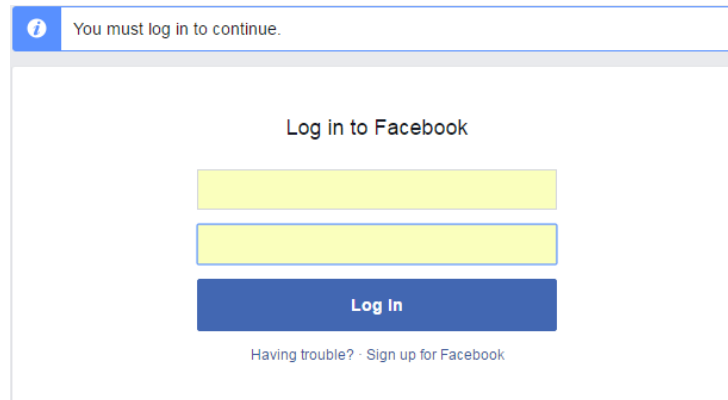


Click Enable to allow the WiNG application to obtain the user's profile information from Google+.

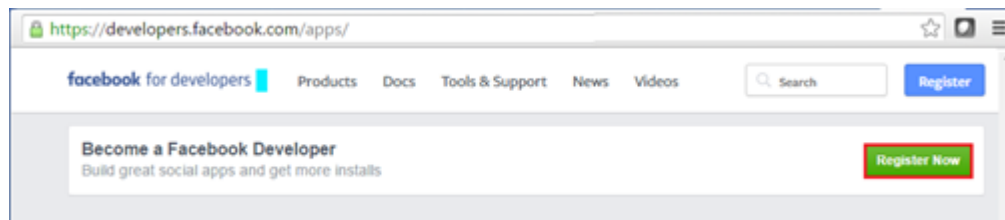


Facebook Registration:

6. For Facebook, login to the [Facebook App Dashboard](https://developers.facebook.com/apps/) (<https://developers.facebook.com/apps/>) with the facebook credentials.



7. Click **Register Now** to register the application.

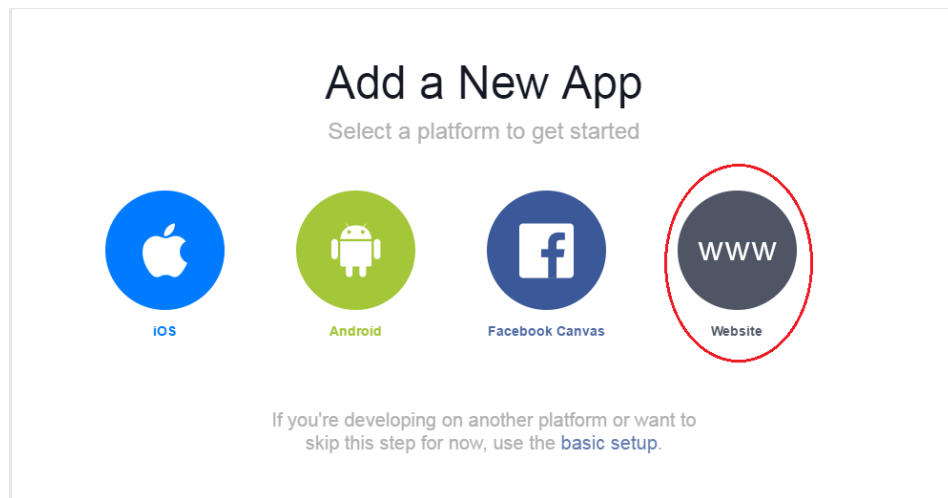


8. Accept the **Facebook Platform Policy & Privacy Policy**.

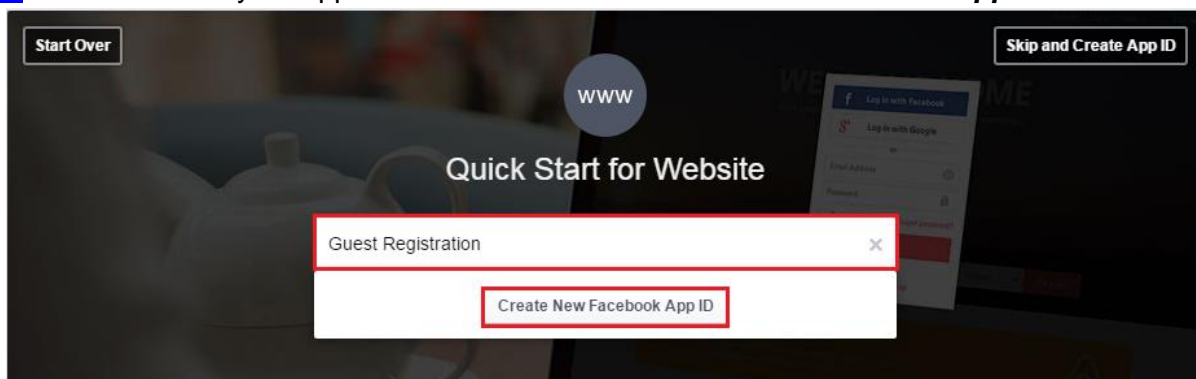


9. Verify your phone number on the next screen.

10. Add a new App. Click on **Website**.



11. Give a name to your application. Then click on **Create New Facebook App ID**.



12. Enter the **contact email** and select the **category**.

13. On the next page, you need to configure the “**Site URL**”

Tell us about your website

Site URL

Next

14. Next, select your application from **My Apps**.

facebook for developers


Products Docs Tools & Support News Videos

Search

My Apps

Search apps by title

+ Add a New App

 **Guest Registration** ○
App ID: 1673856739541753

Add a New App


Requests

Developer Settings

Company Settings

Log Out

15. Please note client ID and the secret for your application. This needs to be configured on WiNG5 captive portal configuration.

 **Guest Registration** ○

APP ID: 1673856739541753 | View Analytics

Tools & Support Docs

Dashboard

Settings

Roles


Alerts

App Review

PRODUCT SETTINGS

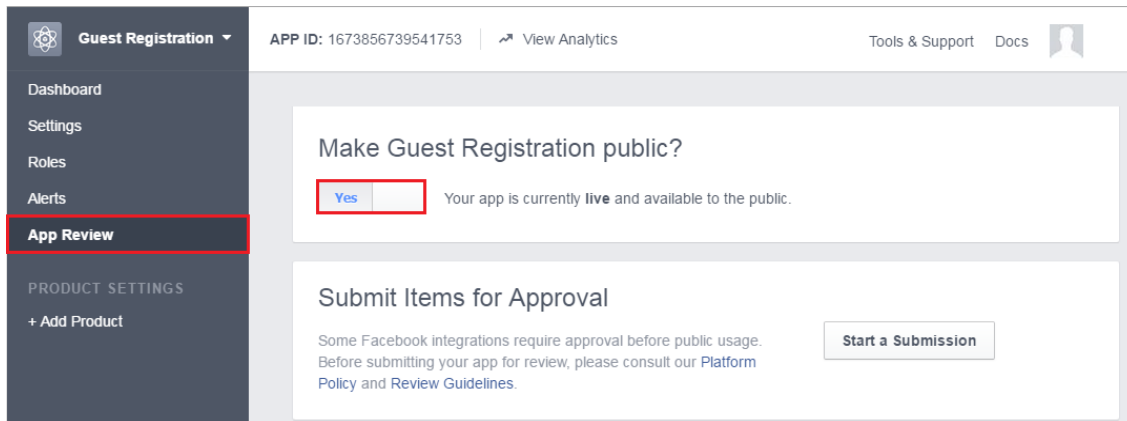
+ Add Product

Dashboard

 **Guest Registration** ○
This app is in development mode and can only be used by app admins, developers and testers (?)
API Version [?] App ID
v2.6 1673856739541753
App Secret
..... Show

Get Started with the Facebook SDK
Use our quick start guides to set up the Facebook SDK for your iOS or Android app, Canvas game or website.
Choose a Platform

[16.](#) Go to the App Review Tab and make the App available to everyone.



3.4 Captive Portal OAuth Setup (AP or Controller)

Once you have obtained the client IDs, you need to configure them under the captive portal policy as below along with setting the access-type to OAuth. The client IDs can be configured in any order.

captive-portal oauth-cp

server host guest.social.com

oauth

oauth client-id client-id Google xxxxxxxxxxxx.apps.googleusercontent.com Facebook

xxxxxxxxxxxxxxxxxxxx

use dns-whitelist oauth-dns-wl

bypass captive-portal-detection

webpage internal welcome use-external-success-url

webpage internal registration field city type text enable label "City" placeholder "Enter City"

webpage internal registration field name type text enable label "Full Name" placeholder "Enter First Name, Last Name"

webpage internal registration field zip type number enable label "Zip" placeholder "Zip"

webpage internal registration field via-sms type checkbox enable title "SMS Preferred"

webpage internal registration field mobile type number enable label "Mobile" placeholder "Mobile Number with Country code"

webpage internal registration field gender type dropdown-menu enable label "Gender" title "Gender"

webpage internal registration field optout type checkbox enable title "Do not remember and use my details"

webpage internal registration field member type text enable label "Loyalty/Member Card Number" placeholder "Enter Loyalty/Member Card Number"

webpage internal registration field dob type date enable label "Date of Birth" placeholder "MM/DD/YYYY"

webpage internal registration field street type text enable label "Address" placeholder "123 Any Street"

webpage internal registration field country type dropdown-menu enable label "Country" title "Enter State, Country"

webpage internal registration field age-range type dropdown-menu enable label "Age Range" title "Age Range"

webpage internal registration field email type e-address enable mandatory label "Email" placeholder "you@domain.com"

webpage internal registration field via-email type checkbox enable title "Email Preferred"

webpage internal registration field disclaimer type checkbox enable title "Use of this information is subject to our Terms and Conditions. By clicking Register, you agree to the terms of this Disclaimer"

3.5 Create a dns-whitelist

This step needs to be done in order for the social plugin buttons to be rendered on the client even before the Captive Portal authentication is done. This enables the client to access the following websites to retrieve the Javascript SDK required to display those colorful login buttons.

For Facebook:

```
dns-whitelist oauth-dns-wl
permit graph.facebook.com suffix
permit fbstatic-a.akamaihd.net
permit s-static.ak.facebook.com suffix
permit www.facebook.com suffix
permit m.facebook.com suffix
permit connect.facebook.net
permit facebook.com suffix
permit static.ak.facebook.com suffix
permit fbcdn.net suffix
```

For Google:

```
dns-whitelist oauth-dns-wl
permit accounts.google.com
permit apis.google.com
permit content.googleapis.com
permit oauth.googleusercontent.com
permit ssl.gstatic.com
```

3.6 Create WLAN:

```
wlan OAuth
ssid OAuth
vlan 20
bridging-mode local
encryption-type none
authentication-type none
use captive-portal oauth-cp
captive-portal-enforcement
```

4. Management

4.1 CLI

[no] oauth	
Context	captive-portal policy
Description	<p>This command is used to enable OAuth based authentication for wireless clients under the captive-portal policy.</p> <p>When not configured, OAuth authentication will be disabled.</p>
Parameters	None
Default	No oauth

[no] oauth client-id	
Context	captive-portal policy
Description	This command is used to configure the Google/Facebook client-id
Parameters	<p>Google WORD Client-id (eg. xxxxxxxxxxxx.apps.googleusercontent.com)</p> <p>WORD Client-id (15 digit facebook App ID)</p>
Default	No oauth client-id

5. Test Framework

5.1 Unit Testing

Case #	Title	Action	Pass/Fail Criteria
1	Configuration	<ol style="list-style-type: none"> 1. Go to Google and Facebook developers' sites (see section 5.3.1) to create an app. 2. On the WiNG device enable "oauth" and enter "oauth client-id" for a new captive-portal policy using the client id obtained from step 1. 	The captive-portal policy is created without error.
2	Login	<ol style="list-style-type: none"> 1. Configure a wlan to use the newly created captive-portal policy. 2. Associate a mobile client to the wlan. 3. Go to a site other than the ones listed in the DNS whitelist. 4. You should be redirected to a login page. 5. Choose one of the supported social media login methods. 6. Follow instruction to login. 	<ol style="list-style-type: none"> 1. Welcome page appears with the configured time remaining. 2. Confirm that the "login source" column in "show captive-portal sessions" shows the correct social media (i.e. facebook, google, etc) used for login.

Note: When using browsers such as Safari and Chrome, the user usually sees a page of previously visited sites when opening a new window or a new tab. Each of these icons is basically a saved URL supposedly linked to the actual site. Unfortunately due to the captive-portal authentication the URL of the first site visited after the client associates to the wlan is linked to the captive-portal's welcome page. Since it is cached by the browser, there is nothing we can do about it. To go around this issue, the user needs to enter that URL directly in the address bar.

6. References

1. <http://oauth.net/>
2. <http://tools.ietf.org/html/rfc6749>
3. <https://developers.facebook.com/docs/facebook-login/login-flow-for-web/#quickstart>
4. <https://developers.google.com/accounts/docs/OAuth2UserAgent>

