

# Windows Azure Active Directory

## Overview

Espresso Logic offers Window Azure Developers the ability to use Azure Active Directory (WAAD) as part of the authentication service for Espresso applications. This service delegates the responsibility to WAAD for user identity and authentication. This means that the login service itself can also be handled by WAAD for multi-factor authentication and identity protection. The authentication service will detect the existence of an existing WAAD user and renegotiate a new access token if necessary. In turn, Espresso Logic will issue a temporary API Key to the requesting application and handle all the internal synchronization.

## Getting Started

The actual authentication service can be found on [GitHub](#). The first step is to build the Azure Java Library (adal4j-0.0.1-SNAPSHOT.jar) and upload this to your Espresso Logic Project (under Authentication Service and Library) along with the JavaScript [WAADSecurityProvider.js](#). Once these parts are in place you will be able to add your the new authentication service for Windows Azure Active Directory. (Authentication Provider tab on the Projects Menu)

The screenshot shows the 'Authentication Providers' tab in the Espresso Logic interface. On the left, a list of providers includes 'Built-in authentication (dbtest\_admin - dbtest\_admin)' and 'Windows Azure Authentication Provider'. The right panel, titled 'Selected Provider', shows the configuration for the 'Windows Azure Authentication Provider'. The 'Name' field is filled with 'Windows Azure Authentication Provider'. The 'Authentication Method' is set to 'JavaScript Auth Provider'. The 'Name for Create Function' field is highlighted with a red box and contains the text 'waadSecurityProvider'. Below this, a message states 'Cannot create authprovider - Missing value for Name for Create Function'. The 'Comments' field contains the text 'This is the JavaScript name in the file [WAADSecurityProvider.js](#)'. At the bottom, there are 'Save' and 'Revert' buttons.

Authentication Providers	Selected Provider
<div>⊕ Add ⊖ Delete</div> <div>Built-in authentication (dbtest_admin - dbtest_admin)</div> <div>Windows Azure Authentication Provider</div>	<div>Name</div> <div>Windows Azure Authentication Provider</div> <div>Authentication Method</div> <div>JavaScript Auth Provider</div> <div>Name for Create Function:</div> <div>waadSecurityProvider</div> <div>Cannot create authprovider - Missing value for Name for Create Function</div> <div>Comments</div> <div>This is the JavaScript name in the file <a href="#">WAADSecurityProvider.js</a></div> <div>Save Revert</div>

Name for Create Function:	<input type="text" value="waadSecurityProvider"/>
Authority	<input type="text" value="https://login.windows.net"/>
Tenant Name	<input type="text" value="tenantName"/>
Client ID	<input type="text" value="myclientid"/>
Client Secret Key	<input type="text" value="myclientsecret"/>
API Version	<input type="text" value="api-version=2013-11-08"/>
API Key Lifetime (Minutes)	<input type="text" value="60"/>
Comments	<input type="text" value="This is the JavaScript name in the file WAADSecurityProvider.js"/>

Once the correct create function name is entered - the new internal fields will appear (above) which are used to connect to your WAAD account. Remember, these are server side values and will not be accessible to anyone without admin access to the Logic Design Studio. Now go back to the Details tab and select the new authentication service.

Details

Settings

Libraries

Authentication Providers

Project name: ?

MyEspressoProject

URL fragment: ?

project1

Authentication provider: ?

Windows Azure Authentication Provider

## Azure AD Setup

Once you have created your Azure account, you will need to setup users and defined groups (below). These group names must match the role names in Espresso Logic to define access rights to your REST API endpoints. Only the matching roles will be used for security access.

Microsoft Azure

tylrm007@gmail.com

Default Directory

default directory

USERS

GROUPS

APPLICATIONS

DOMAINS

DIRECTORY INTEGRATION

CONFIGURE

REPORTS

LICENSES

NAME	DESCRIPTION	SOURCED FROM
Developer	Developer	Windows Azure Active Directory
Management		Windows Azure Active Directory
QA		Windows Azure Active Directory

ADD GROUP

DELETE

Roles

+ Add

+ Delete

Full access

Read only

Authorized per region

Details

Globals

Permissions

REST End Points

Role name:

Full access

Default Database Table Access:

Read ☒

Insert ☒

Update ☒

Delete ☒

Automatic Enabling of REST EndPoints

Leave unchecked to individually select the reachable End Points.

All Tables ☒

All Views ☒

All Resources ☒

All Procedures ☒

All Meta Tables ☒

Description:

## Testing

If everything is working correctly - when you attempt to logon to your account you should be directed to a Microsoft logon window using a WAAD account- you will need to pass a redirect url as part of the authentication to let WAAD know where to send the successful logon.

See Espresso Logic documentation for [details](#) on authentication and [API Keys](#).