CARBONITE

Carbonite Server Backup API – Monitoring v1.5

Installation and Usage Guide

Carbonite, Inc.
Two Avenue de Lafayette
Boston, MA 02111
www.carbonite.com

Acknowledgements: Two encryption methods, DES and TripleDES, include cryptographic software written by Eric Young. The Windows versions of these algorithms also include software written by Tim Hudson. Bruce Schneier designed Blowfish encryption.

The Carbonite Server Backup Agent, Carbonite Server Backup CentralControl, and Carbonite Server Backup Director applications have the encryption option of AES (Advanced Encryption Standard). Advanced Encryption Standard algorithm (named Rijndael, pronounced “Rain Doll”) was developed by cryptographers Dr. Joan Daemen and Dr. Vincent Rijmen. This algorithm was chosen by the National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce to be the new Federal Information Processing Standard (FIPS).

The Carbonite Server Backup Agents and Carbonite Server Backup Director applications also have the added security feature of an over the wire encryption method.

Document History

Version	Date	Description
1	November 2019	Initial installation and usage guide provided for API – Monitoring 1.5*x*.

Contents

1 Introduction to Carbonite Server Backup API – Monitoring

Carbonite Server Backup API – Monitoring is a public OData Web API that enables service providers to:

- Remotely monitor their protected data. As described in Monitoring overview, client applications can make read-only calls to API – Monitoring to obtain information about the following entities in Carbonite Server Backup Portal and in Carbonite Server Backup Director vaults:
  - Agents
  - Companies (sites)
  - Company users
  - Jobs, job schedules and retentions
  - Safesets
  - Data deletion requests
  - Vaults

  Detailed information about available calls is available in the Swagger UI that is installed with API – Monitoring. See View documentation for API – Monitoring calls.
- Delete data from Director vaults in response to requests from Portal. See Data deletion overview.

To install and set up API – Monitoring, see Prepare for an API – Monitoring installation, Install or upgrade API – Monitoring and Configure and maintain API – Monitoring.

To monitor protected data using API – Monitoring, see the required steps in Monitor protected data.

To delete data from Director vaults in response to request from Portal, see the required steps in Set up data deletion.

Note: API – Monitoring is sometimes referred to as "the API" in this guide.

Because API – Monitoring uses structured logging, you can use a tool such as SEQ or Splunk to analyze API log files. By default, log files are saved in C:\Logs\Carbonite Server Backup API\Monitoring.

1.1 Monitoring overview

To monitor agents, jobs, safesets and other entities in Carbonite Server Backup, client applications can make read-only calls to API – Monitoring. In response, the API provides requested data from Portal databases and from a Monitoring database that is installed with the API.

As shown in the Monitoring overview diagram, a Reporting service on each registered vault sends data to API – Monitoring. API components then process and store the data in the Monitoring database. A Reporting service should be installed on each vault where your data is stored.

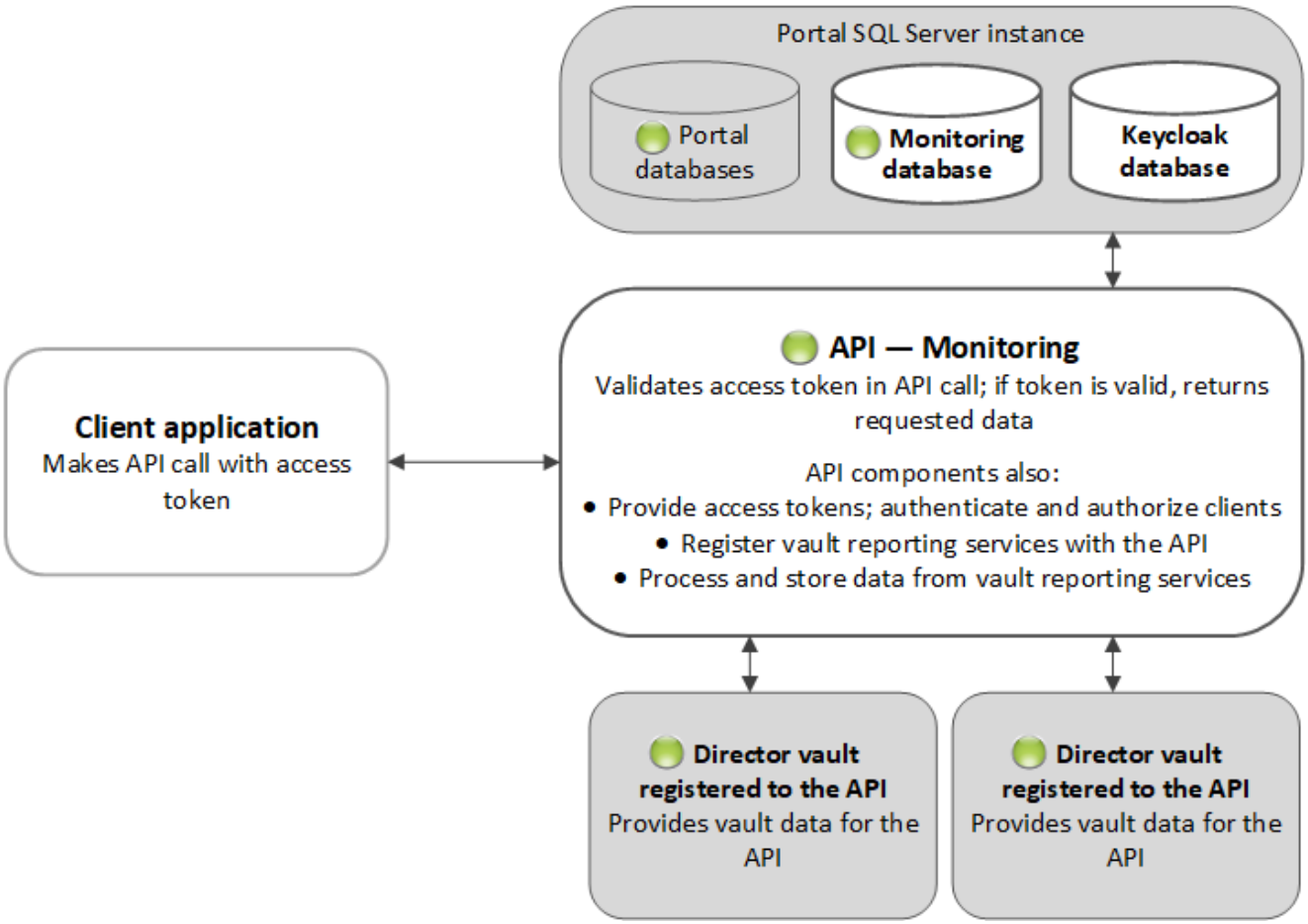
To authenticate and authorize API calls, the API uses an authorization server that is installed with API – Monitoring. Client applications obtain access tokens from the authorization server and include them in calls to the API. If a token is valid, the API returns the requested data from the Portal and Monitoring databases.

Tokens from the authorization server are also used to authorize communications between Reporting services and the API.

To view available API calls, see View documentation for API – Monitoring calls.

To start monitoring protected data using API – Monitoring calls, see required steps in Monitor protected data.

Monitoring overview diagram



1.2 Data deletion overview

As shown in the Data deletion overview diagram, when Portal and Director vaults are registered to the same API – Monitoring instance, Director can delete job and computer data in response to requests from Portal.

When deleting a job or computer in Portal, a Portal Admin user can request that backup data for the job or computer be deleted from all vaults. The data deletion is scheduled for 72 hours after the request is made.

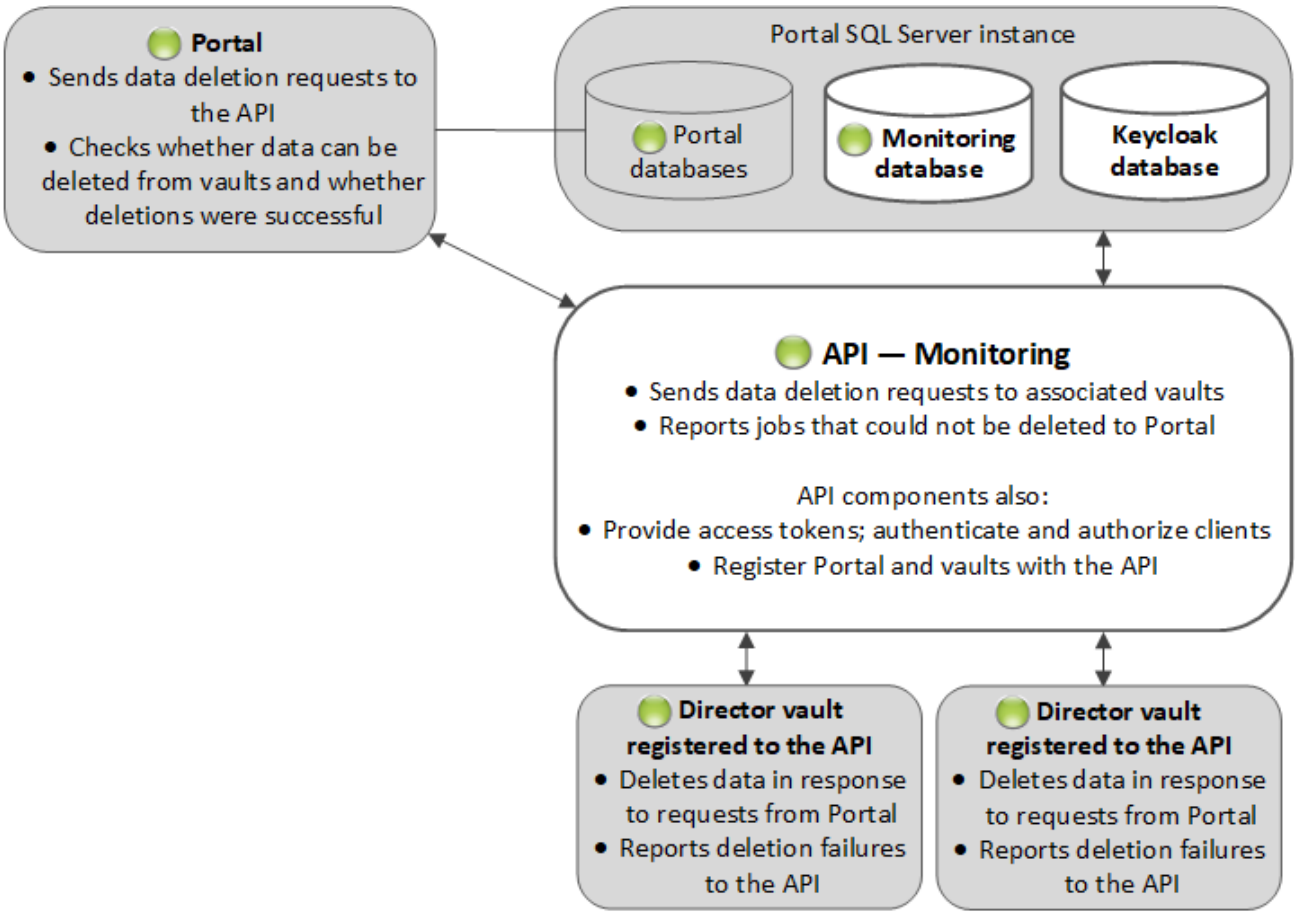
If an Admin user does not cancel a scheduled data deletion during the 72-hour waiting period, the deletion request is sent to Director vaults that are registered to API – Monitoring. In response to the request, Director deletes the data from any standalone, Base or Active vault where the data is stored. Replication processes then delete the data from any associated Satellite or Passive vault.

Note: Although it is not required for data deletion, the Reporting service must be installed with Satellite and Passive vaults and registered to API – Monitoring to provide data through API – Monitoring calls.

If a deletion request fails, an email notification is sent to a vault administrator whose email address is specified in Portal. The vault administrator can then manually delete the data.

For required steps for setting up data deletion, see Set up data deletion.

Data deletion overview diagram



1.3 API – Monitoring components

When you install API – Monitoring, the following system components and their prerequisites are installed:

Component	Description
API – Monitoring	Validates access tokens in API calls. If access tokens are valid and have correct authorization, returns requested data from the Portal and Monitoring databases.
Monitoring database	Stores data received from Director vaults. Installed on the Portal database SQL Server instance.

Component	Description
Keycloak	Third-party authorization server that is used to authenticate and authorize clients and API system components.
Keycloak database	Database that includes client authorization information for Keycloak.
Message Bus Authentication	Validates tokens from vault reporting services and authorizes communications with the message bus.
RabbitMQ Server	Third-party message bus enabling asynchronous communication among Carbonite Server Backup solution components.
Collector	Processes data coming from Carbonite Server Backup solution components and makes them available for querying via the API – Monitoring.
Registration Service	Securely registers Carbonite Server Backup solution components with the API – Monitoring system. Enables secure communication between components and common authorization mechanism.

2 Prepare for an API – Monitoring installation

Before installing API – Monitoring, you must do the following:

- Determine where to install API – Monitoring
- Ensure that the server where you want to install the API meets all system requirements and that all prerequisites are installed. See Check system requirements.
- Determine how to secure API – Monitoring
- Set up a public domain name or IP address
- Plan ongoing SQL Server database maintenance

For compatible Portal and Director versions, see the API – Monitoring release notes.

2.1 Determine where to install API – Monitoring

You can install API – Monitoring on:

- A server where no Portal components are installed. Carbonite recommends this installation location.
- A server where Portal components are installed. Although these installation locations are not recommended:
  - If all Portal components are installed on one server, you can install the API on the Portal server.
  - If Portal is installed as a distributed system, you can install the API on the same server as front-end Portal components. In this case, Portal and API – Monitoring must have different IP addresses or host names (e.g., portal.domain.com and api.domain.com). Do not use a subdirectory of the Portal URL for the API (e.g., portal.domain.com/api).

When installing API – Monitoring on a separate server from the Portal SQL Server instance, you can choose SQL Server authentication or Windows authentication to connect to SQL Server. Carbonite recommends using SQL Server authentication in this case. To use Windows authentication, the servers must be in the same domain and you must run the API installer using a domain administrative account that has sysadmin rights to the SQL Server instance.

2.2 Check system requirements

The server where you install API – Monitoring must meet the requirements described in the following sections:

- Hardware requirements
- Software requirements
- Required ports

2.2.1 Hardware requirements

API – Monitoring requires a minimum of:

- 8 CPUs
- 8 GB of RAM
- 50 GB of free disk space

Note: If API – Monitoring is installed on the same server as the Portal database, these hardware requirements are in addition to Portal requirements.

2.2.2 Software requirements

Before you can install the API, Portal and its prerequisites must be installed, either on the server where you are installing the API or on a separate server. See Determine where to install API – Monitoring.

Portal databases must be running on SQL Server or SQL Server Express 2017, 2016, 2014, 2012 SP1 or 2008 R2 (64-bit). The latest SQL Server service packs and updates are recommended.

TCP/IP must be enabled for the SQL Server instance and the SQL Server Browser service must be running.

The following prerequisites must be installed on the server where you install the API:

- .NET Framework 4.7
- Powershell version 5.0 or later

We recommend using TLS 1.2 for secure communications. API – Monitoring components cannot communicate with each other using TLS 1.0 or SSL 3.0.

2.2.3 Required ports

For the API – Monitoring installation to succeed, port 888 must be reserved for internal communications.

In addition, the external ports shown in following table must be free of other services and open for API – Monitoring components.

Component	Port	Protocol
Registration service	8080	HTTP or HTTPS
Keycloak endpoint	8081	HTTP or HTTPS
RabbitMQ endpoint	5672	AMQP (Advanced Message Queuing Protocol)
	5672 and 5671	AMQPS (secured Advanced Message Queuing Protocol)
API – Monitoring endpoint	80	HTTP
	443	HTTPS

2.3 Determine how to secure API – Monitoring

To secure the API, you can select the HTTPS communication protocol when you install the API and select a certificate in .pfx format. A .pfx file contains the certificate and private key for the API server, and is password-protected. The API – Monitoring installer will prompt you for the .pfx file location and password.

For a production environment, Carbonite recommends obtaining a certificate from a Certificate Authority. For a test environment, you can create a self-signed certificate. The certificate's identity must match the API – Monitoring hostname provided during the installation, and must be trusted by the machine where it is installed.

Alternatively, you can use a third-party TLS termination proxy to handle secure connections, and select the HTTP communication protocol when you install the API. Because Vault Reporting services can only communicate with API – Monitoring with a secured channel using TLS, a TLS termination proxy is required if the API is installed with the HTTP communication protocol.

2.4 Set up a public domain name or IP address

When you install API – Monitoring, you must enter a public domain name or IP address for the API. API – Monitoring must have a different host name or IP address than Portal (e.g., api.domain.com and portal.domain.com). Do not use a subdirectory of the Portal URL for the API (e.g., portal.domain.com/api).

If you enter a domain name, the name must be properly registered with your domain server or it will not be available outside the system. To verify that clients can access API – Monitoring, use the ping command to check that the domain name resolves on the client system that is trying to access the API.

2.5 Plan ongoing SQL Server database maintenance

For best API performance, ongoing SQL Server database maintenance is required.

We recommend rebuilding table indexes for the Monitoring database nightly, or at least once per week. To achieve this, you can create a SQL scheduled job. A fill factor setting of 75-80% is recommended. For more information, see Microsoft SQL Server documentation: https://docs.microsoft.com/en-us/sql/relational-databases/indexes/reorganize-and-rebuild-indexes?view=sql-server-2014. To rebuild indexes using a script, the following page provides a starting point: https://gallery.technet.microsoft.com/scriptcenter/Script-for-rebuilding-all-8d079754

3 Install or upgrade API – Monitoring

After planning your API – Monitoring installation as described in Prepare for an API – Monitoring installation, you can install the API and verify that API – Monitoring is working. See Install API – Monitoring and Verify an API – Monitoring installation. If a previous version of the API is installed, you can upgrade API – Monitoring. See Upgrade API – Monitoring.

After API – Monitoring 1.5 is installed, you can configure and maintain the installation. See Configure and maintain API – Monitoring.

You can also uninstall API – Monitoring or reinstall API – Monitoring.

3.1 Install API – Monitoring

To install API — Monitoring:

1. Sign in as an administrator to the server where you want to install API – Monitoring.

   Note: Carbonite recommends installing API – Monitoring on its own server where no Portal components are installed. For more information, see Determine where to install API – Monitoring.

2. Double-click the API – Monitoring installation kit.

3. On the Welcome page, click **Next**.

4. On the License Agreement page, read the license agreement. Select **I accept the terms in the license agreement**, and then click **Next**.

5. On the Destination Folder page, note the API installation location (C:\Program Files\Carbonite Server Backup API), and then click **Next**.

Note: The Collector, Monitoring, Registration and Message Bus Authentication services will be installed in subfolders in C:\Program Files\Carbonite Server Backup API. The Message Bus Authentication service subfolder name will be "RMQ Authentication Server". The remaining components will be installed in their default locations:

- Adopt OpenJDK: C:\Program Files\AdoptOpenJDK
- Keycloak: C:\keycloak-3.2.1.Final
- Erlang : C:\Program Files\erl9.2
- RabbitMQ: C:\Program Files\RabbitMQ Server

Installation logs for all API components are saved in C:\Logs.

6. On the API Configuration page, do the following:

   a. Select the communication protocol: **https** or **http**

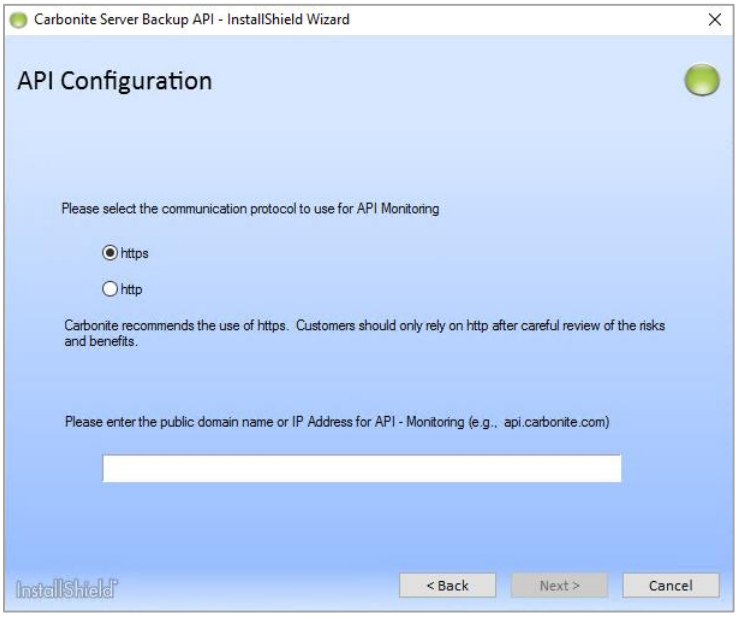
      If you select the https protocol, a certificate in .pfx format is required. If you select the http protocol, a third-party TLS termination proxy is required. See Determine how to secure API – Monitoring.

   b. Type the public domain name or IP address for the API (e.g., api.carbonite.com or 192.168.1.20). Do ***not*** include http:// or https:// with the domain name or IP address.

      API – Monitoring must have a different IP address or host name than Portal (e.g., api.domain.com and portal.domain.com). Do not use a subdirectory of the Portal URL for the API (e.g., portal.domain.com/api).

      As described in Set up a public domain name or IP address, a domain name must be properly registered with your domain server so that clients can access the system.

   Click **Next**.



If you select the https communication protocol, the Certificate Selection page appears. On the Certificate Selection page, do the following:

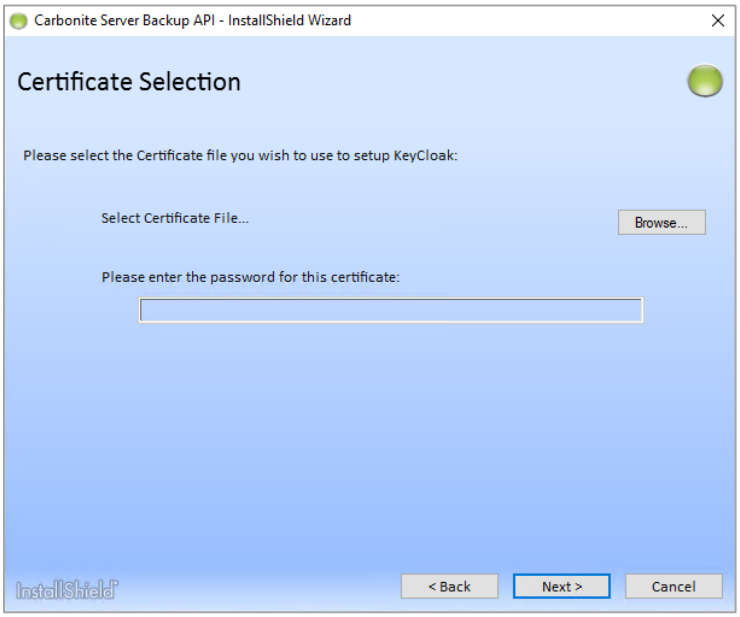
a. Click **Browse**. In the Select Certificate File dialog box, navigate to the certificate file in .pfx format and click **Open.**

   The certificate's identity must match the API – Monitoring hostname provided during the installation.

   For a production environment, Carbonite recommends obtaining a certificate from a Certificate Authority. For a test environment, you can create a self-signed certificate. If the certificate is self-signed, it must be copied into the Trusted Root Certification Authorities Certificate store for the server.

b. In the password box, enter the password for the certificate file in .pfx format.

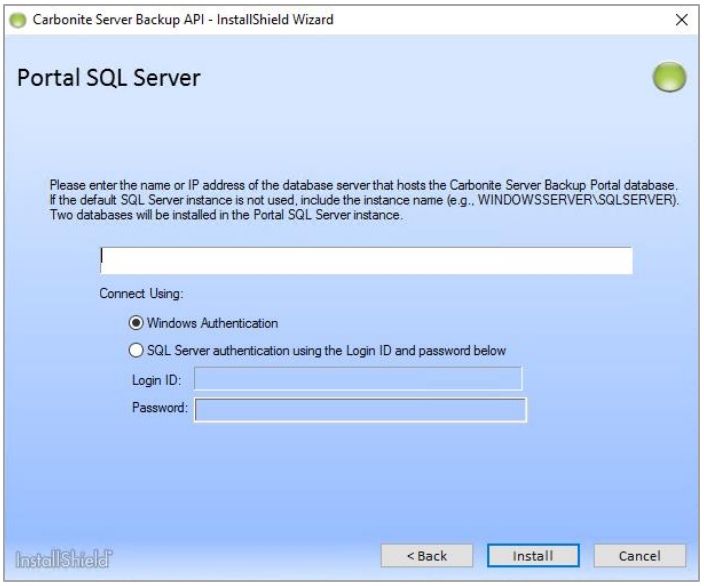
c. Click **Next**.



7. On the Portal SQL Server page, type the name of the database server that hosts the Carbonite Server Backup Portal database (e.g., WINDOWSSERVER). If the default SQL Server instance is not used, include the instance name (e.g., WINDOWSSERVER\SQLSERVER).

   Note: If you are installing API – Monitoring on the same server as the Portal SQL Server instance, you can type `localhost` as the database server name. Do not type `(local)` as the database server name.

   Two databases will be installed in the Portal SQL Server instance: a Monitoring database, which will store data from Vault Reporting services, and a KeyCloak database.



8. Do one of the following:

   - To connect to SQL Server using the current Windows credentials, select Windows Authentication.

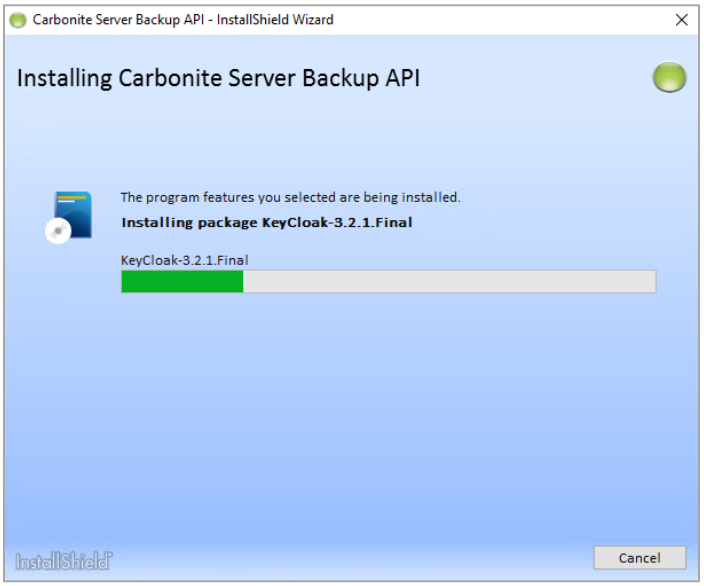
     For Windows authentication, you must be signed in as an administrator that has sysadmin rights to the SQL Server instance where Portal databases are running.

   - To connect to SQL Server using SQL Server credentials, select SQL Server authentication using the Login ID and password below. In the Login ID and Password boxes, enter SQL Server credentials.

9. Click **Install**.

   The installer begins installing and configuring API components and prerequisites. The Installing Carbonite Server Backup API screen shows the component currently being installed, and the installation progress.

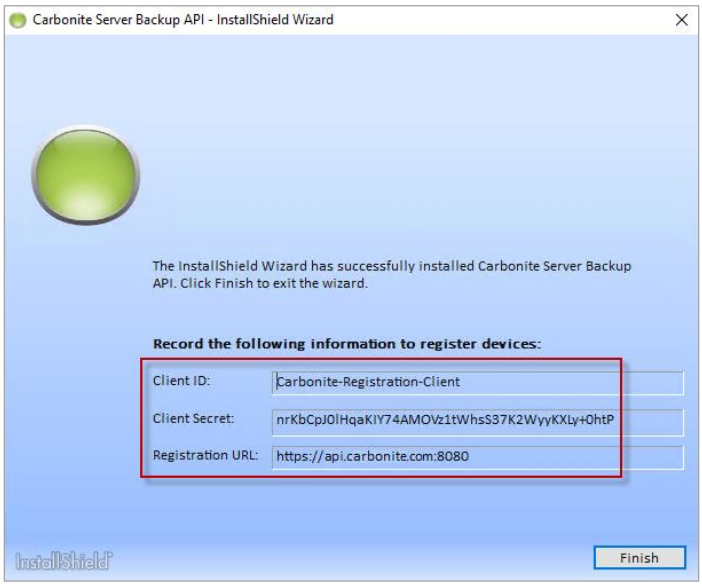
   Note: The installation of API components can take a long time (e.g., 15 minutes).



When the installation is finished, the InstallShield Wizard shows information that is required for registering Portal and Vault Reporting services to the API.

IMPORTANT: Copy the Client ID, Client Secret and Registration URL values from the InstallShield Wizard page. These values are required for registering Portal and Vault Reporting services to the API.

IMPORTANT: To ensure the security of your data, keep the Client ID and Client Secret values private and secure.



10. Click **Finish**.

    Once API – Monitoring is installed, you can view detailed information about available API – Monitoring calls. See View documentation for API – Monitoring calls.

3.2 Upgrade API – Monitoring

You can upgrade API – Monitoring. After an upgrade, the Client ID, Client Secret and Registration URL values used to register Vault Reporting services remain the same as before the upgrade. You can also register Portal to the API using these values.

Note: You cannot change from Windows authentication to SQL Server authentication during an upgrade.

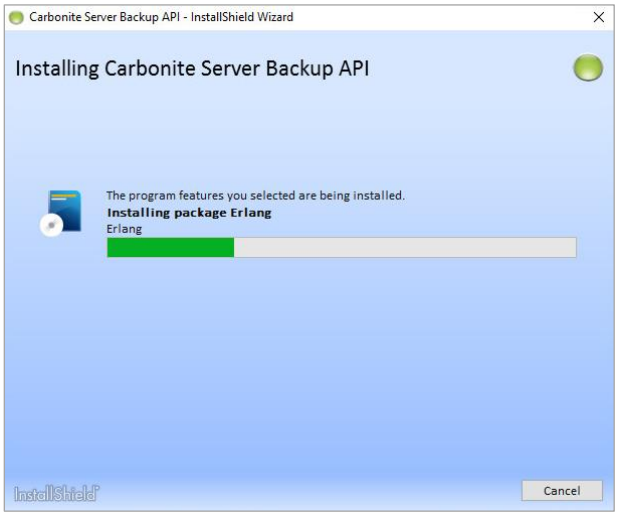
To upgrade API – Monitoring:

1. Sign in as an administrator on the server where you want to upgrade API – Monitoring.
2. Double-click the API – Monitoring installation kit.

   Note: There could be a significant delay before the upgrade begins.
3. On the Welcome page, click **Next**.

   The installer begins upgrading API components. The Installing Carbonite Server Backup API screen shows the component currently being upgraded and the upgrade progress.

Note: The upgrade can take a long time (e.g., five minutes).



4. When the upgrade is finished, click **Finish**.

   Once API – Monitoring is installed or upgraded, you can view detailed information about available API – Monitoring calls. See View documentation for API – Monitoring calls.

3.3 Troubleshoot an API – Monitoring installation

If a problem occurs during an API – Monitoring installation, you can run the installer in debug mode to collect more detailed logs.

To troubleshoot an API – Monitoring installation:

1. Create a directory for saving the detailed installation logs (e.g., C:\Logs).
2. At a command prompt, run the following command:

   installKitName `/debuglog"`*location*`\`*fileName*`"`

   Where:

   - installKitName *is the name of the API – Monitoring installation kit.*
   - location *is the directory for saving the logs (created in Step 1).*
   - fileName *is the file name of the log file.*

   For example, the command could be:

   ```
   CarboniteServerBackupAPI.Monitoring.1.5.x.xxxx
   /debuglog"C:\Logs\Install.log"
   ```

3.4 Verify an API – Monitoring installation

To verify the API – Monitoring installation:

1. Check that the following services are running:
   - Carbonite Registration Service
   - Carbonite Server Backup API – Collector
   - Carbonite Server Backup API – Message Bus Authentication
   - Carbonite Server Backup API – Monitoring
   - KeyCloak
   - RabbitMQ
2. In a web browser, go to the Swagger UI for API – Monitoring:

   `https://`*APIdomainNameOrIPaddress*`/monitoring/swaggerui/index`

   Where *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of Install API – Monitoring.

   For example: `https://api.carbonite.com/monitoring/swaggerui/index`

   For more information about the Swagger UI, see View documentation for API – Monitoring calls and Test API – Monitoring calls.

3.5 Uninstall API – Monitoring

To uninstall API – Monitoring:

1. In the Control Panel, click **Uninstall a program**.
2. On the Uninstall or change a program page, click an API – Monitoring component (e.g., Carbonite Registration Service) and then click **Uninstall.**
3. In the confirmation dialog box, click **Yes**.
4. Double-click the API – Monitoring installation kit.
5. On the Welcome page, click **Next**.

6. On the Ready to Remove API – Monitoring Components page, click **Begin Removal**.

7. When a message states that the removal of pre-existing components is complete, click **Cancel**.

8. In the confirmation message box, click **Yes**.
9. On the wizard interrupted page, click **Finish**.
10. If desired, manually uninstall Adopt OpenJDK, which was installed with API – Monitoring 1.5.

    If you uninstall API – Monitoring version 1.4.2 or earlier, you can manually uninstall the following software that was installed with earlier API versions: Python and Oracle Java Runtime Environment.

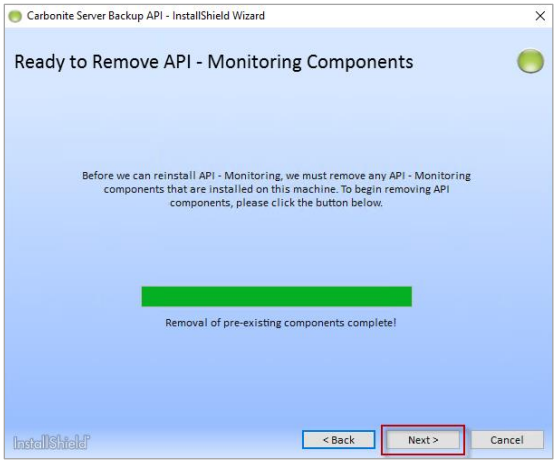
3.6 Reinstall API – Monitoring

Important: Vault Reporting services do not remain registered to API – Monitoring after you reinstall the API. You must re-register Reporting services to API – Monitoring if you reinstall the API. See Register Director Reporting services to the API.

To reinstall API – Monitoring:

1. In the Control Panel, click **Uninstall a program**.
2. On the Uninstall or change a program page, click an API – Monitoring component (e.g., Carbonite Registration Service) and then click **Uninstall.**
3. Double-click the API – Monitoring installation kit.
4. On the Welcome page, click **Next**.
5. On the Ready to Remove API – Monitoring Components page, click **Begin Removal**.

6. When a message states that the API components have been removed, click **Next**. Complete the installation as described in Install API – Monitoring.



4 Configure and maintain API – Monitoring

4.1 Configure the refresh interval for job and safeset data

To improve the performance of job and safeset calls, API – Monitoring 1.5 caches some required data from Portal databases in the Monitoring database.

Data is refreshed in the Monitoring database every 5 minutes by default, but you can change the refresh interval to a value from 1 minute to 60 minutes. A lower interval (e.g., 1 minute) ensures that data returned by API calls is more up-to-date but requires more resources. A higher interval (e.g., 60 minutes) results in less traffic but the data returned by API calls may be out-of-date.

To configure the refresh interval for job and safeset data:

1. In a text editor, open the PlatformAPI.Monitoring.API.exe.config file from the C:\Program Files\Carbonite Server Backup API\Monitoring folder.
2. Find the following line in the config file:

```
<add key="PlatFormAPI.Monitoring.API.EtlExecutionIntervalMins" value=""></add>
```

   When no value is specified in this line, the default interval of 5 minutes is used.

3. Enter a value from 1 minute to 60 minutes in the line.

   For example, if you want the data to be refreshed every 2 minutes, enter the following line:

```
<add key="PlatFormAPI.Monitoring.API.EtlExecutionIntervalMins" value="2"></add>
```

4. Save the file.

4.2 Replace the API – Monitoring certificate

If API – Monitoring was installed using the HTTPS communication protocol and secured using a certificate, you can replace the certificate. For example, you can replace the certificate if it is approaching its expiry date.

As described in Determine how to secure API – Monitoring, Carbonite recommends obtaining a certificate from a Certificate Authority for a production environment. The certificate must be in .pfx format.

To replace the API – Monitoring certificate:

1. Check whether the Microsoft Visual C++ 2015 (or later) Redistributable is installed on the API – Monitoring server. If it is not installed, download the vc_redist.x64.exe file from Microsoft (https://support.microsoft.com/en-ca/help/2977003/the-latest-supported-visual-c-downloads) and install it on the API – Monitoring server.
2. In a Powershell window, navigate to C:\Program Files\Carbonite Server Backup API\Tools.
3. Run the following command:

   `.\Update-pAPICert.ps1` *logPathAndFileName certificate certificatePassword*

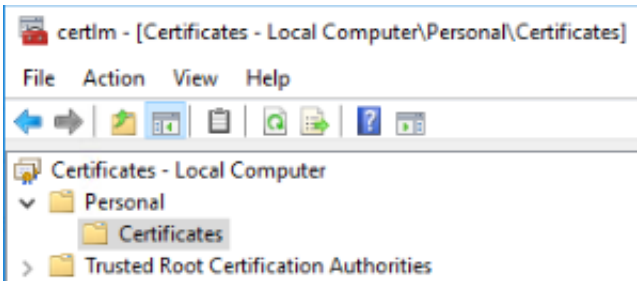
Where:

- *logPathAndFileName* is the path and filename of a log file for the certificate replacement process.
- *certificate* is the path and filename of the new certificate file in .pfx format.
- *certificatePassword* is the password for the certificate file in .pfx format.

For example, you could run the following command:

```
.\Update-pAPICert.ps1 C:\CertLog.txt C:\Certificate.pfx password
```

4. To check that the API – Monitoring certificate was replaced, open the certificate manager on the machine, and view the certificates in the Local Computer Personal certificate store.



5 Register Director Reporting services to the API

To provide vault data through API calls or to delete data in response to requests from Portal, Director Reporting services must be registered to API – Monitoring.

A vault administrator can register a Director Reporting service to the API when the Reporting service is being installed or upgraded, or using a command after the Reporting service is installed. See "Install and register the Reporting service" in the *Director Installation Guide*.

We recommend registering Reporting services to API – Monitoring using a registration token and registration URL. To obtain a registration token, see Obtain a token for registering Director Reporting services to the API.

Director Reporting services can also be registered to the API using Client ID and Client Secret values from the last page of the API – Monitoring installation wizard, but this is not recommended. To protect your data, access to these values should be limited. Keep the Client ID and Client Secret private and secure.

5.1 Obtain a token for registering Director Reporting services to the API

Using a script that is provided with API – Monitoring, you can obtain a token for registering Director Reporting services to the API. This ObtainRegistrationToken.ps1 script requires you to enter the Registration URL, Client ID and Client Secret values from the last page of the API – Monitoring installation wizard, and returns a registration token. You can then give the registration token to the person who is registering one or more Director Reporting services to the API.

A registration token is valid for 48 hours after it is created. If someone tries to register a Reporting service to the API using an expired token, an error appears in the API Registration service log file.

To obtain a registration token for registering a Director Reporting service to API - Monitoring:

1. Save the ObtainRegistrationToken.ps1 script on a machine that can communicate with the API – Monitoring server. Powershell version 5.0 or later must be installed on the machine.
2. In a Powershell window, navigate to the directory where the script was saved in Step 1 of this procedure.
3. Run the following command:
   `.\ObtainRegistrationToken.ps1`
4. At the following prompts, enter the required information from the last page of the API – Monitoring installation wizard (see Install API – Monitoring):
   - **Registration Service Address** – Enter the Registration URL (e.g., `https://api.carbonite.com:8080`).
   - **Client ID** – Enter the Client ID (`Carbonite-Registration-Client`).
   - **Client Secret** – Enter the Client Secret (e.g., `nrKbCpJOlHqaKlY74AMOVz1tWhsS37K2WyyKXLy+0htP`).

   The registration token is returned in the Powershell window. You can copy the token and provide it to the person who is registering Director Reporting services to the API.

6 View documentation for API – Monitoring calls

Client applications can make read-only calls to API – Monitoring to obtain information about entities such as agents, jobs and safesets in Portal and in Director vaults.

For detailed information about available API – Monitoring calls, view documentation in the Swagger UI that is installed with the API. For information about Swagger, see https://swagger.io.

To view documentation for API – Monitoring calls:

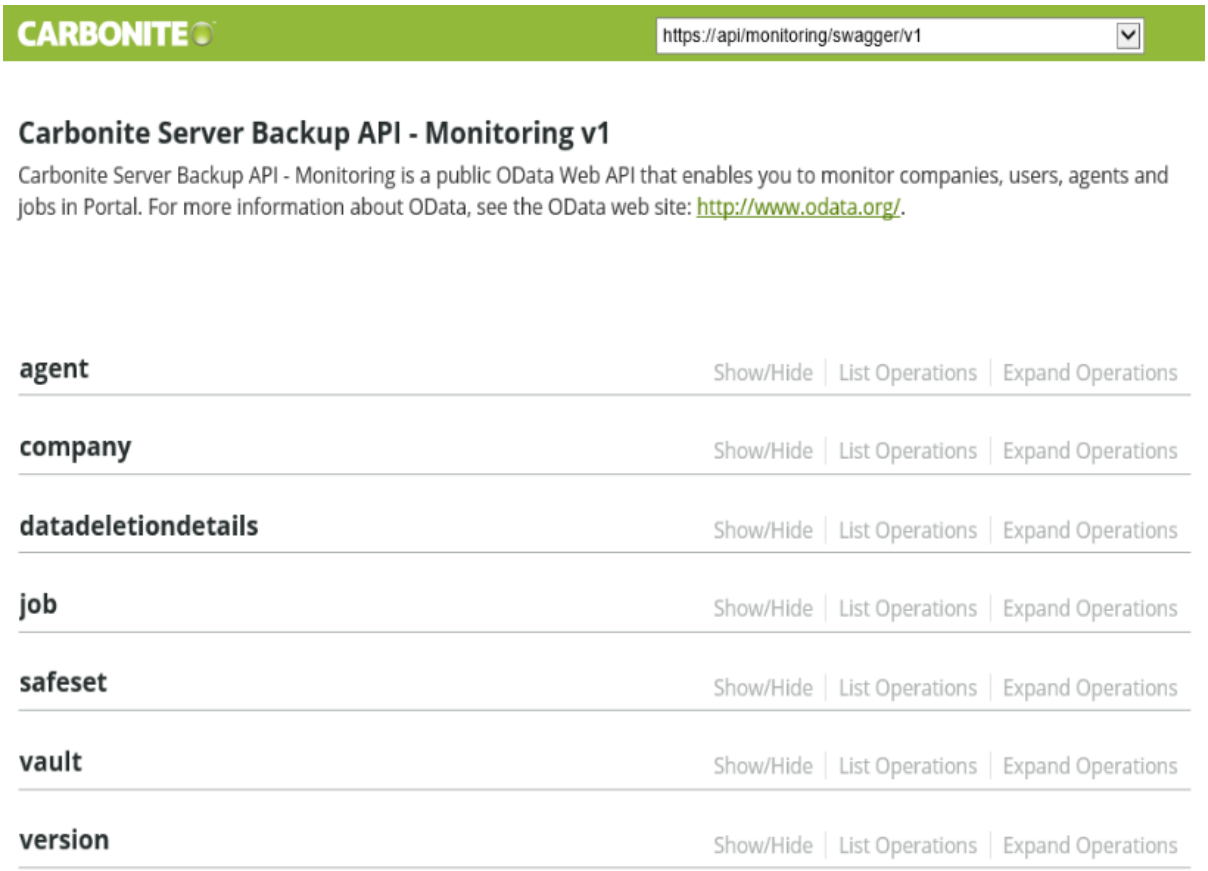
1. In a web browser, go to the Swagger UI for API – Monitoring:

   `https://`*APIdomainNameOrIPaddress*`/monitoring/swaggerui/index`

   Where *APIdomainNameOrIPaddress* is the public domain name or IP address entered in Step 6 of Install API – Monitoring.

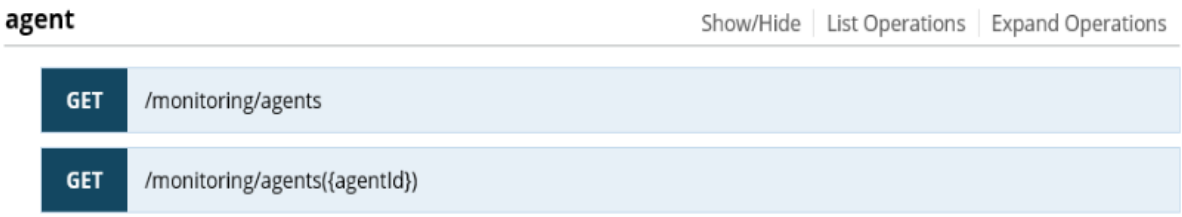
   For example: `https://api.carbonite.com/monitoring/swaggerui/index`

   Note: You might not be able to view the Swagger UI in Internet Explorer 11. If you encounter problems, use a different web browser to access the Swagger UI.

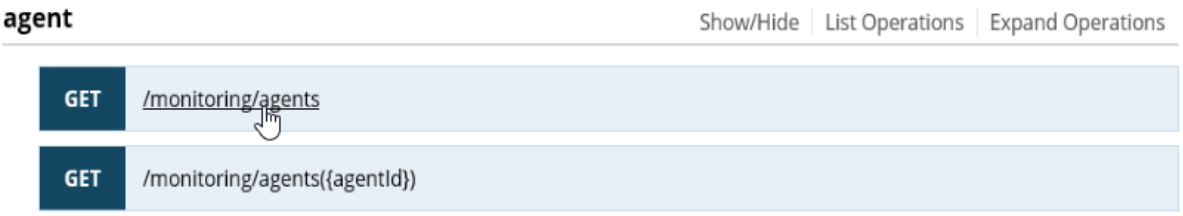


2. To view available calls for an entity (e.g., agent), click the entity name or click **List Operations** for the entity.

Available calls appear below the entity name.

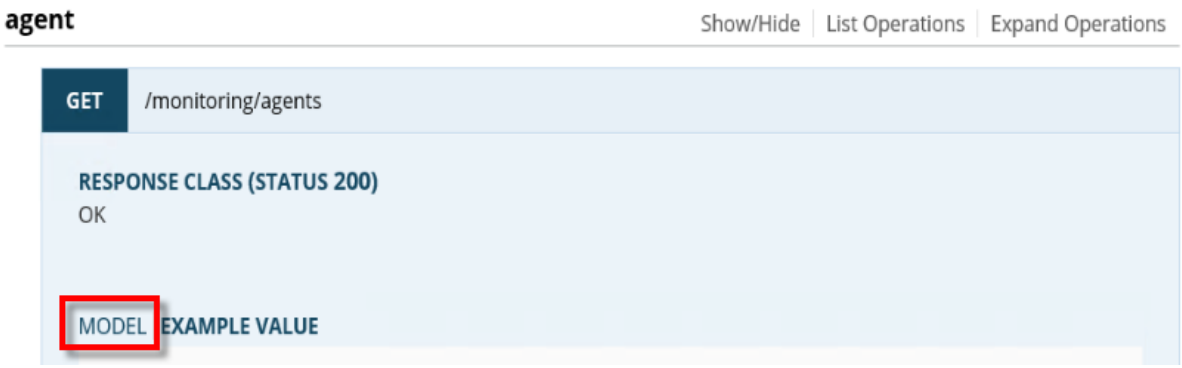


3. To view information about a particular call, click the operation or click **Expand Operations**.

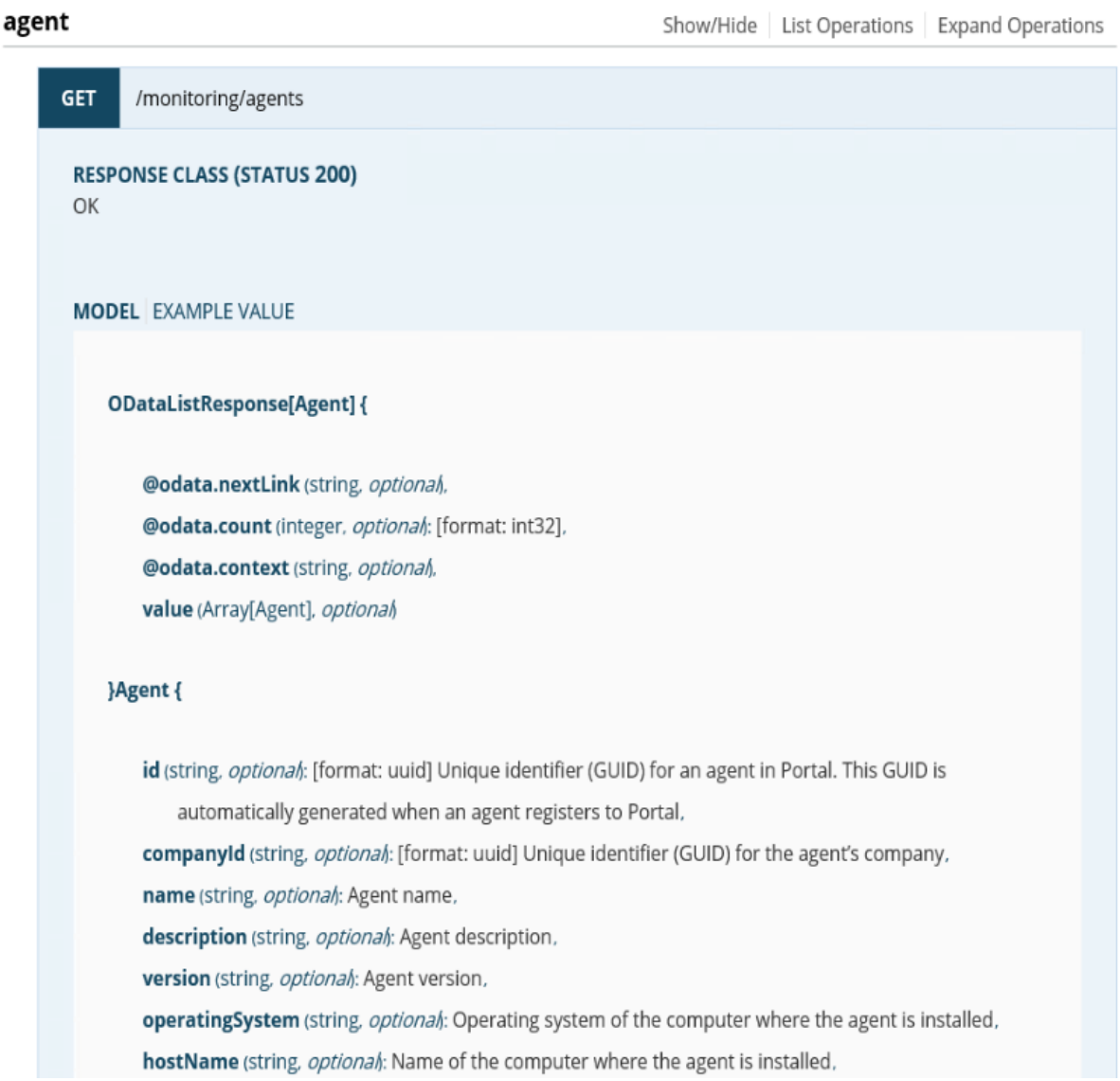


Information about the call appears below the call name.

4. To view detailed information about the call, click **Model**.



Information about the call, including descriptions of each value returned, appears.



agent Show/Hide | List Operations | Expand Operations

GET /monitoring/agents

RESPONSE CLASS (STATUS 200)
OK

MODEL | EXAMPLE VALUE

ODataListResponse[Agent] {

@odata.nextLink (string, *optional*),
@odata.count (integer, *optional*): [format: int32],
@odata.context (string, *optional*),
value (Array[Agent], *optional*)

}Agent {

id (string, *optional*): [format: uuid] Unique identifier (GUID) for an agent in Portal. This GUID is automatically generated when an agent registers to Portal,
companyId (string, *optional*): [format: uuid] Unique identifier (GUID) for the agent's company,
name (string, *optional*): Agent name,
description (string, *optional*): Agent description,
version (string, *optional*): Agent version,
operatingSystem (string, *optional*): Operating system of the computer where the agent is installed,
hostName (string, *optional*): Name of the computer where the agent is installed,

7 Monitor protected data

As described in Monitoring overview, client applications can make read-only calls to API – Monitoring to obtain information about agents, jobs and other entities in Portal and registered vaults. Client applications are authenticated and authorized using Keycloak: the third-party authorization server that is installed with API – Monitoring.

After API – Monitoring is installed and Director Reporting services are registered to the API (see Install API – Monitoring and Register Director Reporting services to the API), you must do the following so that Keycloak can authenticate and authorize client applications:

1. Create a Keycloak admin user. This admin user is required before you can create API – Monitoring clients in Keycloak.
2. Create API – Monitoring clients. API – Monitoring clients must be created in Keycloak so they can be authenticated and authorized when they make calls to the API.
3. Obtain the ID and secret for a client. A Client ID and secret is generated for each client created in Keycloak.

   Note: These Client ID and secret values are client-specific. These values are not the same as the values shown on the last page of the API – Monitoring InstallShield wizard.

Client applications use these Client ID and secret values to obtain access tokens and make API – Monitoring calls. For more information about access tokens, see Obtain an access token. For examples of client applications that retrieve data using API calls, see the API Client sample code that is available from Carbonite.

You can also check that the system is working by testing API – Monitoring calls using the Swagger UI that is installed with API – Monitoring. See Test API – Monitoring calls.

7.1 Create a Keycloak admin user

You must create a Keycloak admin user before you can create clients that obtain data from API – Monitoring. Keycloak is the third-party authorization server that is installed with the API and is used to authenticate and authorize client applications.

To create a Keycloak admin user:

1. On the server where API – Monitoring is installed, open a Powershell window.
2. In the Powershell window, navigate to the following directory: C:\keycloak-3.2.1.Final\scripts
3. Run the following command:

   `.\KeyCloak-AddUser.ps1 -u` *KeycloakAdminUser* `-p` *KeycloakAdminPassword*

   For example, to create an admin user for Keycloak named “keycloakadmin”, you could run the following command:

   ```
   .\KeyCloak-AddUser.ps1 -u keycloakadmin -p strongpassword
   ```

7.2 Create API – Monitoring clients

Before a client application can obtain data from API – Monitoring, you must create the client in Keycloak using a script that is provided with the API. Each client can have one of the following three access types:

- Admin access type. Clients with the Admin access type have unrestricted access to all data in a Portal instance and all vault information. Typically, this client type is used by the service provider that hosts the Portal/Platform API instance. See Create an API client with the Admin access type.
- Partner access type. Clients with the Partner access type have access to all data in a Portal instance but do not have access to vault information. See Create an API client with the Partner access type.
- Reseller access type. Clients with the Reseller access type have access to data for one or more specific companies in a Portal instance, and safesets that belong to agents in these companies. This type of client is equivalent to a Portal Admin user in a parent company. If the client has access to a parent company, it also has access to data for the child companies. See Create an API client with the Reseller access type.

7.2.1 Create an API client with the Admin access type

A client with the Admin access type has unrestricted access to all data in a Portal instance and all vault information. Typically, this client type is used by the service provider that hosts the Portal/Platform API instance.

To create an API client with the Admin access type:

1. On the server where API – Monitoring is installed, open a Powershell window.
2. In the Powershell window, navigate to the following directory: C:\keycloak-3.2.1.Final\scripts
3. Run the following command:

   `.\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final" "https://`*APIdomainNameOrIPaddress*`:8081/auth"` *KeycloakAdminUser KeycloakAdminPassword* `Carbonite-Monitoring` *APIclientName* `urn:carb:sb:api:monitoring admin`

   Where:

   - *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of Install API – Monitoring.
   - *KeycloakAdminUser* and *KeycloakAdminPassword* are the Keycloak admin user name and password created in Create a Keycloak admin user for creating API clients.
   - *APIclientName* is the name of the API client that you want to create.

   For example:

   ```
   .\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final"
   "https://api.carbonite.com:8081/auth" KeycloakAdmin strongpassword
   Carbonite-Monitoring APIadmin urn:carb:sb:api:monitoring admin
   ```

   If you selected the HTTP communication protocol when you installed the API, you can run the command using HTTP. For example:

```
.\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final"
"http://api.carbonite.com:8081/auth" KeycloakAdmin strongpassword Carbonite-
Monitoring APIadmin urn:carb:sb:api:monitoring admi
```

7.2.2 Create an API client with the Partner access type

A client with the Partner access type has access to all data in a Portal instance but does not have access to vault information.

To create an API client with the Partner access type:

1. On the server where API – Monitoring is installed, open a Powershell window.
2. In the Powershell window, navigate to the following directory: C:\keycloak-3.2.1.Final\scripts
3. Run the following command:

   `.\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final" "https://`*APIdomainNameOrIPaddress*`:8081/auth" `*KeycloakAdminUser KeycloakAdminPassword* `Carbonite-Monitoring `*APIclientName* `urn:carb:sb:api:monitoring partner`

   Where:

   - *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of Install API – Monitoring.
   - *KeycloakAdminUser* and *KeycloakAdminPassword* are the Keycloak admin user name and password created in Create a Keycloak admin user.
   - *APIclientName* is the name of the API client that you want to create.

   For example:

   ```
   .\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final"
   "https://api.carbonite.com:8081/auth" KeycloakAdmin strongpassword
   Carbonite-Monitoring Partner1 urn:carb:sb:api:monitoring partner
   ```

   If you selected the HTTP communication protocol when you installed the API, you can run the command using HTTP. For example:

   ```
   .\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final"
   "http://api.carbonite.com:8081/auth" KeycloakAdmin strongpassword Carbonite-
   Monitoring Partner1 urn:carb:sb:api:monitoring partner
   ```

7.2.3 Create an API client with the Reseller access type

A client with the Reseller access type has access to data for one or more specific companies in a Portal instance but does not have access to vault information. This type of client is equivalent to a Portal Admin user in a parent company. If the client has access to a parent company, it also has access to data for the child companies.

To create an API client with the Reseller access type:

1. On the server where API – Monitoring is installed, open a Powershell window.
2. In the Powershell window, navigate to the following directory: C:\keycloak-3.2.1.Final\scripts

3. Run the following command:

   `.\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final" "https://`*APIdomainNameOrIPaddress*`:8081/auth" `*KeycloakAdminUser KeycloakAdminPassword*` Carbonite-Monitoring `*APIclientName*` urn:carb:sb:api:monitoring reseller `*companyGUID*

   Where:

   - *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of Install API – Monitoring.
   - *KeycloakAdminUser* and *KeycloakAdminPassword* are the Keycloak admin user name and password created in Create a Keycloak admin user.
   - *APIclientName* is the name of the API client that you want to create.
   - *companyGUID* is the globally unique identifier (GUID) of the company for which the API client can access data. To obtain the company GUID for a client, see Obtain the reseller's Company ID from Portal.

   For example:

   ```
   .\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final" "https://api.carbonite.com:8081/auth" keycloakadmin strongpassword Carbonite-Monitoring Site1Client urn:carb:sb:api:monitoring reseller 7a877d8c-b20b-4f70-984d3a990719
   ```

   If you selected the HTTP communication protocol when you installed the API, you can run the command using HTTP. For example:

   ```
   .\KeyCloak-CreateMonitoringClient.ps1 "C:\keycloak-3.2.1.Final" "http://api.carbonite.com:8081/auth" keycloakadmin strongpassword Carbonite-Monitoring Site1Client urn:carb:sb:api:monitoring reseller 7a877d8c-b20b-4f70-984d3a990719
   ```

7.2.3.1 Obtain the reseller's Company ID from Portal

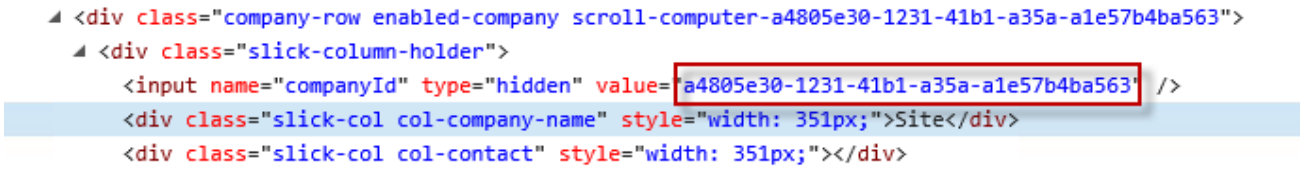
To obtain the reseller's Company ID from Portal:

1. In a web browser, go to the Portal website. Sign in to Portal as a super user.
2. Click **Sites** on the navigation bar.

   Note: A "site" in Portal is the same as a "company" in API – Monitoring.
3. Find the reseller's site. To find the site, you can enter criteria in the filter row under the column headings.

4. Right-click the reseller's site, and choose **Inspect** (in Google Chrome) or **Inspect Element** (in Firefox or Internet Explorer) from the menu.

The source code for the site appears. The companyId appears in the line before the company (site) name.



5. Record the companyId value (a GUID).

7.3 Obtain the ID and secret for a client

Before a client can access data using the API, the authorization server administrator must provide the customer's client ID and secret values. The client can then obtain a JWT access token from the authorization server. See Obtain an access token.

Note: The client ID and secret provided through this process are not the same as the client ID and secret values shown on the last page of the API – Monitoring InstallShield wizard. The values in this procedure are client-specific.

To obtain the ID and secret for a client:

1. In a web browser, go to the KeyCloak administration console:

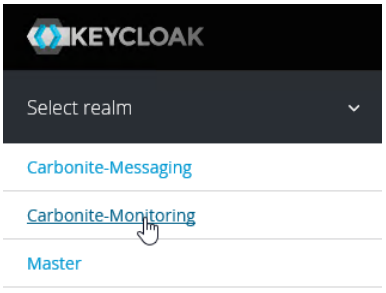
   `https://`*APIdomainNameOrIPaddress*`:8081/auth/admin`

   Where *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of Install API – Monitoring.

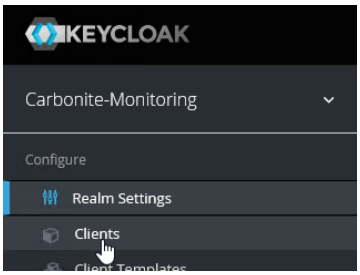
   For example: `https://api.carbonite.com:8081/auth/admin`

2. Log in to the KeyCloak administration console with the Keycloak admin user that you created in Create a Keycloak admin user.

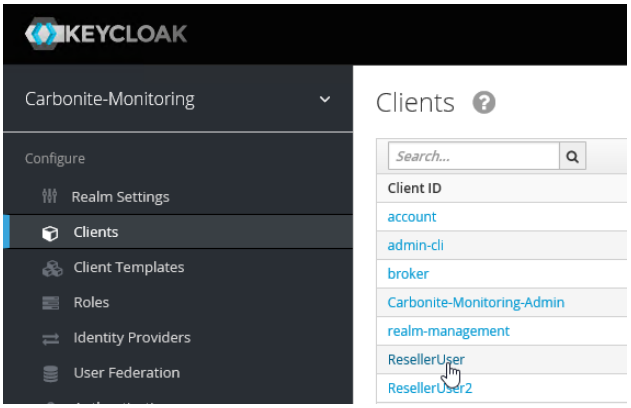
3. Point to **Select realm** in the top left corner, and select the **Carbonite-Monitoring** realm from the list.



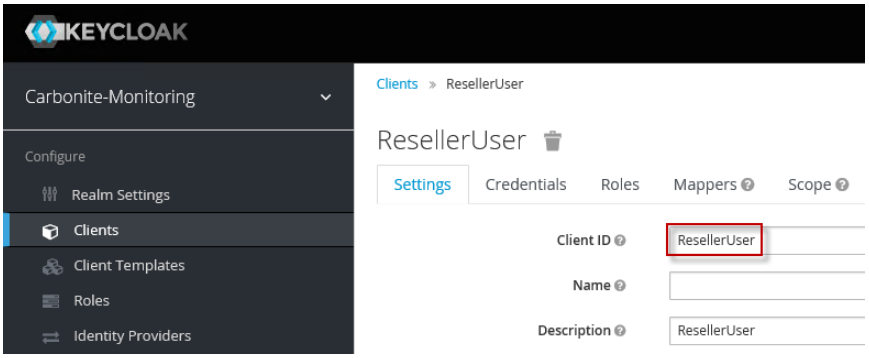
4. Select **Clients** from the menu on the left.



5. Click the client's Client ID link.

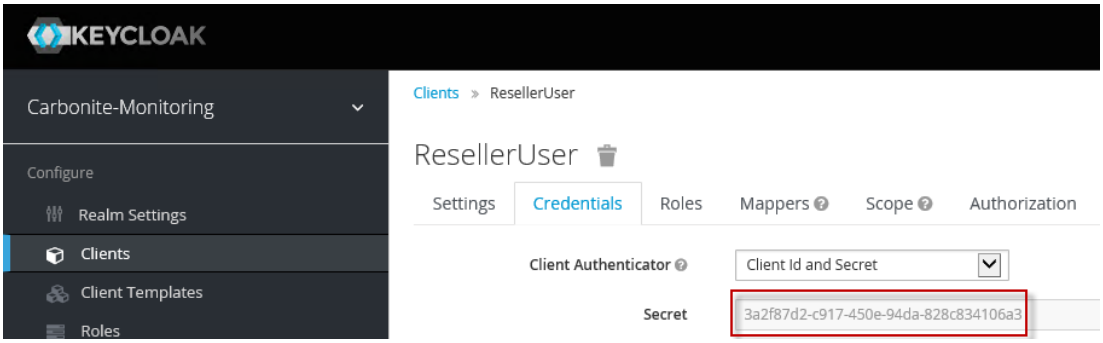


6. Go to the Settings tab. Copy the Client ID value and provide it to the customer.



7. Go to the Credentials tab. Copy the Secret value and provide it to the customer.

   Important: If the Credentials tab is hidden, enter a description for the client in the **Description** field, and then click **Save** at the bottom of the page. The credentials tab will then appear.



7.4 Test API – Monitoring calls

This section describes how to test API – Monitoring calls using the Swagger UI that is installed with the API.

For examples of client applications that retrieve data using API calls, see the API Client sample code that is available from Carbonite.

An authorization parameter, or access token, is required for each API – Monitoring call. This access token is a JSON Web Token (JWT) that specifies the data that the client can access. See Obtain an access token.

Each API – Monitoring call can return a maximum of 500 records. To obtain more than 500 records, and for best performance, use query options in API calls. See Best practice: Use query options in API – Monitoring calls. For more information about OData, see the OData website: http://www.odata.org

To test API – Monitoring calls:

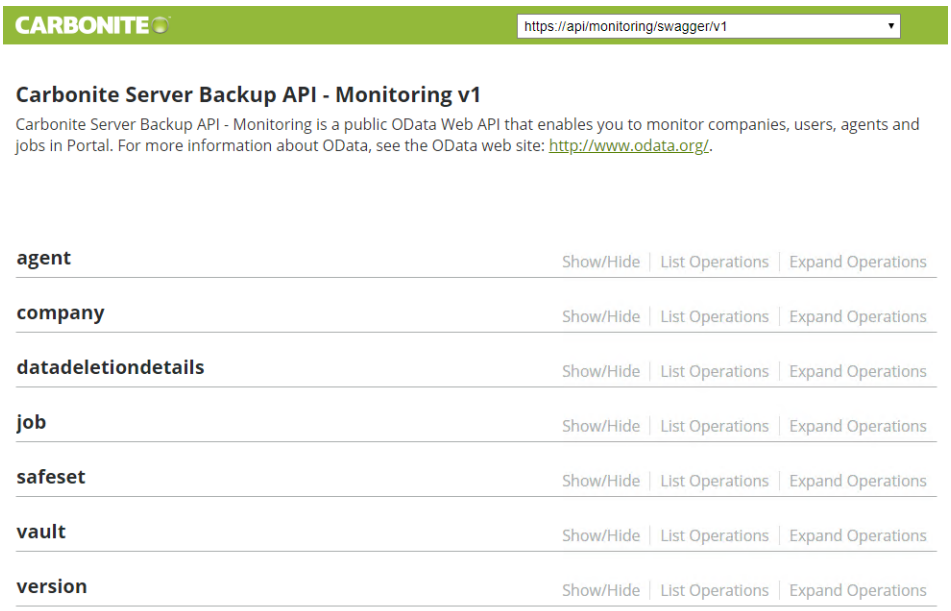
1. In a web browser, go to the Swagger UI for API – Monitoring:

   `https://APIdomainNameOrIPaddress/monitoring/swaggerui/index`

   Where *APIdomainNameOrIPaddress* is the public domain name or IP address entered in Step 6 of Install API – Monitoring.

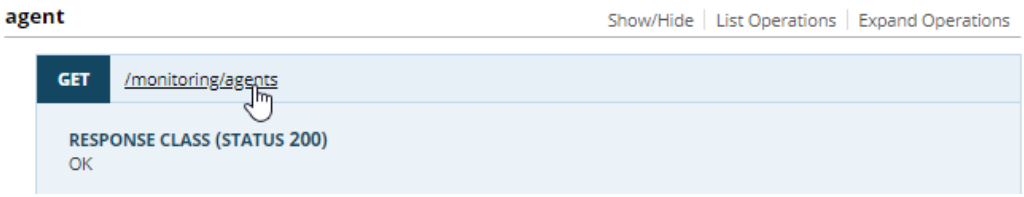
   For example: `https://api.carbonite.com/monitoring/swaggerui/index`

   Note: You may not be able to view the Swagger UI in Internet Explorer 11. If you encounter problems, use a different web browser to access the Swagger UI.



2. Click the entity (e.g., agent) for which you want to make a call, or click **List Operations** for the entity.

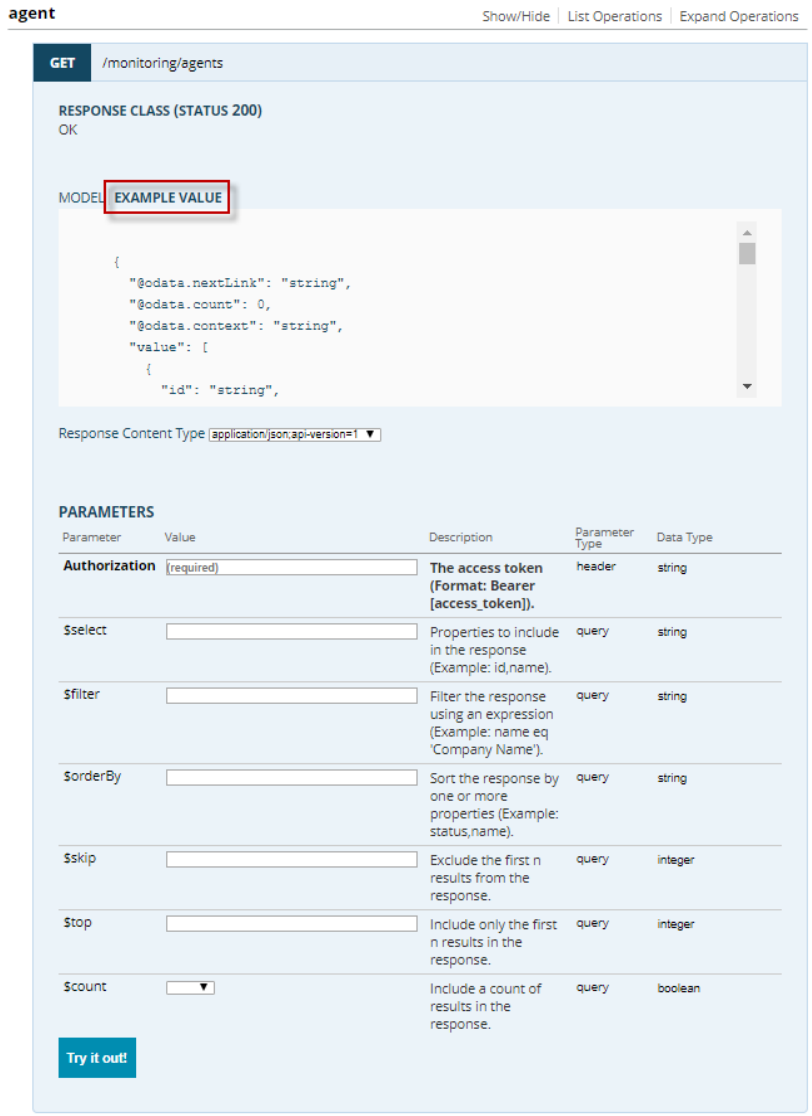
3. Click the call that you want to test or click **Expand Operations** for the entity.



4. To test the call, click **Example Value**. Enter parameters in the fields, and then click **Try it out!**

   You must enter an authorization parameter (access token) for each API – Monitoring call. To obtain an access token, submit a post request to the authorization server using information from your authorization server administrator. See Obtain an access token.

   In general, strings must be enclosed in single quotation marks in filters. GUID, Boolean, int and enum values do not need to be enclosed in quotation marks in filters. For more information, see OData documentation.



7.4.1 Obtain an access token

An authorization parameter, or access token, is required for each API – Monitoring call. This token is a JSON Web Token (JWT) that specifies the data that the client can access.

To obtain an access token:

1. Contact your authorization server administrator for the following required information:
   - URL of the authorization server endpoint
   - Client ID
   - Client Secret

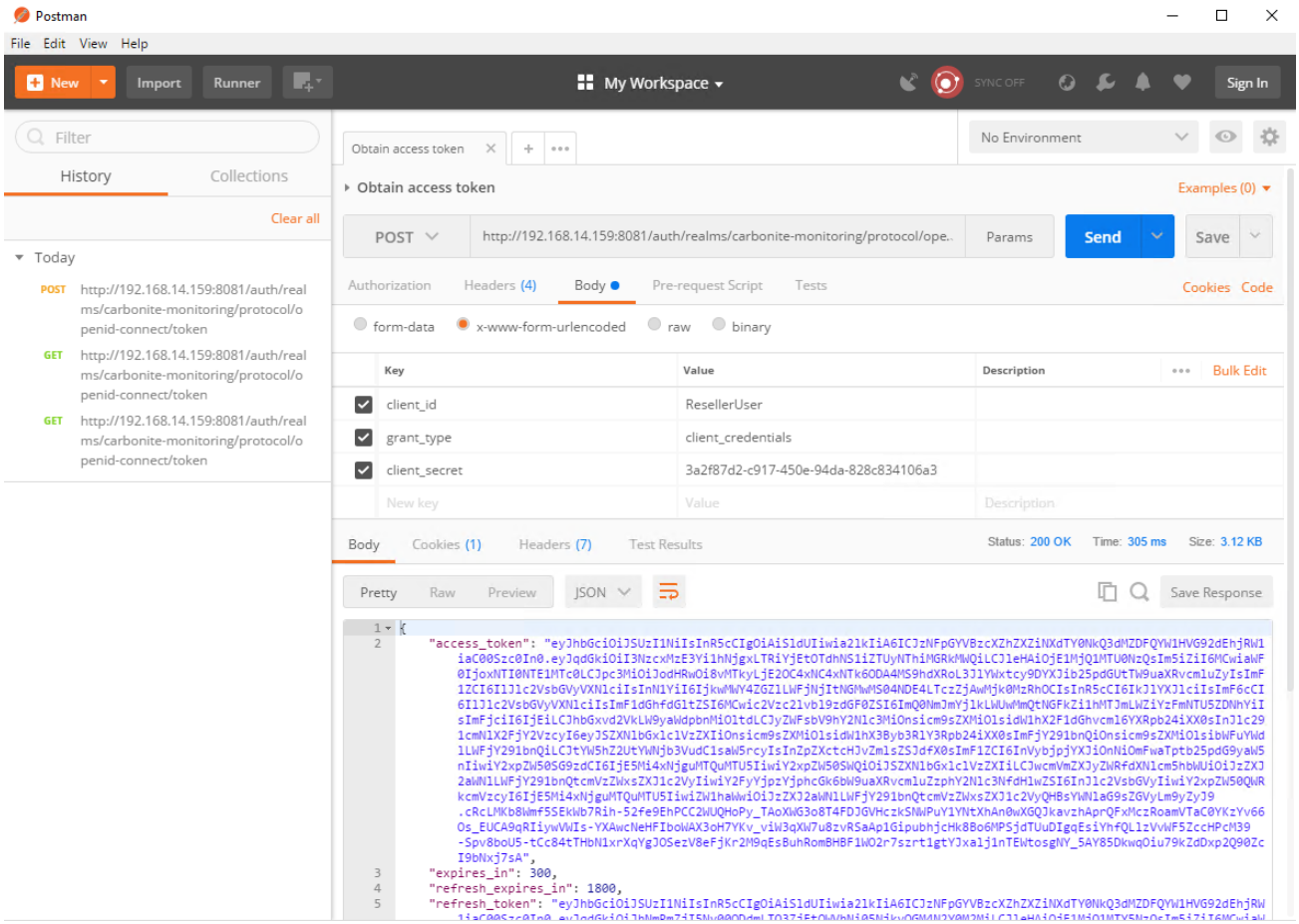
   The administrator obtains the Client ID and Secret values as described in Obtain the ID and secret for a client.
2. To obtain an access token, submit a post request using a REST client, as described in the following table:

Method:	**POST**

URL of the Keycloak endpoint	`https://`*APIdomainNameOrIPaddress*`:8081/auth/realms/carbonite-monitoring/protocol/openid-connect/token`
	Where *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of Install API – Monitoring. For example:
	`https://api.carbonite.com:8081/auth/realms/carbonite-monitoring/protocol/openid-connect/token`

Body type:	`x-www-form-urlencoded`
Form fields	
client_id	*clientID*
grant_type	`client_credentials`
client_secret	*clientSecret*

The following screenshot shows a post request in Postman (www.getpostman.com).



7.4.2 Best practice: Use query options in API – Monitoring calls

For best performance, use query options to reduce the number of records or properties returned in API calls. For example, if you only require specific information for some agents, use query options to specify the agents and properties instead of obtaining data for all agents.

You can use the following query options to reduce the number of records or properties in API calls:

- $select – Specifies a subset of properties to return.
- $filter – Specifies an expression that must be true for a record to be returned.
- $top – Specifies the maximum number of records to return.
- $skip – Specifies the number of records to skip before returning records.

For performance reasons, each API – Monitoring call can return a maximum of 500 records. To determine whether there are more than 500 results for a call, include the $count query option. You can then use additional calls with the $skip query option to return the remaining results. For example, if the $count query option indicates that there are 1422 agents in your environment, you could use one call to obtain

the first 500 agent records, a second call with the $skip query option to skip the first 500 agents, and a third call with the $skip query option to skip the first 1000 agents.

For more information about OData query options, see the OData website: http://www.odata.org

Examples

To obtain status information for agents from a specific company, you could use the following call:

```
GET
APIdomainNameOrIPaddress/monitoring/agents?$select=name,operatingSystem,stat
us&$filter=companyId eq 980abfd9-5ddd-4bf3-b9f2-cc8bbd10ce0e
```

Where *APIdomainNameOrIPaddress* is the API domain name or IP address entered in Step 6 of Install API – Monitoring.

To return only the top 50 agent records and determine how many results there are for the call, use the following call:

```
GET APIdomainNameOrIPaddress/monitoring/agents?$top=50&$count=true
```

To skip the first 50 agents and return the next 50 agent records, use the following call:

```
GET APIdomainNameOrIPaddress/monitoring/agents?$top=50&$skip=50
```

To skip the first 500 agents and return agents 501 to 550, use the following call:

```
GET APIdomainNameOrIPaddress/monitoring/agents?$top=50&$skip=500
```

8 Set up data deletion

As described in Data deletion overview, when Portal and Director vaults are registered to the same API – Monitoring instance, Director can delete job and computer data in response to requests from Portal.

After API – Monitoring is installed and Director Reporting services are registered to the API (see Install API – Monitoring and Register Director Reporting services to the API), the following steps must be completed so that Director can delete vault data in response to requests from Portal:

1. Portal must be registered to the same API – Monitoring instance as Director Reporting services, and the data deletion feature must be enabled in Portal. See “Enable data deletion” in the *Portal Administration Guide* or online help.
2. A machine key must be added in Portal config files. See “Add a machine key required for data deletion” in the *Portal Installation Guide*.
3. Data deletion email notifications must be set up in Portal. See “Set up automatic emails” in the *Portal Installation Guide*.
4. The Data Deletion feature must be enabled in Portal. See “Enable data deletion” in the *Portal Administration Guide* or online help.