Office365Mon and AzureServiceMon Developer API

Office365Mon and AzureServiceMon provide a set of services for retrieving report data, and soon for managing subscriptions. This document describes how you can create an application to programmatically access report data via the Office365Mon and/or AzureServiceMon API. You can see a complete list of the report data feeds as well as download data for each on our web site at https://www.office365mon.com/reports/reportdata and https://azureservicemon.com/reports/reportdata. You can use that to view the data to better understand its format so you know how to process it.

API Endpoints

All of the Office365Mon API endpoints are described below for the default data center location, which most customers will be in. However, if your subscription is in an Office365Mon Private deployment, then you'll need to replace the default root URL with the one for your deployment.

All of the AzureServiceMon API endpoints are for the default data center location, which is *https://azureservicemon.com*.

Reporting APIs

Office365Mon and AzureServiceMon provide a set of REST endpoints that you can use to retrieve reporting data for your subscriptions. The **Office365Mon** endpoints are as follows:

- Outages: **https://www.office365mon.com/report/data/outages** - this endpoint provides all of the data used to create all of the various outage reports
- Health Pings: **https://www.office365mon.com/report/data/healthpings** - use the Health Pings endpoint to get a report of all of the health probes that have been issued against your resources. Please note that this data is cleaned up once a day, so you will not have more than one day's worth of probes in this report.
- Monthly Rollup: **https://www.office365mon.com/report/data/monthlyrollup** - this data contains all of the rollup data that are used in the monthly summary reports. That includes information such as how many probes were issued, what the average response time was, etc.
- Monthly Rollup: **https://www.office365mon.com/report/data/availability** - this data gives you the availability of your resources for each month that we have been monitoring Office 365 for you.
- Resource Status: **https://www.office365mon.com/report/data/resourcestatus** - the Resource Status data gives you real time information about each of your monitored resources in terms of whether they are currently in an outage or not.
- Service Health Status: **https://www.office365mon.com/api/reportdata/servicehealthstats** - overall health statistics for all Office365Mon customers is contained in the Service Health Status data.
- Service Outage Status: **https://www.office365mon.com/api/reportdata/serviceoutagestats** - overall outage statistics for all Office365Mon customers is contained in the Service Outage Status data.
- Email Transport Performance: **https://www.office365mon.com/api/reportdata/serviceemailtransportstats** - the Overall Email Transport Performance report shows you the average inbound and outbound email transport delivery times across all Office365Mon customers.
- Recent Email Transport Probes: **https://www.office365mon.com/api/reportdata/recentemailtransport** - this report shows you the recent email transport probes that have been sent and what the performance has been for them - both inbound and outbound. The number of seconds starts immediately when the email is sent, and stops when the email is received.
- Monthly Email Transport Performance: **https://www.office365mon.com/report/data/monthlyemailtransportperf** - shows you average response time, in seconds, for emails to get routed in and out of your Office 365 email transport infrastructure. The number of seconds starts immediately when the email is sent, and stops when the email is received.
- Email Transport Issues: **https://www.office365mon.com/report/data/emailtransportissues** - shows you instances in which latent email transport issues caused a notification to be sent. A notification is sent when it takes longer to deliver a message than you allowed, according to values you have configured in the Configure Email Transport Monitoring page.
- Recent Email Transport Errors: **https://www.office365mon.com/report/data/recentemailtransporterrors** - shows you any of the 100 most recent errors that may have been encountered when monitoring the email transport for your tenant.

- Monthly Email Transport Errors: **https://www.office365mon.com/report/data/monthlyemailtransporterrors** - shows you the how many times a particular error has occurred each month when monitoring the email transport for your tenant.
- Recent Query Latency: **https://www.office365mon.com/report/data/querylatency** - shows you the latency experienced over the last hour or so when executing the query that you've defined in the Configure Search Monitoring page. It represents the time, in seconds, that it took to get search results returned.
- Recent Crawl Latency: **https://www.office365mon.com/report/data/crawllatency** - shows you the latency experienced over the last several hours that it has taken for Office 365 to index new content that was added to the site being monitored by Office365Mon. It represents the time, in minutes, that it took to get content crawled and search results returned for newly added content.
- Daily Query Latency: **https://www.office365mon.com/report/data/dailyquerylatency** - shows you the average latency experienced each day when executing the query you've defined in the Configure Search Monitoring page. It represents the time, in seconds, that it takes to get search results returned.
- Daily Crawl Latency: **https://www.office365mon.com/report/data/dailycrawllatency** - shows you the average latency experienced each day that it has taken for Office 365 to index new content that was added to the site being monitored by Office365Mon. It represents the time, in minutes, that it took to get content crawled and search results returned for newly added content.
- Monthly Query Latency: **https://www.office365mon.com/report/data/monthlyquerylatency** - shows you the average latency experienced each month when executing the query you've defined in the Configure Search Monitoring page. It represents the time, in seconds, that it takes to get search results returned.
- Monthly Crawl Latency: **https://www.office365mon.com/report/data/monthlycrawllatency** - shows you the average latency experienced each month that it has taken for Office 365 to index new content that was added to the site being monitored by Office365Mon. It represents the time, in minutes, that it took to get content crawled and search results returned for newly added content.
- Overall Search Performance: **https://www.office365mon.com/api/reportdata/servicesearchstats** - shows you the average query latency and average crawl time across all Office365Mon customers.
- Performance by Geography: **https://www.office365mon.com/api/reportdata/geohealth** - gives you a geographical heat map of the performance for all the locations you are monitoring using our Distributed Probes and Diagnostics service.
- Outages by Geography: **https://www.office365mon.com/api/reportdata/geooutages** - gives you a geographical heat map of outages for all the locations you are monitoring using our Distributed Probes and Diagnostics service.
- Latest Performance: **https://www.office365mon.com/api/reportdata/recentgeopings** - shows you the latest performance from each location where you've installed the Distributed Probes and Diagnostics service.

- Geographical Service Performance: **https://www.office365mon.com/api/reportdata/servicegeohealthstats** - overall performance by geography for all Office365Mon customers is contained in the Geo Service Performance data.
- Geographical Service Outages: **https://www.office365mon.com/api/reportdata/servicegeooutagestats** - overall outages by geography for all Office365Mon customers is contained in the Service Outages data.
- Remote Agents: **https://www.office365mon.com/api/reportdata/dpdhostinfo** - shows you all of the hosts that are running the Distributed Probes and Diagnostics agent for this subscription.
- Current Threat Intel Details: **https://www.office365mon.com/api/reportdata/threatdetails** - shows you the individual threat intelligence reports that were detected during the current month. You can see all of the individual message details that contained malware that were attempted to deliver in your organization.
- Current Targeted Users: **https://www.office365mon.com/api/reportdata/threattargetedusers** - shows you the users that have been targeted by malware the most during the current month.
- Current Malware Trends: **https://www.office365mon.com/api/reportdata/currentthreattrends** - shows you a count of the different malware types that were received each day during the current month.
- Other Current Threat Counts: **https://www.office365mon.com/api/reportdata/threatmonthlycounts** - shows you a count of different data points from malware data for the current month.
- Other SharePoint Threats: **https://www.office365mon.com/api/reportdata/sharepointthreatmonthlycounts** - shows you a count of different data points from SharePoint and OneDrive malware data for the current month.
- Current Detected Malwares: **https://www.office365mon.com/api/reportdata/detectedmalwares** - shows you a count of the different malware types have been detected at your organization.
- 60 Day Malware Trends: **https://www.office365mon.com/api/reportdata/rollingthreattrends** - shows you a count of the different malware types that were received each day during the previous and current month.
- Historical Malware Details: **https://www.office365mon.com/api/reportdata/threathistoricalcounts** - shows you a count of different data points from malware data from the history of monitoring this tenant.
- Log Shipped Counts: **https://www.office365mon.com/api/reportdata/logshippedcounts** - contains historical statistical information for Office 365 logs that have been shipped to Office365Mon.Com.
- Log Shipped Trends: **https://www.office365mon.com/api/reportdata/logshippedtrends** - contains operational trending information for Office 365 logs that have been shipped to Office365Mon.Com.
- SharePoint Health Scores: there are three slices of data for SharePoint Health Scores – recent, daily and monthly. They contain health scores and request duration times for SharePoint Online and OneDrive for Business sites. These scores reflect the health of the SharePoint farm for your tenant, and how long the internal execution time is taking for health probe requests. They are found at the following Urls respectively:

    - **https://www.office365mon.com/api/reportdata/healthscoresrecent**
    - **https://www.office365mon.com/api/reportdata/healthscoresdaily**
    - **https://www.office365mon.com/api/reportdata/healthscoresmonthly**
- SharePoint Network Time: there are three slices of data for SharePoint Health Scores – recent, daily and monthly. They contain network transportation time for SharePoint Online and OneDrive for Business across all of the locations you are issuing health probes. That includes both our cloud-based probes, as well as any Distributed Probe agents you have running. They are found at the following Urls respectively:
    - **https://www.office365mon.com/api/reportdata/networkperfrecent**
    - **https://www.office365mon.com/api/reportdata/networkperfdaily**
    - **https://www.office365mon.com/api/reportdata/networkperfmonthly**
- Host Network Performance - Time of Day: **https://www.office365mon.com/api/reportdata/hostnetperfbytime** - gives you the network performance of each of the hosts as well as cloud probes by Time of Day. The data is displayed in milliseconds, and represents the total network time for health probes on each day of the week for the prior 30 days.
- Host Network Performance - Day of Week: **https://www.office365mon.com/api/reportdata/hostnetperfbyday** - gives you the network performance of each of the hosts as well as cloud probes by Day of Week. The data is displayed in milliseconds, and represents the total network time for health probes on each day of the week for the prior 30 days.
- Probe Processing Pipeline: **https://www.office365mon.com/api/reportdata/probeprocessingpipeline** - gives you a view of how much time - server and network - each host spends while issuing health probes The data is displayed as a percentage of time processing on the server versus being routed around the network, and is based on data for the prior 30 days.
- Tenant Wide Health – **ONLY AVAILABLE FOR OFFICE365MON PRIVATE DEPLOYMENTS!**: **https://www.office365mon.com/api/reportdata/tenantwidehealth** - shows the number of health checks and average response time across all SharePoint, OneDrive for Business, and Exchange resources in your tenant that have been sent health probes. With Office365Mon Private Deployments, resources from across your tenant are randomly checked.
- Distributed Service Ping Time: there are three slices of data for Distributed Service Ping Time – recent, daily and monthly. They contain ping times to core services – Dns, Proxy, and Office 365 services – across all of the locations you are using the latest Distributed Probes and Diagnostics agent. They are found at the following Urls respectively:
    - **https://www.office365mon.com/api/reportdata/serviceperfrecent**
    - **https://www.office365mon.com/api/reportdata/serviceperfdaily**
    - **https://www.office365mon.com/api/reportdata/serviceperfmonthly**
- Internet Egress Info: **https://www.office365mon.com/api/reportdata/ipegressinfo** - provides you a current and historical view of the Internet egress points being used at each location where you are using the Distributed Probes and Diagnostics agent.

The **AzureServiceMon** endpoints are as follows:

- Monthly Resource Outages: **https://azureservicemon.com/report/azuredata/resourceoutages** - shows you monthly outage summary data for each monitored Azure resource.
- Monthly Resource Outages: **https://azureservicemon.com/report/azuredata/resourcetypeoutages** - shows you monthly outage summary data for each monitored Azure resource type.
- Monthly Resource Health Checks: **https://azureservicemon.com/report/azuredata/resourcehealthchecks** - shows you the number of health checks by month for each monitored resource.
- Monthly Resource Pivot Data: **https://azureservicemon.com/report/azuredata/pivotdata** - shows you a normalized set of data for outages and health checks as used in the Pivot table and chart reports.
- Outage Details: **https://azureservicemon.com/report/azuredata/outagedetails** - returns a list of each individual outage that has been recorded.
- Health Check Failures: **https://azureservicemon.com/report/azuredata/healthcheckfailures** - returns a list of each time a health check failed to get processed by Azure.
- Recent Metric Alerts: **https://azureservicemon.com/report/azuredata/recentmetricalerts** - returns a list of each individual metric monitoring alert that was issued by AzureServiceMon.
- Recent Metric Checks: **https://azureservicemon.com/report/azuredata/recentmetricchecks** - returns a list of each monitored metric that was recorded over a maximum of the previous 24 hours. The checks are made every five minutes and removed every 24 hours and started over.
- Last Week's Metric Checks: **https://azureservicemon.com/report/azuredata/hourlymetricchecks** - returns a list of metric checks that have been summarized into hourly totals (sums, averages, counts). These metrics go back for the previous seven days.
- Monthly Metric Checks: **https://azureservicemon.com/report/azuredata/monthlymetricchecks** - returns a list of metric checks that have been summarized into monthly totals (sums, averages, counts).

Data Format Options

When you retrieve the data from an endpoint you have the option of specifying whether it should be returned as JSON or CSV. If you don't add an ACCEPT header to your request specifying that you want the data in JSON format, it will be returned as CSV. If you are using the .NET HttpClient class to request the data, you can add the ACCEPT header as follows:

```
myHttpClientInstance.DefaultRequestHeaders.Accept.Add(MediaTypeWithQualityHeaderValue.Par
se("application/json"));
```

A complete code sample is provided at the end of this document. You can also find samples on the Report Data page on our site at https://www.office365mon.com/reports/reportdata and https://azureservicemon.com/reports/reportdata.

Scoping Your Data Request

By default, requests to the API return data for all subscriptions for which the currently logged in person is an administrator. For most customers, you only have one subscription so you only get back one set of data. However if you have multiple subscriptions, such as if you are an Office 365 reseller, then you will get back data from every subscription you administer in Office365Mon.

You can scope the data request though to an individual subscription by using a standard REST format for the subscription – you simply append the subscription ID to the end of the REST Url. For example, suppose you have a subscription with an ID of 07F1C948-6637-4D9A-84DB-67F4008FDF57. If you wanted to get outage information for just that subscription then the Url you would use to retrieve the data is https://www.office365mon.com/report/data/outages/07F1C948-6637-4D9A-84DB-67F4008FDF57.

If you have a few subscriptions then retrieving them all at once should be inconsequential from a performance perspective. However if you have many subscriptions, or you just want to segment the data you are looking at, then it is strongly recommended to retrieve them individually.

Working with Azure Active Directory and Office365Mon

There are a set of very specific steps you need to follow in order to work with the Office365Mon APIs:

- Log into the Office365Mon site and create a subscription
- Create an application in Azure Active Directory that trusts the Office365Mon application
- Get an access token
- Request the data and provide the access token as part of the request.

Each of these steps are described in detail below. If you have any other questions please feel free to email support@office365mon.com.

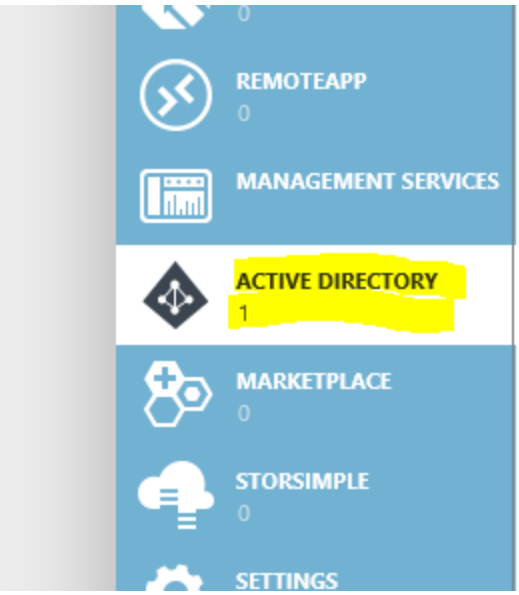
Getting An Access Token and Retrieving Data from Office365Mon.Com

Before you can access the Office365Mon API you must first sign into Office365Mon.Com at least once, and consent to using the application. Once that is complete you can continue below.

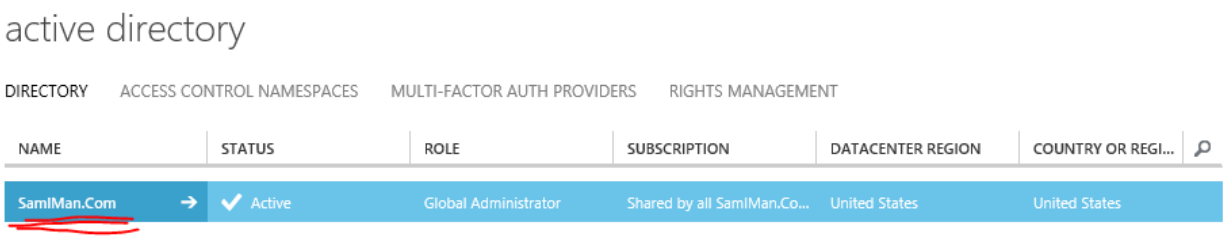
Create A Client Application for Accessing Office365Mon

The first step you need to do is to create a client application in your Azure Active Directory (AAD) tenant that you will use for accessing the Office365Mon API. You should create this application in whatever AAD tenant contains the user account that you are going to use to access the Office365Mon API. In this example, a new client application will be created in the SamlMan.Com AAD tenant.

1. Open your browser and navigate to the Azure management portal.
2. Click on Active Directory in the left navigation.



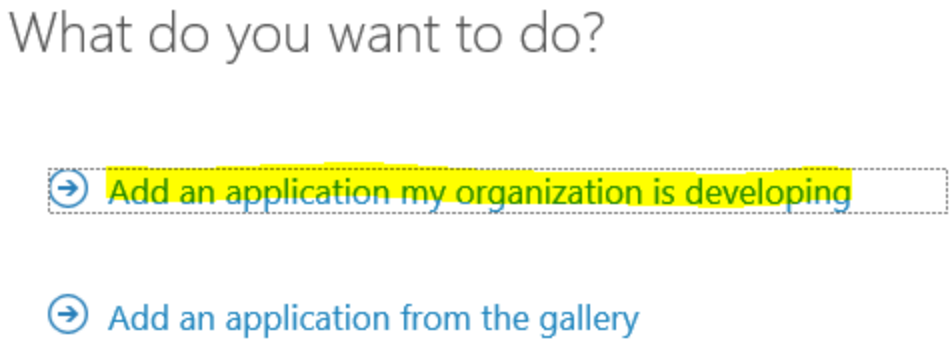
3. Click on the name of your AAD tenant.



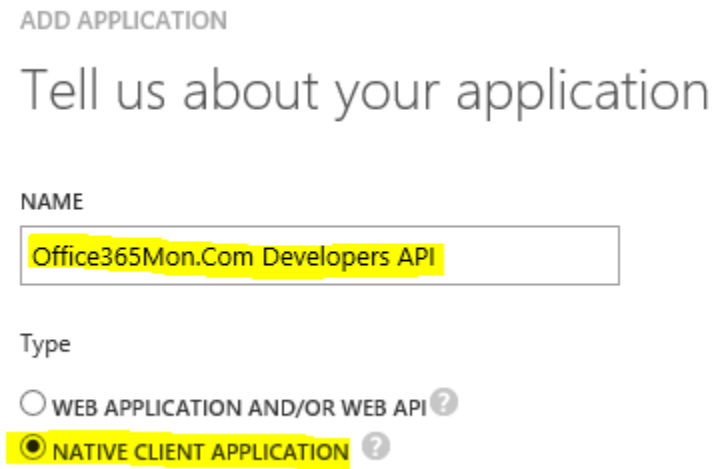
4. Click on Applications in the top navigation bar.

5. Click on ADD+ in the bottom tool bar.

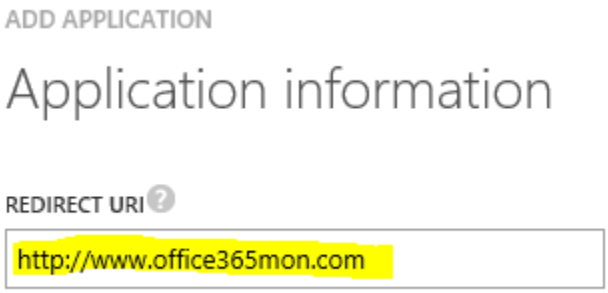
6. Click on “Add an application my organization is developing”



7. Type in a name for the application and select the NATIVE CLIENT APPLICATION option if you are building a console app (such as this example is based on), or WEB APPLICATION AND/OR WEB API if you are building a web site, such as an Office365Mon partner white label web site.



8. Type a value in the REDIRECT URI edit box. It can be any value you want, it just needs to be unique among all applications you have registered in your AAD tenant.



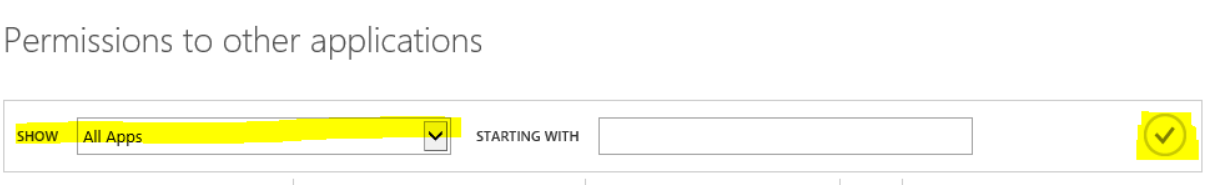
9. Click the checkmark button to complete the application.

Configure the Client Application Permissions to Access Office365Mon

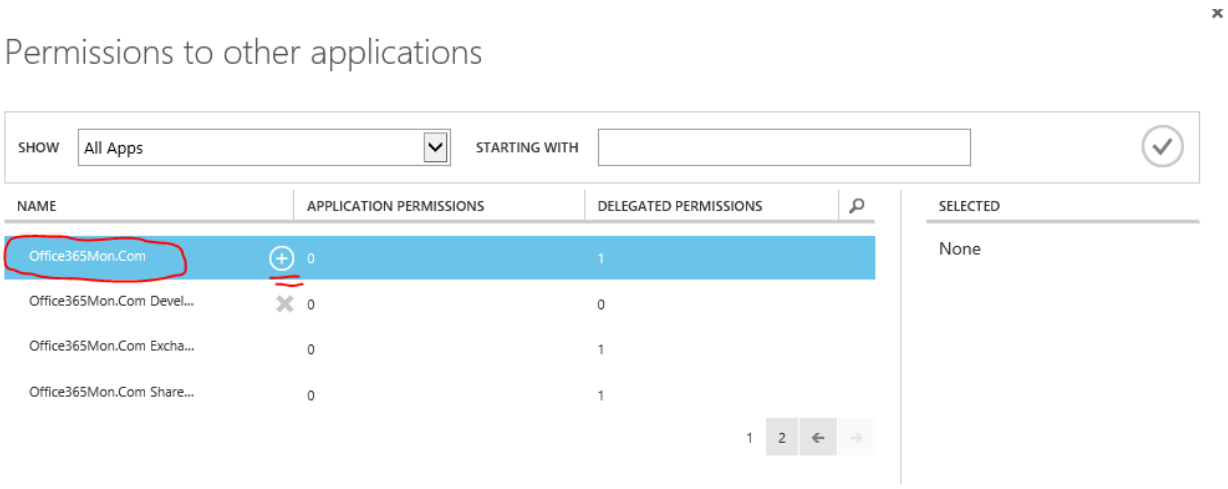
In this step you're going to configure the permission set for the application to include Office365Mon. It assumes that you still have the browser open, and it is displaying the dashboard for the application you created in the previous section. If not, you need to go back into the Azure management portal, click on your AAD tenant, click on Applications, and then click on the application in AAD that you created in the previous section.

1. Click on the Configure tab in the top navigation.

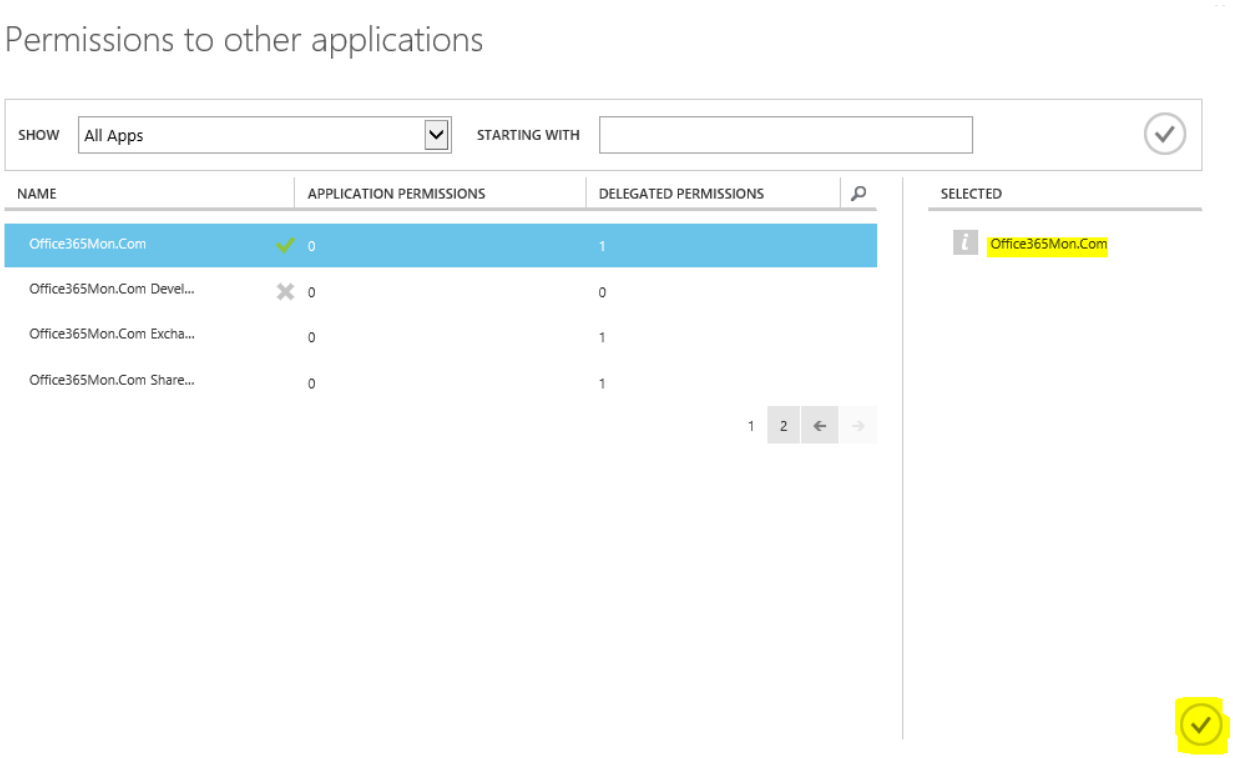
2. Scroll to the bottom of the page to the "permissions to other applications" section.
3. Click the "Add application button"
4. In the "SHOW" drop down at the top of the dialog, select "All Apps" then click the checkmark button to find all of your apps.



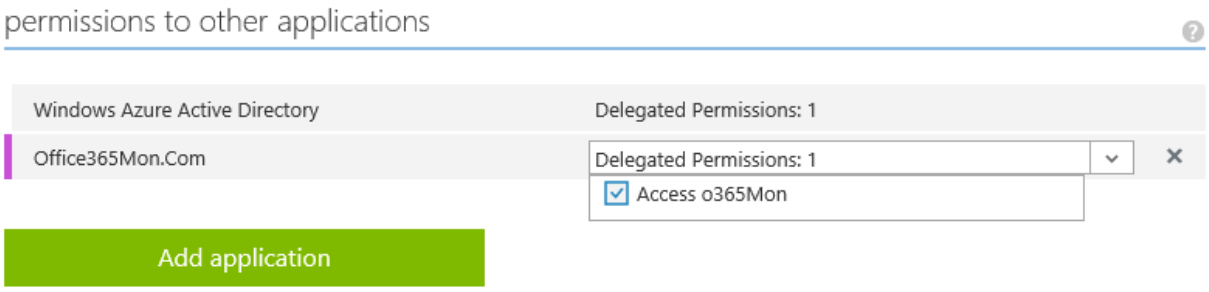
5. Scroll through the list of apps until you find the "Office365Mon.Com application. Note that if you have several apps you many need to page through them until you find it. Also note that there are three apps used by Office365Mon, and depending on what you've configured in the Office365Mon.Com site, you may see one, two or all three of them. Regardless of how many you see, there is only one titled "Office365Mon.Com" and that is the one you need to select. To select it, click the + sign next to the application name.



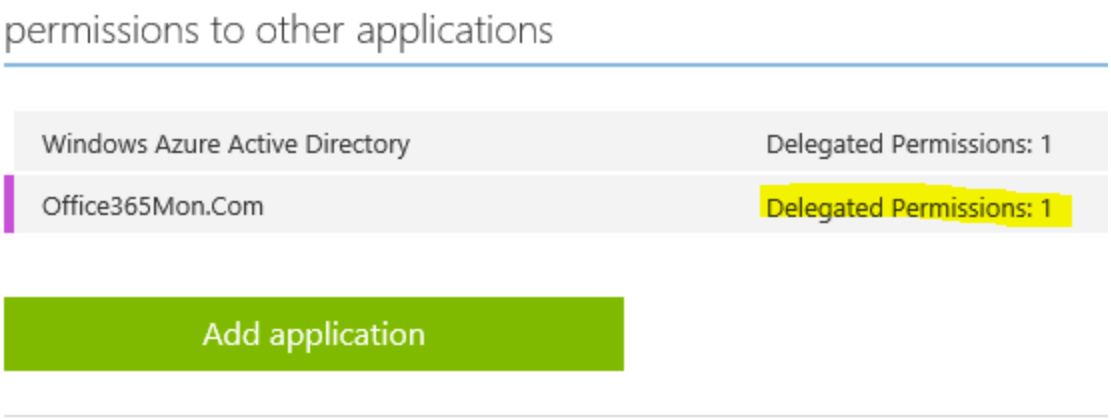
6. The Office365Mon.Com application should now appear in the SELECTED column in the dialog; when it does you can click the checkmark button on the bottom of the dialog to continue.

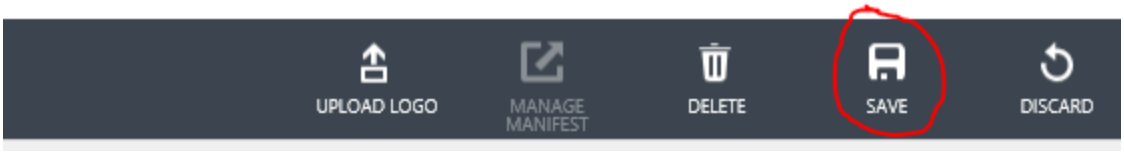


7. Click on the Delegated Permissions drop down next to the Office365Mon.Com application in the “permissions to other applications” section, and check the box next to “Access o365Mon”.



8. The Delegated Permissions input should now have the number “1” next to it, then click the SAVE button in the bottom toolbar.

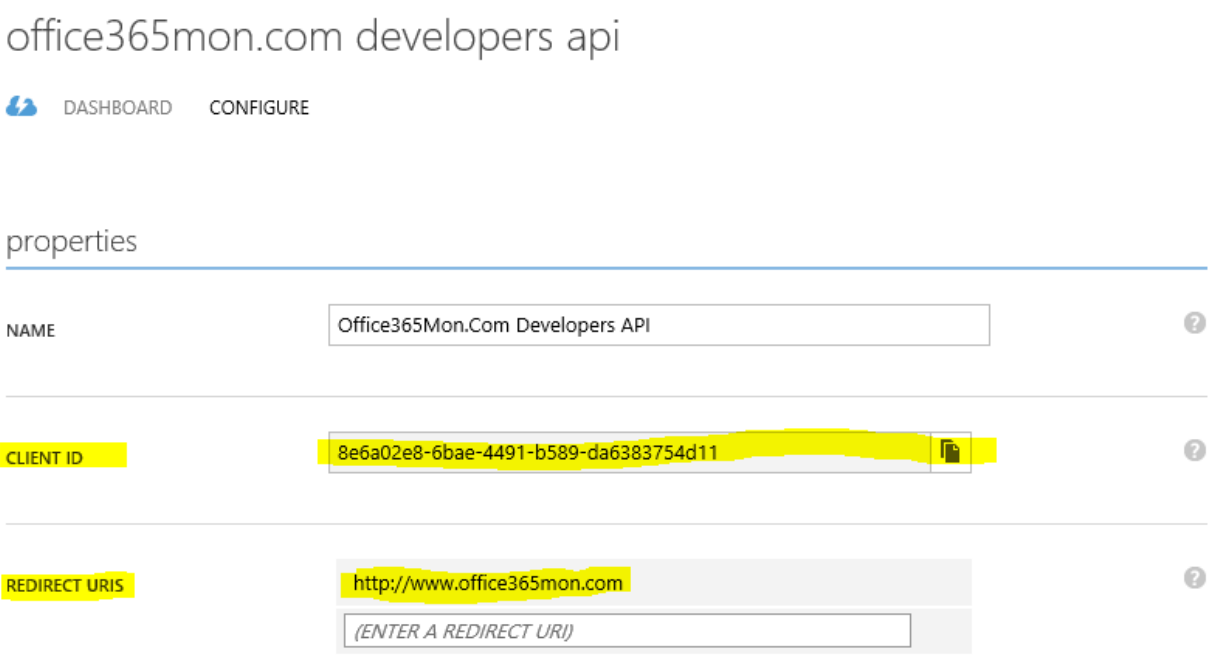




Copy Client ID and Redirect URI

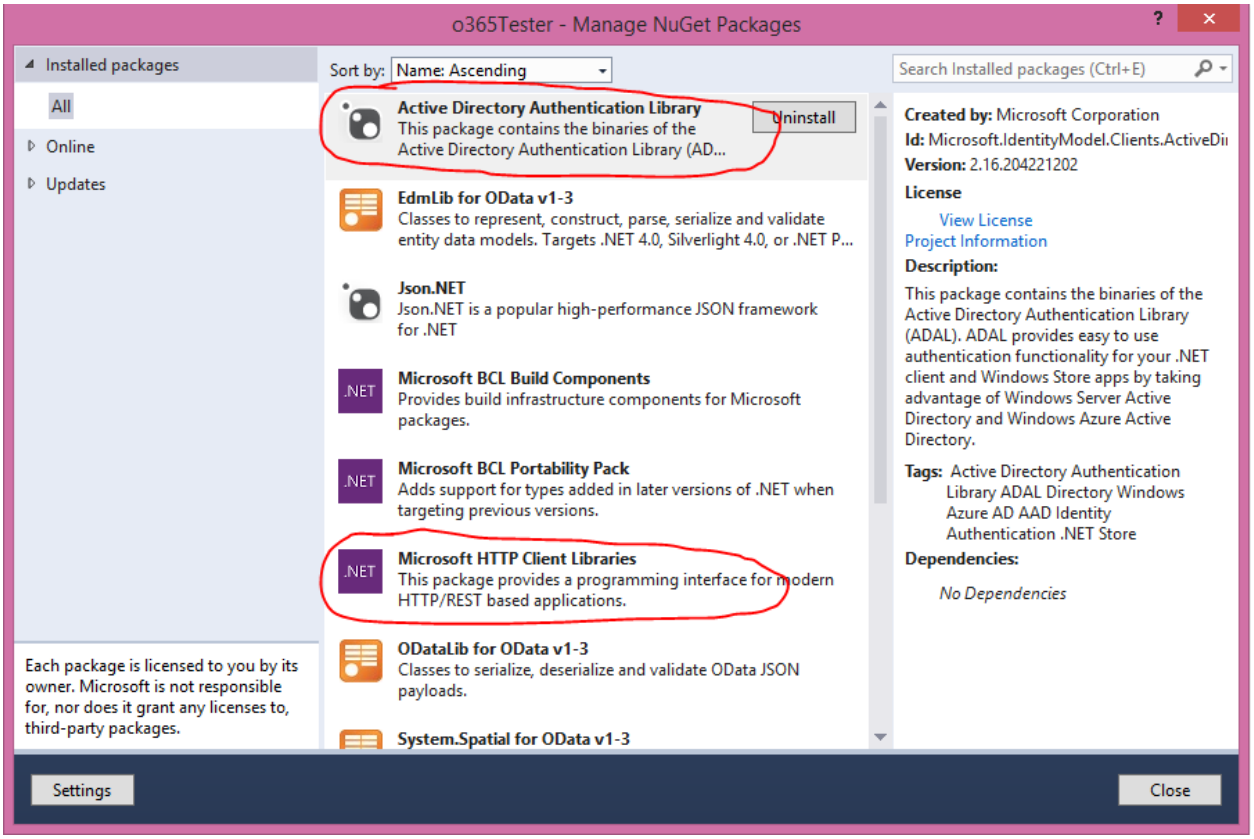
In this step you're going to copy the client ID and redirect URI for the application you created. You need these values in order to get an access token that you can use to use the Office365Mon developers API. It assumes that you still have the browser open, and it is displaying the Configure page for the application you created in the first section.

1. Scroll to the top of the page of the application.
2. Find the CLIENT ID edit box and copy the value. This will be the "CLIENT_ID" value in the code sample.
3. Find the REDIRECT URIS edit box and copy the value. This will be the "REDIRECT_URI" value in the code sample.



Get an Access Token and Access Office365Mon Data

You are now ready to take the values created in the previous section to get an access token and retrieve data from the Office365Mon API. Following is some sample code to demonstrate how to do this. This sample is used in a C# winforms application and assumes you have added NUGET packages for Active Directory Authentication Library and Microsoft HTTP Client Libraries:



This is the code sample:

```
AuthenticationContext ac =
       new AuthenticationContext("https://login.windows.net/common");

//here are what these parameters are:
//1.  https://www.office365mon.com/monitor is the ID for Office365Mon Developers API
//2.  8e6a02e8-6bae-4491-b589-da6383754d11 is the CLIENT_ID of your Azure AD app
//3.  http://www.office365mon.com is the REDIRECT_URI of your Azure AD app

AuthenticationResult ar =
       ac.AcquireToken("https://www.office365mon.com/monitor",
       "8e6a02e8-6bae-4491-b589-da6383754d11",    //CLIENT_ID
       new Uri("http://www.office365mon.com"));  //REDIRECT_URI

HttpClient hc = new HttpClient();

hc.DefaultRequestHeaders.Authorization =
       new AuthenticationHeaderValue("Bearer", ar.AccessToken);

//NOTE:  the ACCEPT header has not been added so you will get CSV back
//the URL used here is a REST endpoint you can use to test your connection
HttpResponseMessage rm =
       hc.GetAsync("https://www.office365mon.com/report/data/outages")
       .Result;

if (rm.IsSuccessStatusCode)
```

```
{
        string data = rm.Content.ReadAsStringAsync().Result;
        MessageBox.Show("The data is: " + data);
}
else
{
        MessageBox.Show("request failed!!: " + rm.ReasonPhrase);
}
```