



Azure Prerequisites

v 1.0

[22 March 2018]

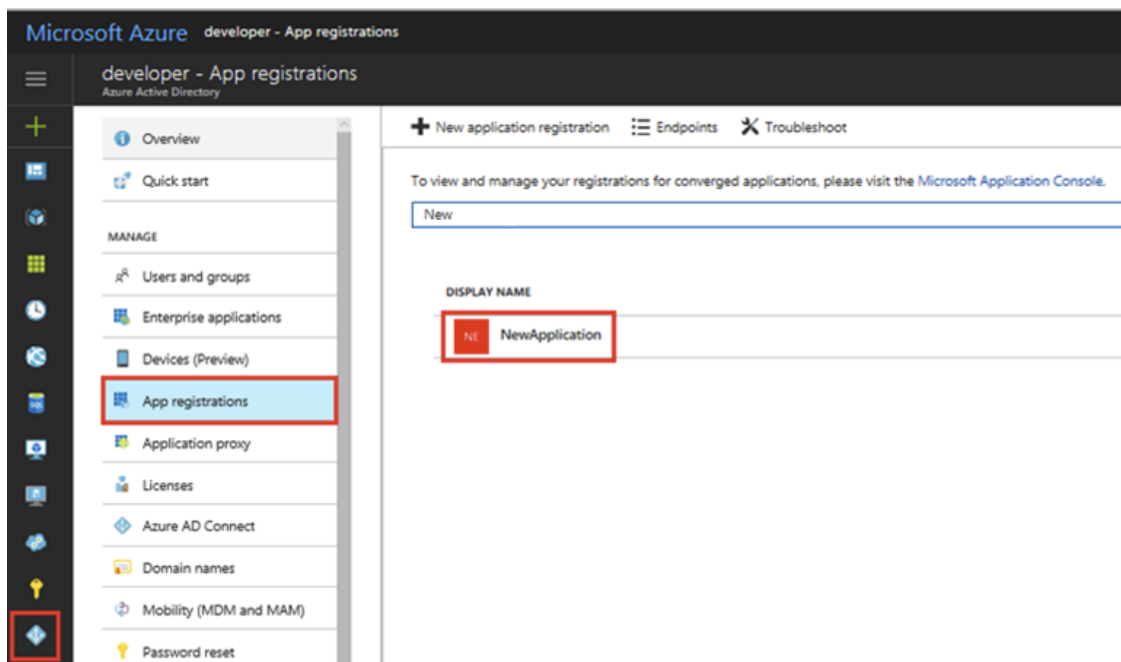
Confidential, Nubeva, Inc.

Author: Erik Freeland
Title: Dir of Customer Engineering
Email: erik@nubeva.com

Setup Azure Prerequisites

1. Now that the nubevapoc environment installed into the Azure environment, it is necessary to link the Manager with the AD accounts used in Azure. This is critical to the integration with the Azure authentication environment.
2. **Create a service principal.** This can be completed in several ways. One method using the Azure Portal GUI can be found below in Appendix A.
 - a. A service principal can also be created with the following CLI command:

```
az ad sp create-for-rbac
```
3. **Adding Permissions.** Nubeva's product uses Microsoft's OAuth authentication system in order to authenticate and authorize users. In order to access the authentication system, you must grant your application certain permissions. This process should be done through Azure Portal.
 - a. In the left-hand navigation pane, click **Azure Active Directory** > **App registrations** > then click on the application that you want to configure.



- b. In the **Settings** blade, click on **Required permissions** under "API ACCESS"

NewApplication
Registered app

Settings Manifest Delete

Essentials ^

Display name
NewApplication

Application type
Web app / API

Home page
http://localhost:8000

Application ID
1ca88f0c-0c6d-42cf-a215-c28c67da8833

Object ID
e274d7b2-d85c-44ee-b66f-5400d8c68012

Managed application in local directory
NewApplication

All settings →

Filter settings

GENERAL

Properties >

Reply URLs >

Owners >

API ACCESS

Required permissions >

Keys >

TROUBLESHOOTING + SUPPORT

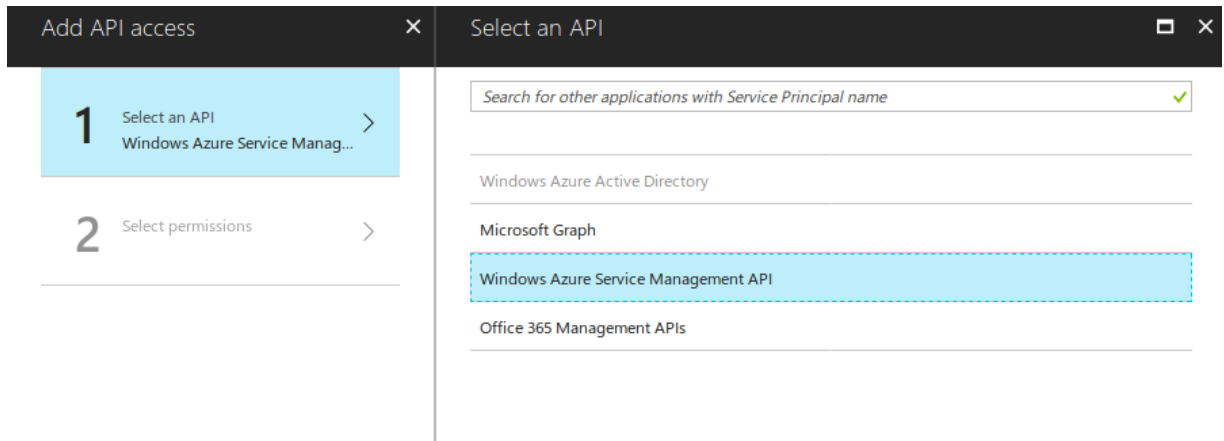
Troubleshoot >

New support request >

+ Add Grant Permissions

API	APPLICATION PERM...	DELEGATED PERMIS...
Windows Azure Active Directory	0	1

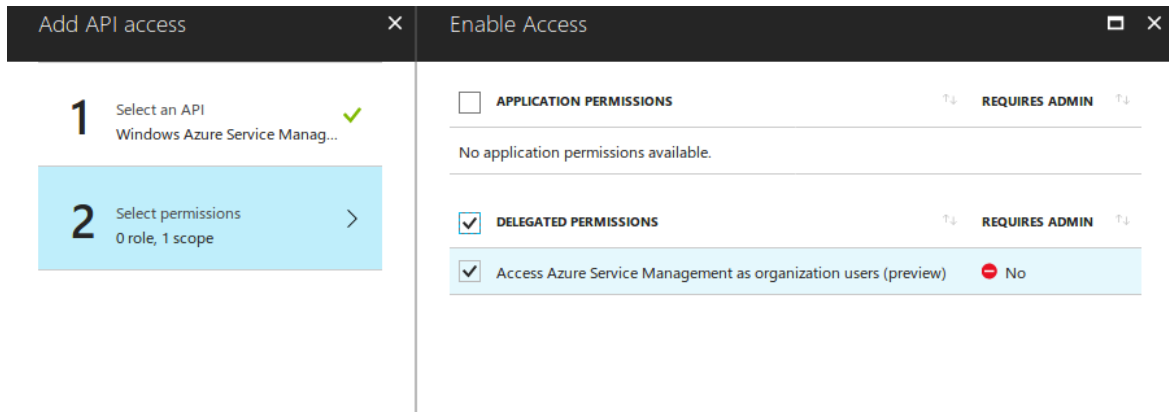
- c. Click **Add** and select **Windows Azure Service Management API**. Then press **Select** at the bottom of the blade.



The screenshot shows the 'Add API access' blade with two steps: '1 Select an API' and '2 Select permissions'. The first step is active, showing a search bar and a list of APIs. The 'Windows Azure Service Management API' is highlighted in blue.

Search for other applications with Service Principal name
Windows Azure Active Directory
Microsoft Graph
Windows Azure Service Management API
Office 365 Management APIs










- d. Select **Access Azure Service Management as organization users (preview)** under "DELEGATED PERMISSIONS". Then press **Select** and **Done** once finished.



The screenshot shows the 'Enable Access' blade with two steps: '1 Select an API' and '2 Select permissions'. The second step is active, showing a list of permissions. The 'Access Azure Service Management as organization users (preview)' permission is selected under the 'DELEGATED PERMISSIONS' section.

APPLICATION PERMISSIONS	REQUIRES ADMIN
No application permissions available.	
DELEGATED PERMISSIONS	REQUIRES ADMIN
<input checked="" type="checkbox"/> Access Azure Service Management as organization users (preview)	<input checked="" type="checkbox"/> No

- e. You will be brought back to the application's **Settings** and **Required permissions** blades. Under **Required Permissions**, Add **Windows Azure Active Directory** and select the following under "DELEGATED PERMISSIONS":
- **Read all users' basic profiles**
 - **Sign in and read user profile**

<input type="checkbox"/> DELEGATED PERMISSIONS	REQUIRES ADMIN
Access the directory as the signed-in user	 No
Read directory data	 Yes
Read and write directory data	 Yes
Read and write all groups	 Yes
Read all groups	 Yes
Read all users' full profiles	 Yes
<input checked="" type="checkbox"/> Read all users' basic profiles	 No
<input checked="" type="checkbox"/> Sign in and read user profile	 No
Read hidden memberships	 Yes

- f. Click **Save** at the top left of the blade. In the end, the **Required permissions** blade should look like the following:

GENERAL

Properties

Reply URLs

Owners

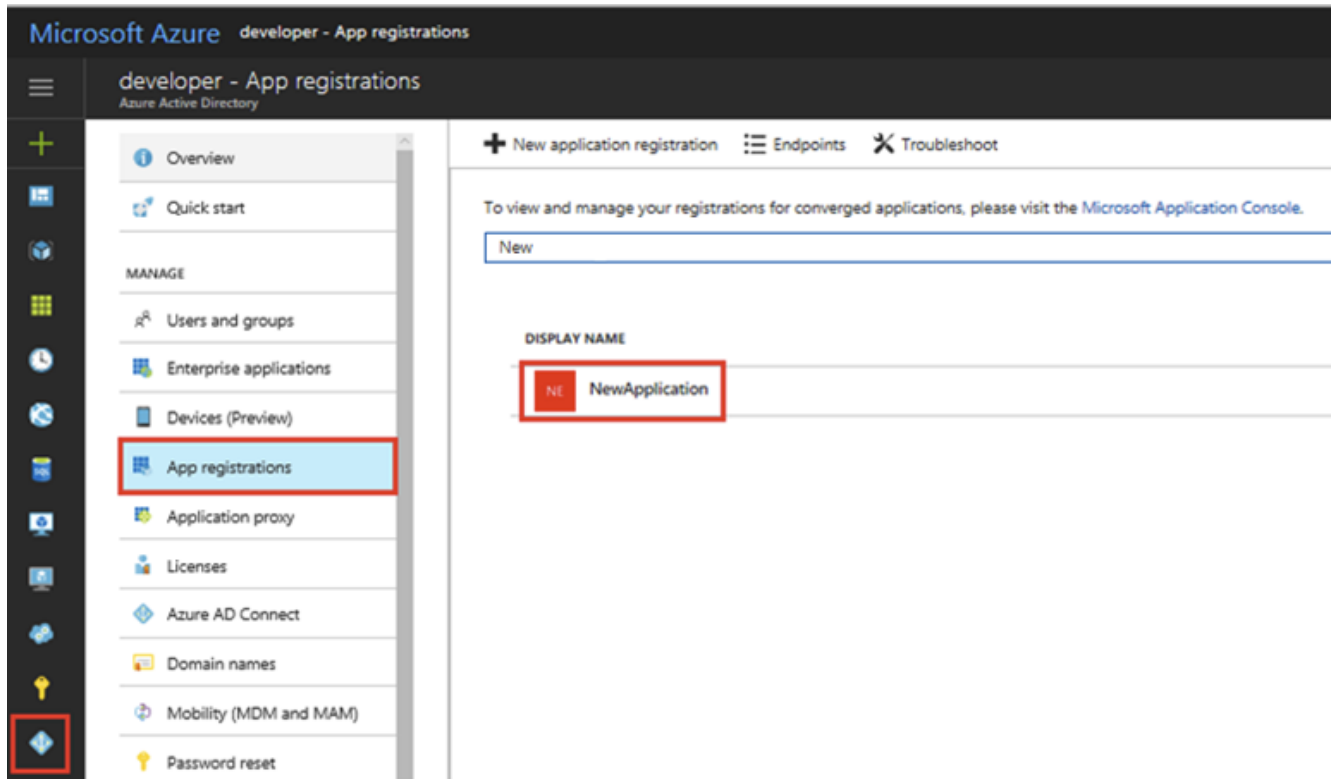
API ACCESS

+ Add

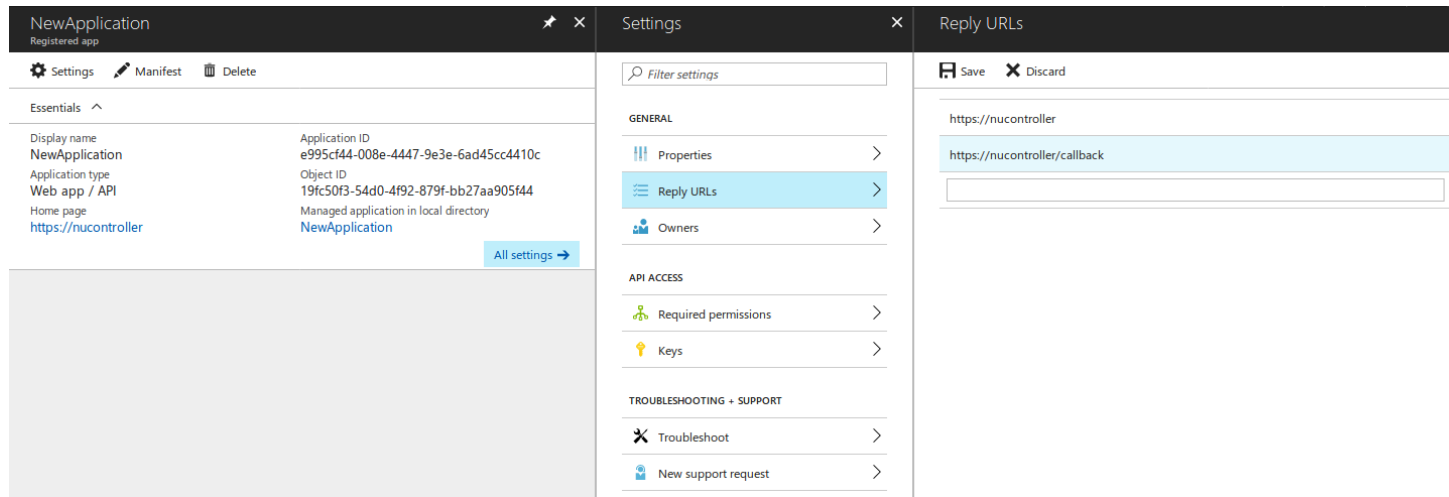
Grant Permissions

API	APPLICATION PERMI...	DELEGATED PERMIS...
Windows Azure Active Directory	0	2
Windows Azure Service Management API	0	1

4. **Updating Reply URL.** The Azure Active Directory Reply URL must be updated so that an authentication token can be received after login.
 - a. In the left-hand navigation pane, click **Azure Active Directory > App registrations** > then click on the application that you want to configure.



- b. In the **Settings** blade, click on **Reply URLs** under "GENERAL", and add the following reply url: <https://nucontroller/callback>.



The screenshot displays the Azure portal interface for a 'NewApplication' (Registered app). The 'Settings' blade is active, showing the 'GENERAL' section with 'Reply URLs' selected. The 'Reply URLs' pane on the right shows a list of URLs, including 'https://nucontroller' and 'https://nucontroller/callback'. The 'Save' button is visible at the top of the 'Reply URLs' pane.

NewApplication	
Registered app	
Settings	Manifest
Delete	
Essentials	
Display name	Application ID
NewApplication	e995cf44-008e-4447-9e3e-6ad45cc4410c
Application type	Object ID
Web app / API	19fc50f3-54d0-4f92-879f-bb27aa905f44
Home page	Managed application in local directory
https://nucontroller	NewApplication
All settings	

Settings	
Filter settings	
GENERAL	
Properties	>
Reply URLs	>
Owners	>
API ACCESS	
Required permissions	>
Keys	>
TROUBLESHOOTING + SUPPORT	
Troubleshoot	>
New support request	>

Reply URLs	
Save	
Discard	
https://nucontroller	
https://nucontroller/callback	
<input type="text"/>	

- c. Click **Save**.

5. **Retrieving Credentials.** Nubeva's Controller requires the Subscription ID, Tenant ID, Password, and Application ID. All of this information can be found in the output of the CLI command above in Step 4a plus the command: `az account show`. The instructions for obtaining this information via the Azure Portal is detailed below.
 - a. To retrieve the Subscription ID, click on **Subscriptions** on the left hand navigation pane of the portal. Then copy the **Subscription ID** that you have linked your application to.

Subscriptions

stevenubedgegmail (Default Directory)

Add

My role ⓘ

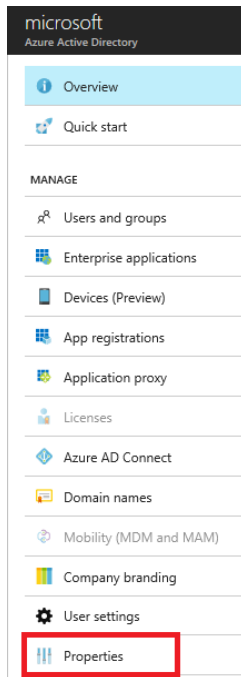
7 selected

Apply

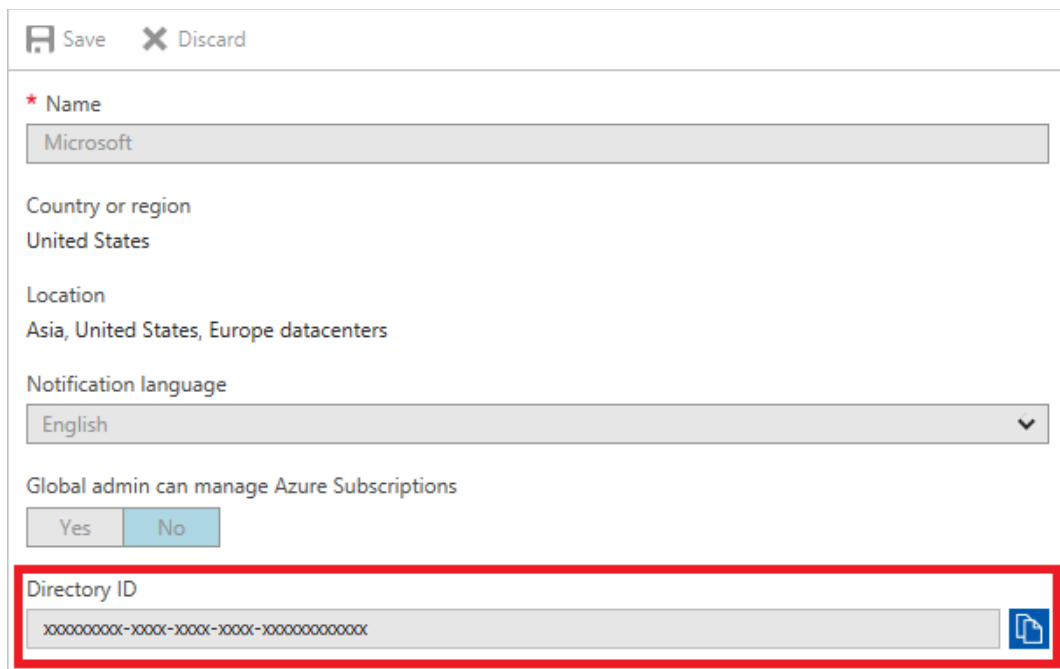
Search to filter items...

SUBSCRIPTION	↑↓ SUBSCRIPTION ID
--------------	--------------------

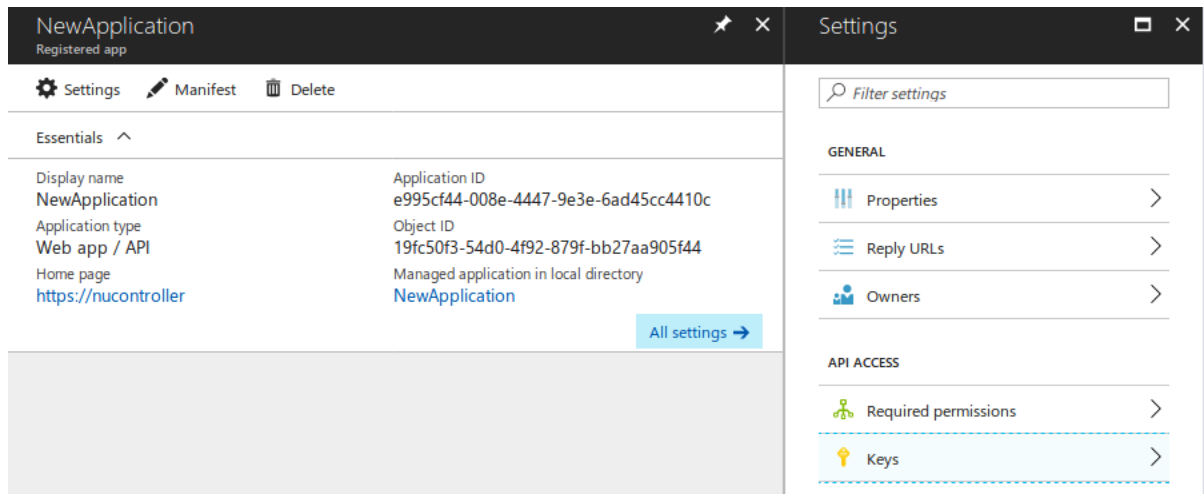
- The Tenant ID is under Directory ID. Go to **Azure Active Directory** on the left hand navigation pane of the portal. Then click **Properties**.



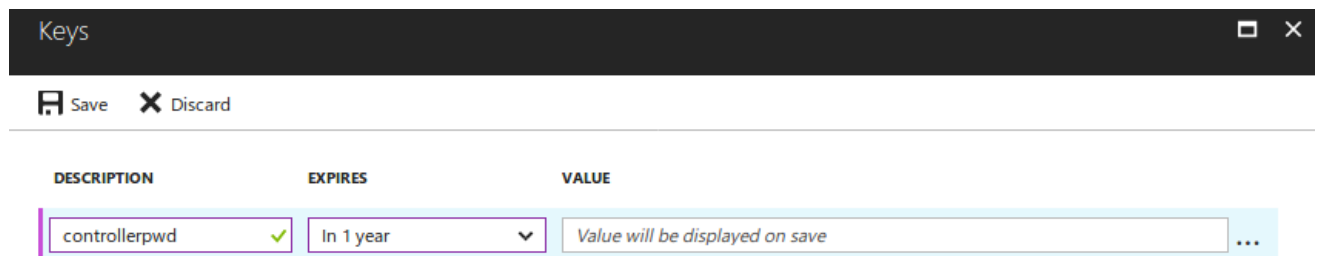
- Copy the **Directory ID** since this is your Tenant ID.

The image shows the 'Properties' page for an Azure Active Directory tenant. At the top, there are 'Save' and 'Discard' buttons. The page contains several fields: 'Name' (with a red asterisk) containing 'Microsoft', 'Country or region' set to 'United States', 'Location' set to 'Asia, United States, Europe datacenters', and 'Notification language' set to 'English'. Below these is a toggle for 'Global admin can manage Azure Subscriptions' with 'Yes' and 'No' buttons. At the bottom, the 'Directory ID' field is highlighted with a red rectangular border. It contains a masked value 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx' and a copy icon to its right.

8. The password is a **Keys** Value under the Azure Active Directory application's keys. Go to **Azure Active Directory** on the left hand navigation pane, click on **App registrations**, and click on your desired application.
9. The password can be created under **Keys** in the "API ACCESS" section:



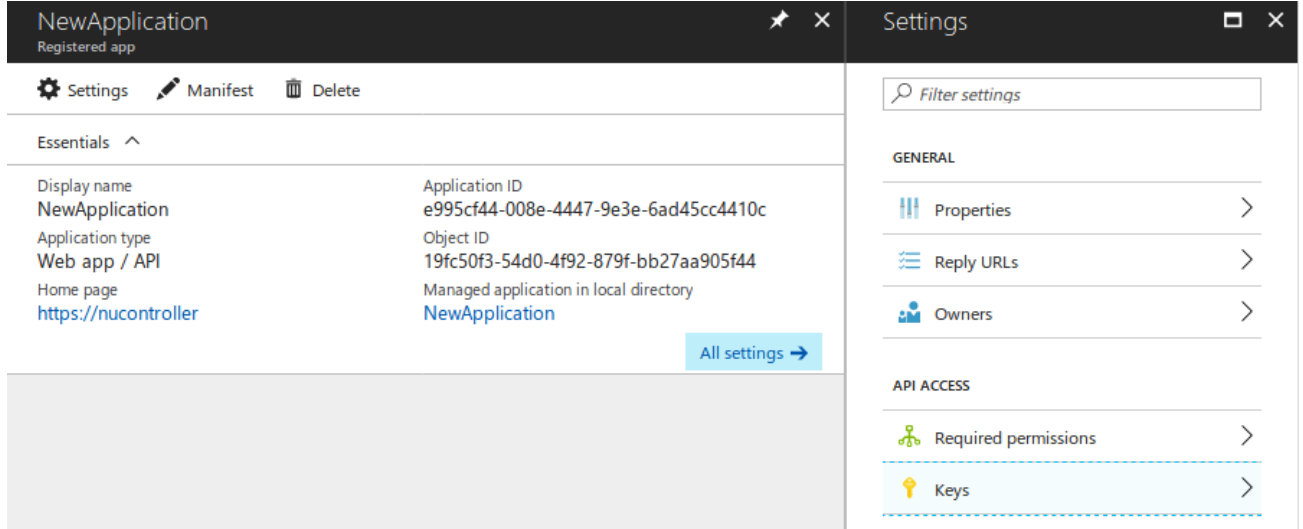
10. Fill in the fields under the **Keys** blade and click **Save**. Make sure to keep record of the **Value** that is displayed right after you click **Save**.
11. **IMPORTANT:** You will need that key value when sending your credentials to the Controller API endpoint.



DESCRIPTION	EXPIRES	VALUE
controllerpwd	In 1 year	Value will be displayed on save

12. For the **Application ID** . Go to **Azure Active Directory** on the left hand navigation pane, click on **App registrations**, and click on your desired application.

13. Under the application's pane, you will see **Application ID** field. Copy this field and save it for your credentials.



The screenshot shows the Azure Active Directory application settings page for an application named 'NewApplication'. The left pane displays the 'Essentials' section with the following details:

Property	Value
Display name	NewApplication
Application type	Web app / API
Home page	https://nucontroller
Application ID	e995cf44-008e-4447-9e3e-6ad45cc4410c
Object ID	19fc50f3-54d0-4f92-879f-bb27aa905f44
Managed application in local directory	NewApplication

The right pane shows the 'Settings' section with a search bar and two main categories: 'GENERAL' and 'API ACCESS'. Under 'GENERAL', there are links for 'Properties', 'Reply URLs', and 'Owners'. Under 'API ACCESS', there are links for 'Required permissions' and 'Keys'.

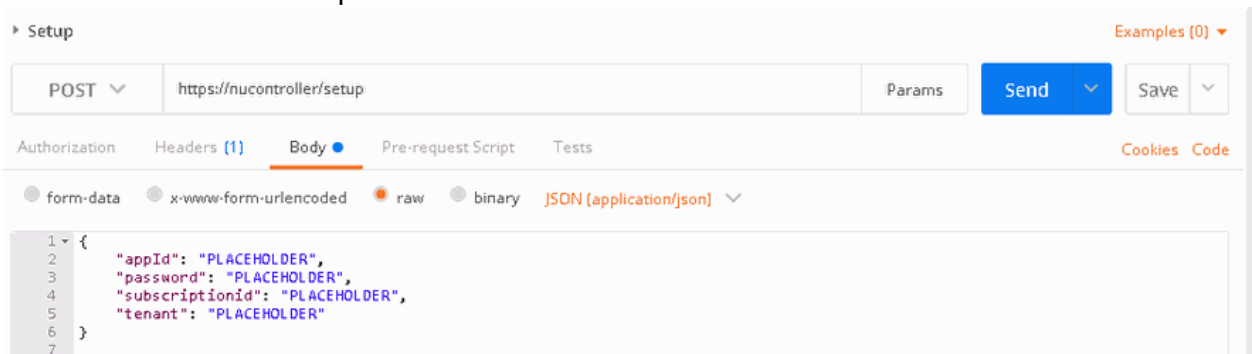
One-Time Manager Configuration

1. The final step in the process is to link the manager to the Azure service principal. All Manager configuration will use Postman to make the appropriate RestAPI calls. Any program or script that has the ability to make RestAPI calls can be used.

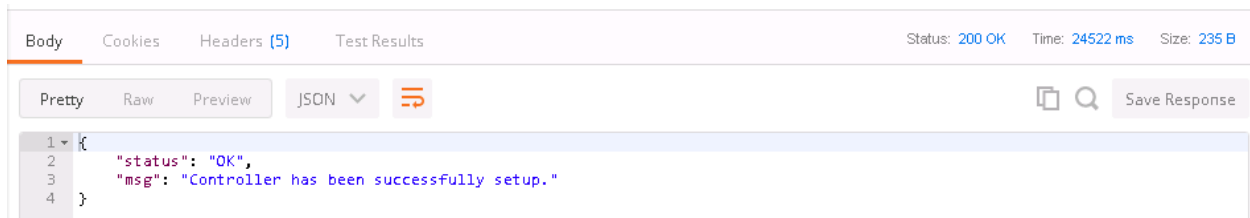
Note: The controller can only be accessed from inside the Vnet/VPC by default. This can be changed using various cloud connectivity options. The simple solution is to launch a windows VM and load postman on the VM. This is part of the Nubeva POC environment.

Note: If you need additional assistance in setting up postman with the various Nubeva collections and environmental parameters, please refer to the Postman Setup for Nubeva.

2. In postman, locate the POST command with the following URL: <https://{{apiUrl}}/setup>
 - a. The {{apiUrl}} is an environmental variable. It is usually “nucontroller” in most environments. The value can be any IP address or FQDN.
 - b. This is the command which will associate this controller with the service principal created earlier.
3. Then click on the **Body** tab, and include your Azure service principal credentials created above. There is also example data in the Nubeva Postman collection.



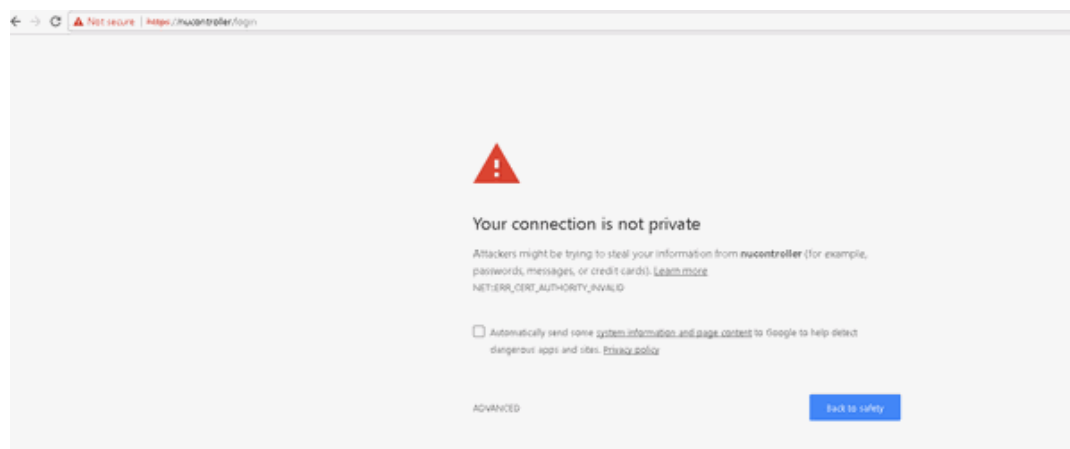
- Then click **Send**. This call should take about 30 seconds. At the end, you should see a similar result as below.



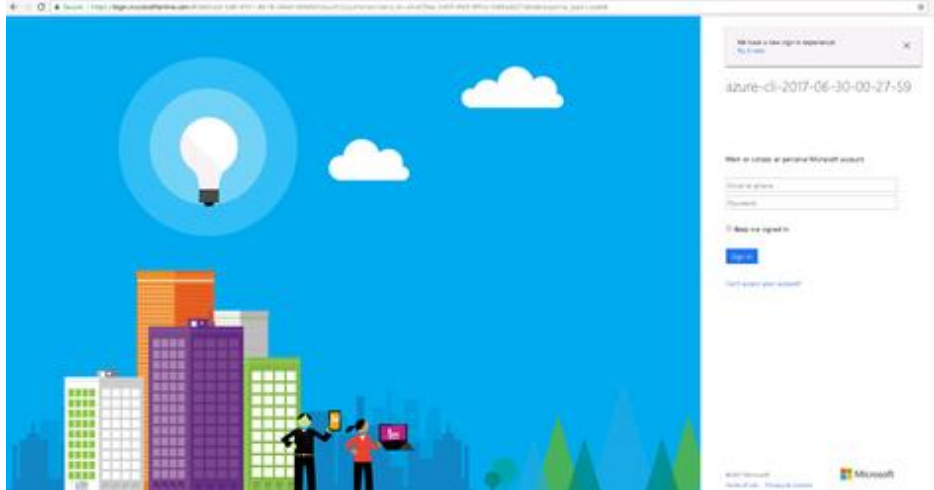
NOTE: If you do not have permissions to access Microsoft's Graph API endpoints, then your reply url cannot be confirmed. So, you will receive a message to check that the reply url has been set through the Azure Portal. Even though you may receive this message, your controller has been successfully setup. Please make sure that the reply URL is included in your Azure Active Directory Application's list of reply URLs.

```
{
  "status": "OK",
  "msg": "Controller has been successfully setup.",
  "unconfirmedValidations":
    ["ReplyURLUnconfirmed: https://nucontroller/callback. Possibly
    due to insufficient privileges or requirement was not met. Verify
    through Azure Active Directory application's reply-urls list."]
}
```

- The last step in the setup is validating an authentication token. This process will be repeated every time a new token is required. As with most RestAPI environments, these tokens last for 60 minutes.
- To begin the authentication process, access <https://nucontroller/login>, via a web browser.



- Accept the SSL certification errors and proceed to the destination URL. By proceeding to the webpage, you will be redirected to Microsoft's Account login page.



- Please enter in your credentials. If valid, you will be redirected to the /callback endpoint, which will display your "auth_code." This "auth_code" maps directly to the "token" variable inside the "Nubeva Controller" environment in Postman. Copy this auth_code into the token field.



- With the auth_code/token set, the Nubeva Controller is fully installed and configured. In the next section, we will document how to install a StratusEdge node for tapping.