# CA4K™ “RESTful API”

## Installation Guide

Revision: Rev A

Date: 1/27/2022

# DISCLAIMER

Continental Instruments LLC makes no representations or warranties with respect to the contents hereof and specifically disclaim any implied warranties of merchantability or fitness for any particular purpose. Further, Continental Instruments LLC reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Continental Instruments LLC to notify any person of such revision or changes.

**Copyright © 2022 by Continental Instruments LLC.** All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, or stored in a retrieval system, without the prior written permission of Continental Instruments LLC, 355 Bayview Avenue, Amityville, NY 11701. Telephone: 631-842-9400 • FAX: 631-842-9135

This document contains proprietary information of NAPCO Security Technologies. Unauthorized reproduction of any portion of this manual without the written authorization of NAPCO Security Technologies is prohibited. The information in this manual is for informational purposes only. It is subject to change without notice. Companies, names and data used in examples herein are fictitious unless otherwise noted. NAPCO Security Technologies assumes no responsibility for incorrect information this manual may contain.

*A NAPCO SECURITY TECHNOLOGIES COMPANY*
Publicly traded on NASDAQ Symbol: NSSC

Visit our websites:
http://www.cicaccess.com/
http://www.napcosecurity.com/
http://www.alarmlock.com/

# *Table of Contents*

***Must Read Before Installation*** ..... *4*

***Overview*** ..... *5*

***Prerequisites*** ..... *5*

***System Requirements*** ..... *5*

***Related Documents*** ..... *6*

***Prerequisites*** ..... *6*

***Quick Start Steps*** ..... *7*

***IIS Installation in Window 10*** ..... *8*

***IIS Installation in Window Server 2012 R2*** ..... *10*

***IIS Installation in Windows Server 2019*** ..... *16*

***CA4K RESTful API Installation*** ..... *17*

***RESTful API Programming Tips*** ..... *21*

## Must Read before you begin installation

- To use the RESTful API, you must be an experienced Web developer.

- Prior to installing the RESTful API, you must install Microsoft Internet Information Services (IIS). IIS is included in the Windows operating system.

- The installation of IIS and the RESTful API require Full Administrator privileges without any restrictions. This includes lowering the UAC settings to "Never Notify" and verifying there are no domain restrictions.

- The RESTful API is a wrapper for the Windows .Net API.

- The Restful API is language-neutral.

- By default, the RESTful API will be setup to use **port 8081**.

- The RESTful API demo code is written in C# and Javascript. The demo code can be found on the installation media, in the same folders as the Setup.exe files.

- You can use the POSTman testing utility to test the API functions.

## Overview

This document provides a step by step procedure for installing Microsoft Internet Information Services (IIS) for use with the CA4K™ RESTful API. Prior to starting the installation, you must verify the computer meets the recommended computer specifications and you have the CA4K™ DVD (Installation media) and the correct CA4K™ software license with the API Enabled.

## System Requirements

Ensure Recommended Server Requirements are met. Verify that the computer meets the specifications that are documented in the **CA4K™ System Requirements** (Computer Specifications) document.

## Verify Full Administrator Permissions

The installation of CA4K™ and IIS requires Full Administrator permissions. Failure to provide these permissions will result in installation failures.

## Related Documents

**CA4K™ TCP/IP Ports Reference Guide –** TCP/IP Ports document

**CA4K™ System Requirements –** Computer Specifications

**CA4K™ Complete Install** - Installation Guide

**ASP.Net RESTful API Help Page**

## Prerequisites

- A Functional CA4K™ Complete Server with V1.1.16 SP-3 or V1.1.35 SP-1 and later.

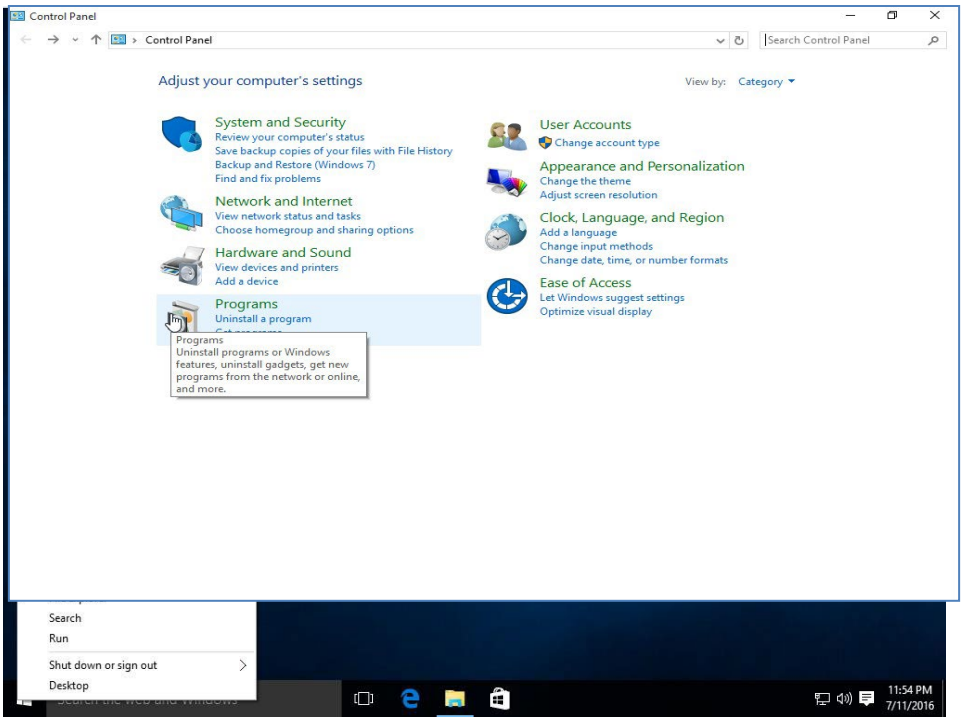
# Quick Start Steps

1. Purchase a CA4K, version 1.1.x Software license with the API Enabled.
2. Perform a CA4K Complete Server installation (V1.1.16 SP-3 or V1.1.35 SP1 or later.
3. Install Windows Microsoft Internet Information Services (IIS). After IIS is installed, verify the Default Website displays.
4. Install the RESTful API from the CA4K Installation Media, using the setup.exe program.
5. Verify the CA4K RestfulAPI folder was created in the CardAccess4K folder.
6. Verify the CA4K RESTfulAPI Website displays in IIS, and is started.
7. Refer to the RESTful API demo code if needed.
8. Use Postman test utility to do testing on the RESTful API.

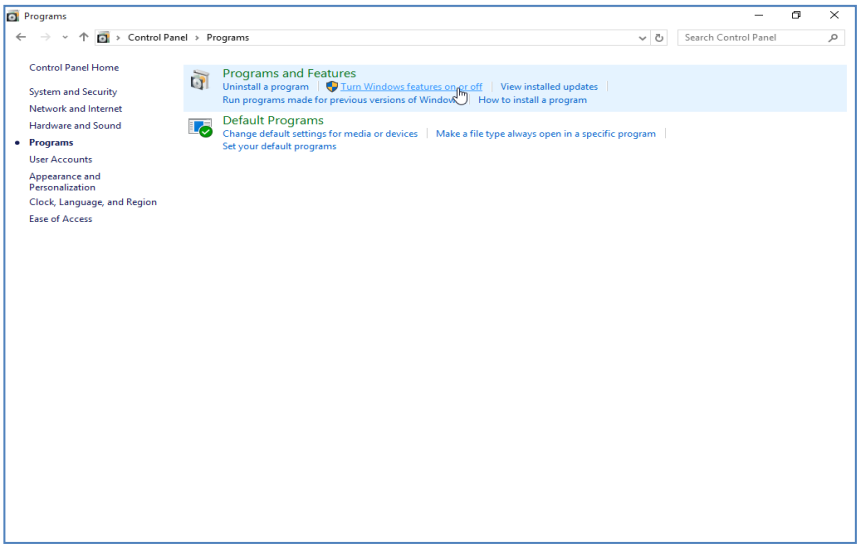
# IIS Installation in Windows 10

Note: Prior to installing the CA4K RESTful API, you must install Internet Information Services (IIS) Server. The following demonstrations are for IIS in Windows 10 and Server 2012 R2.

1. Open **Control Panel** from the Start Menu.



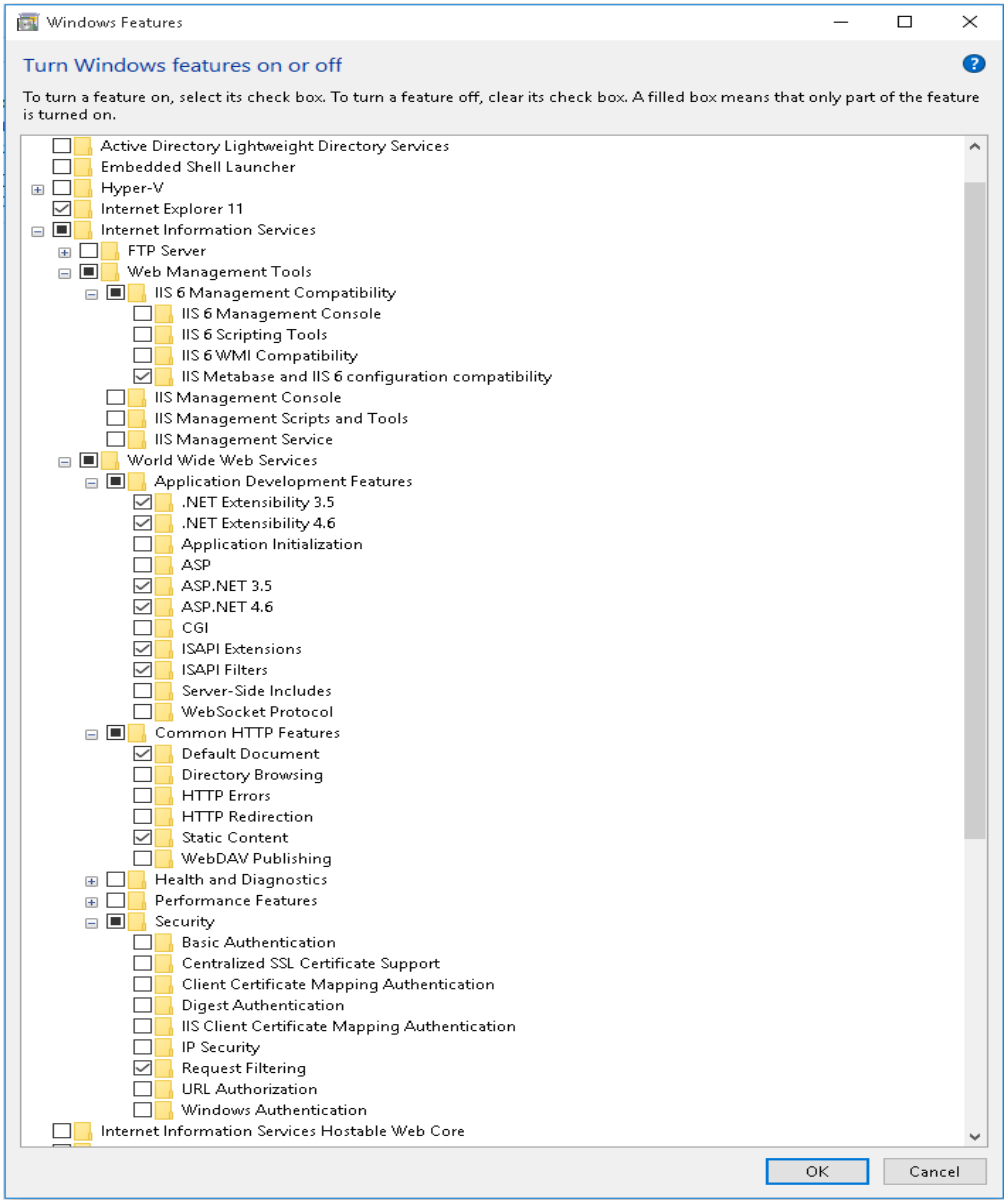

2. In the **Control Panel**, click **Programs.** In the **Programs and Features** section, select **Turn Windows features on or off** link.



3. Now, click on the features that are checked on the following screens and then click the **OK** button.

**Notes:**

- It is recommended to install IIS Manager Tool. Select **IIS Management Console** under **Web Management Tools.**
- The IIS selection screens will vary between operating systems. The following IIS selection screen is for Windows 10. The selections screens for Server 2012 R2 and Server 2019 can be found later in the document.
- Always select the latest .NET versions available on the selections screen.



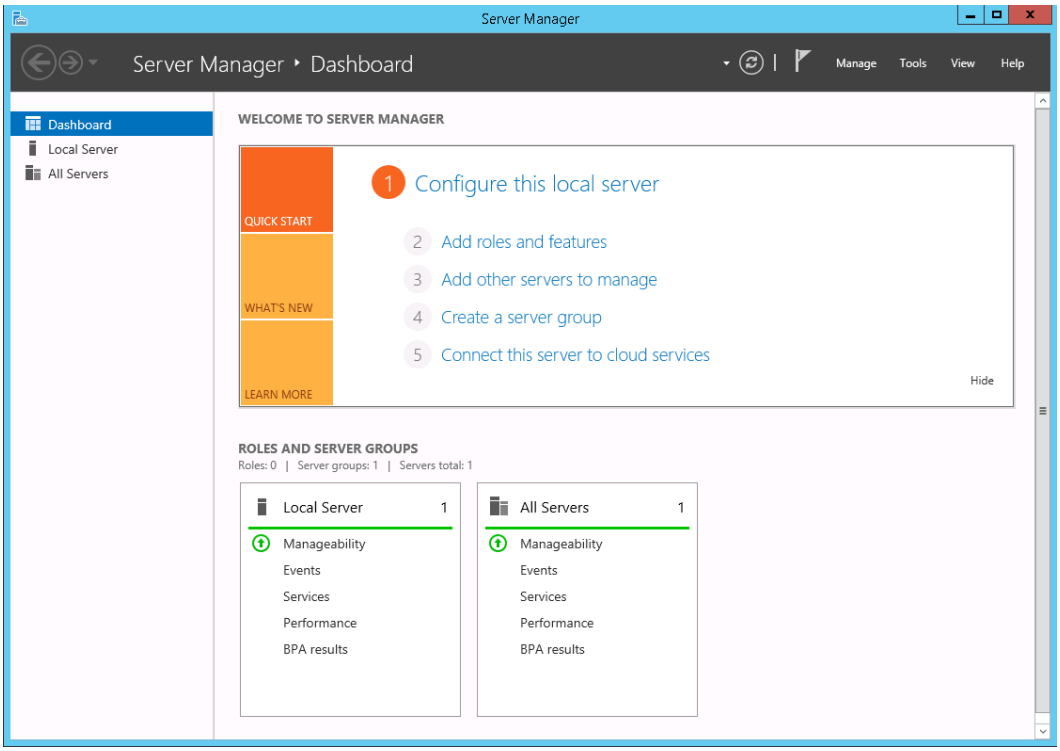

Note: After IIS is successfully installed, you must install the RESTful API.

# IIS Installation on Windows Server 2012 R2

Note: Prior to installing the CA4K RESTful API, you must install Internet Information Services (IIS) Server. The following demonstration is for Windows Server 2012 R2.

1. Open **Server Manager** from the start page.

2. In Server Manager, click on **Add roles and features** under the **Server Manager** menu.

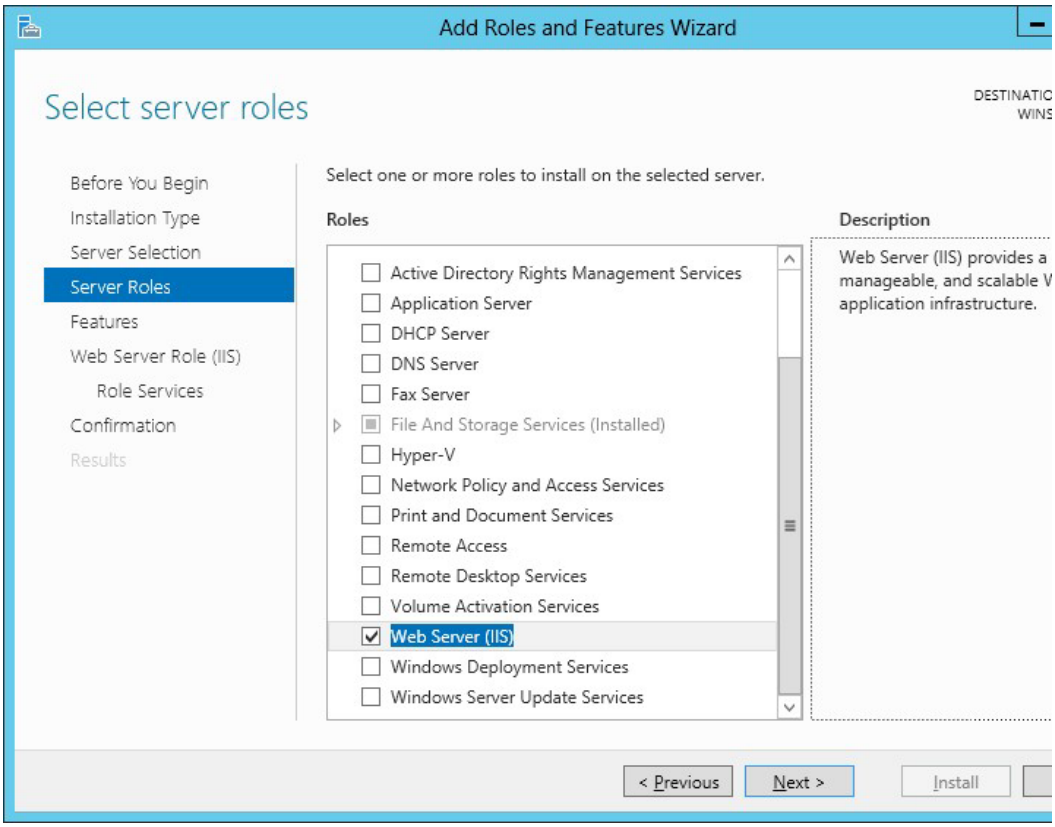



3. Under **Add Roles and Features Wizard**, on the **Before you begin** page, click **Next**.

4. On the **Select installation type** page, select **Role-based or feature-based installation** radio button, and click **Next.**

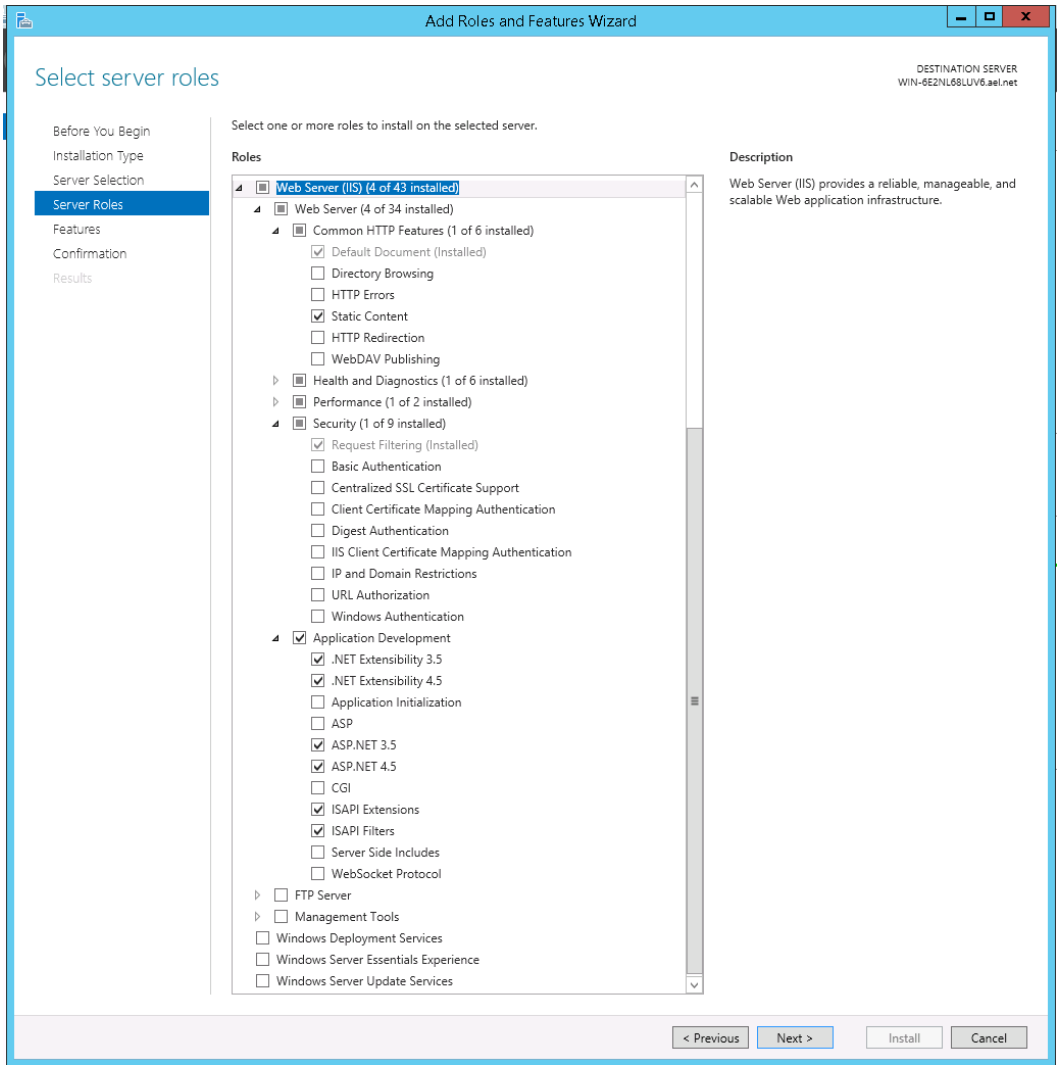
5. On the **Select destination server** page, **Select a server from the server pool** radio button, and click **Next.**

6. Select **Web Server (IIS)** and then click on the button **Next.**

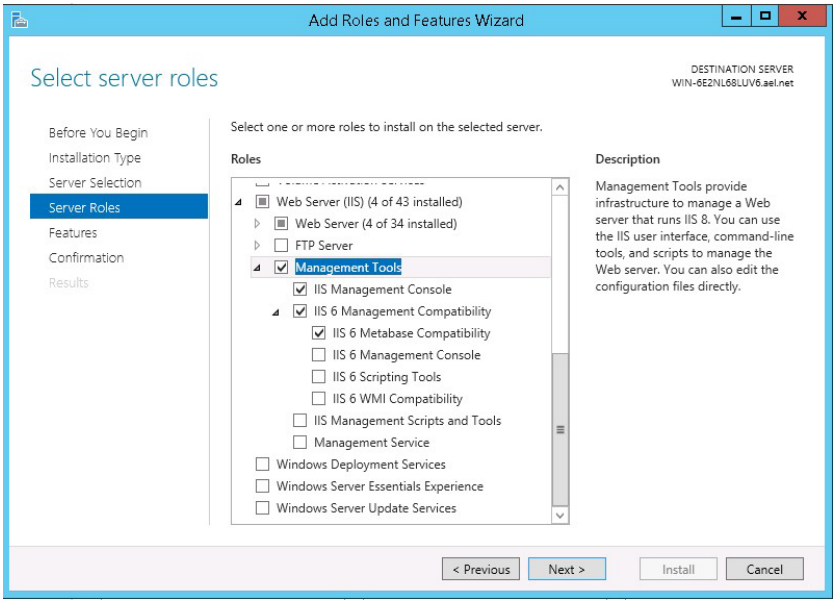



7. In the **Select features** page, no additional features are needed for IIS. Click **Next** to continue.

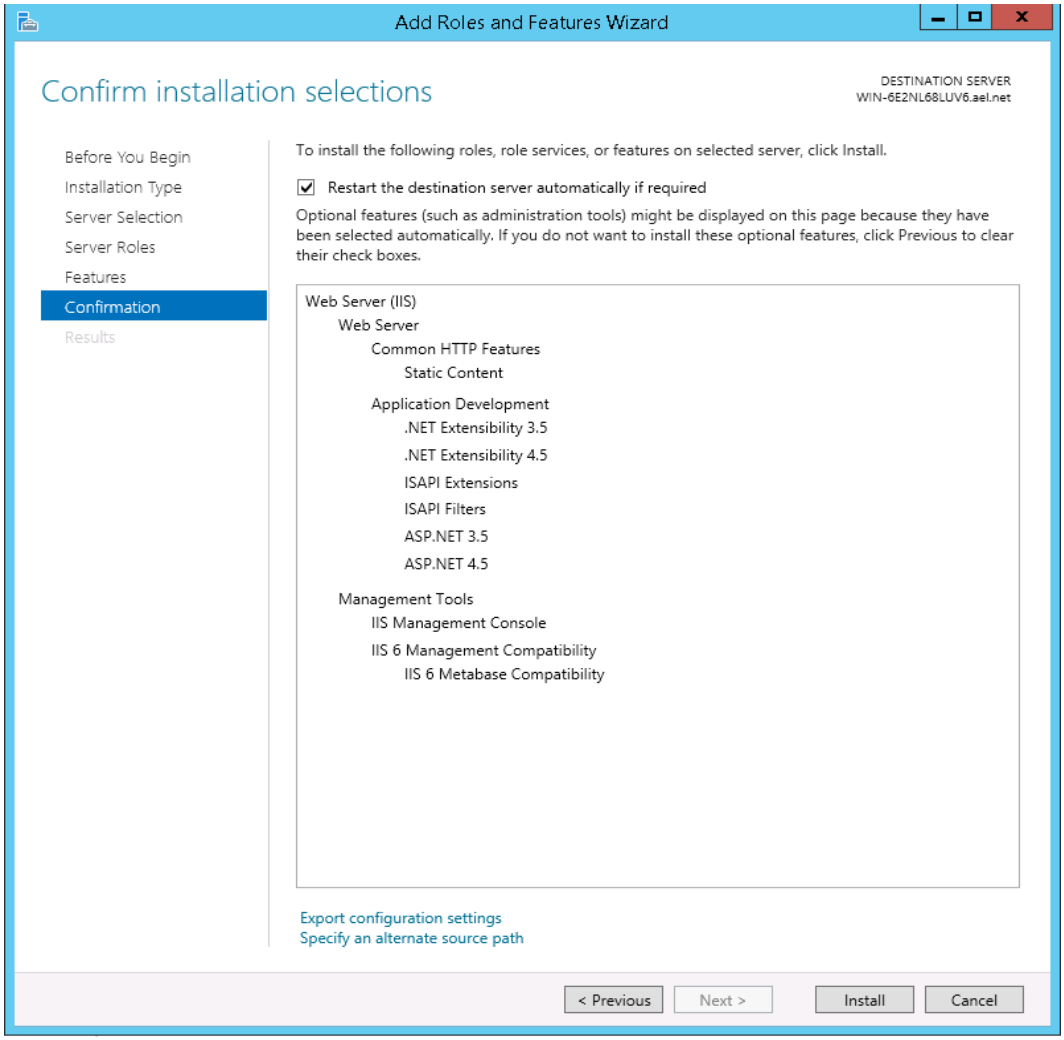
8. The next window is the **Select Role Services**. This very important screen will allow you to add only the modules necessary for your planned installation of IIS. Now, click on the role services that are checked on the Following screens.




9. Under **Management Tools** section, the following items must be checked (IIS Management Console and IIS 6 Metabase Compatibility).


Add Roles and Features Wizard
Select server roles
DESTINATION SERVER
WIN-6E2NL68LUV6.ael.net
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
Web Server (IIS) (4 of 43 installed)
Web Server (4 of 34 installed)
FTP Server
Management Tools
IIS Management Console
IIS 6 Management Compatibility
IIS 6 Metabase Compatibility
IIS 6 Management Console
IIS 6 Scripting Tools
IIS 6 WMI Compatibility
IIS Management Scripts and Tools
Management Service
Windows Deployment Services
Windows Server Essentials Experience
Windows Server Update Services
Description
Management Tools provide infrastructure to manage a Web server that runs IIS 8. You can use the IIS user interface, command-line tools, and scripts to manage the Web server. You can also edit the configuration files directly.
< Previous
Next >
Install
Cancel

10. Click **Next** to get to the **Confirm installation selections** screen to verify your chosen settings and then click **Install**.

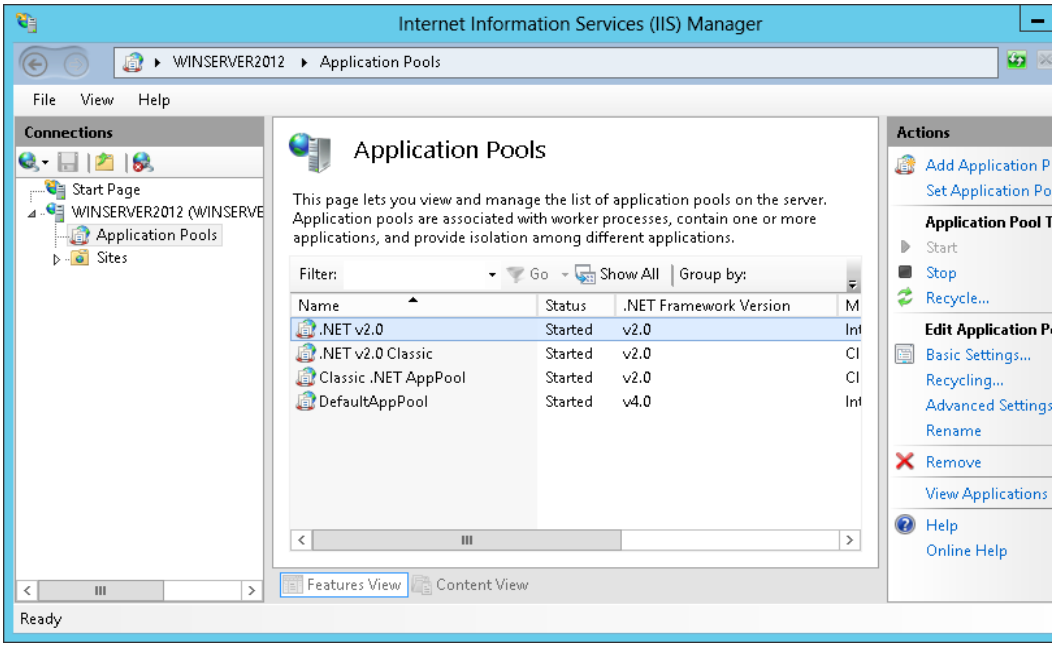



11. After installation succeeded you should see the **Installation progress** page. Click **Close** to finish the IIS Installation process.
12. You can verify that IIS installed successfully, type the following into a web browser:
    http://localhost

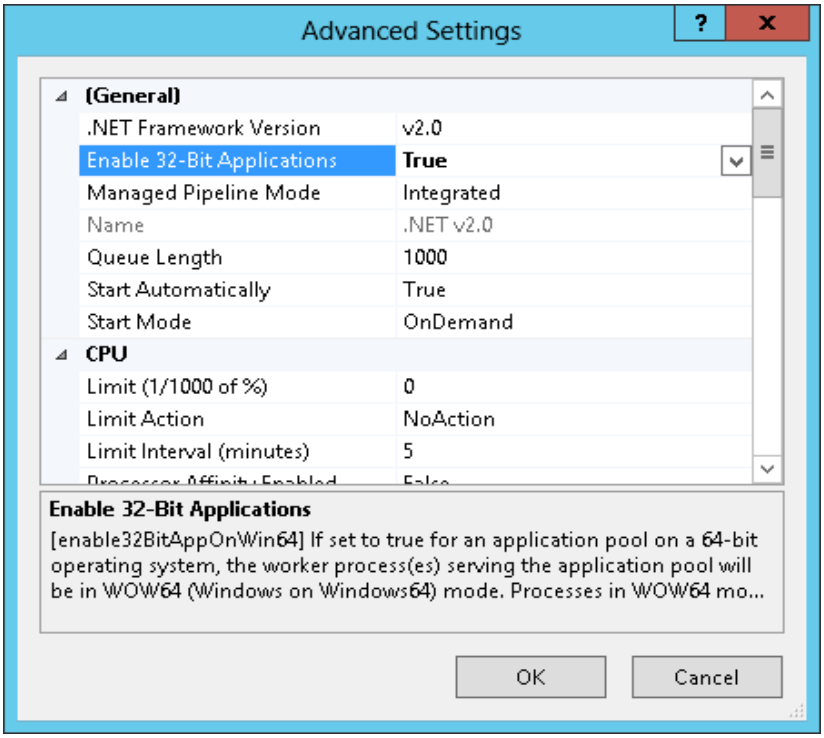
***[Steps (13-16) are only for 64 Bit OS]***

13. Open **Internet Information Services (IIS) Manager** again (top-right corner of Server Manager window).

14. Go to Current Server -> Application Pools.

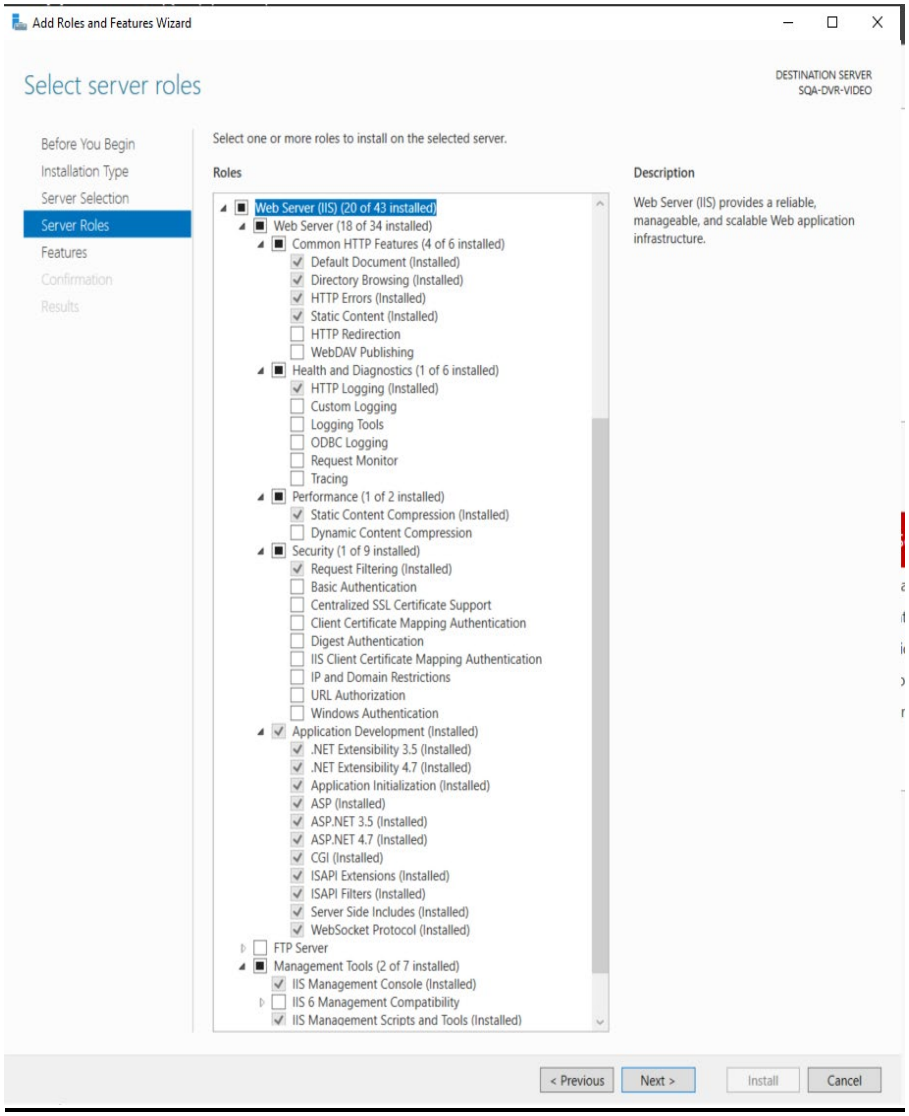



15. Select the application pool your 32-bit application will run under CA4K RESTful API and click **Advanced Settings** on **Actions** pane.



# IIS Installation on Windows Server 2019

Note: The following IIS Server Roles selection screen is for Windows Server 2019. The IIS setup will be very similar to the Server 2012 R2 steps above. In addition to the steps above, use the Server Roles settings screen below to configure all the Server Roles.



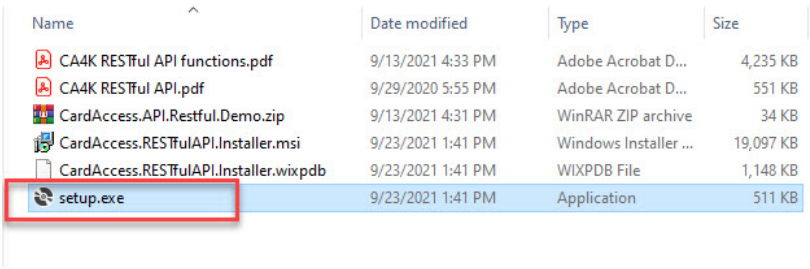

Note: After IIS is successfully installed, you must install the RESTful API (Refer to the following steps)

# CA4K RESTful API Installation

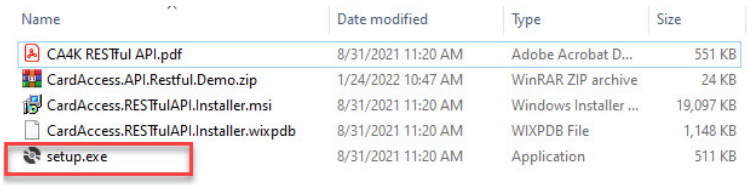
**Note:** The Setup files for the RESTful API are on the CA4K installation media. The setup files are in different locations in CA4K V1.1.16 SP-3 and CA4K V1.1.35 SP-1. It is very important to also install the service packs for each version.

1. Perform one of the following steps depending on the version of CA4K you are using.

- **V1.1.16 SP-3** Go to the **CardAccess4K>RESTfulAPI** folder on the installation media, and open it. Click on **setup.exe** as per the screenshot below.

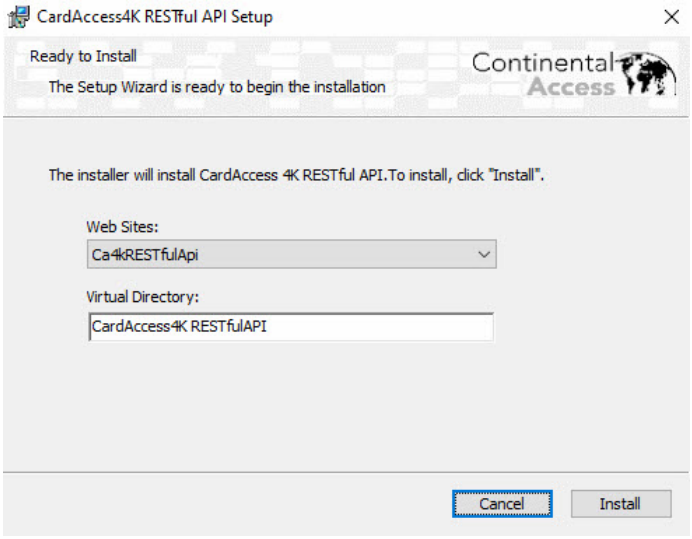



- **V1.1.35 SP-1 and later**-Go to the **CardAccess4K>API>RESTfulAPI** folder on the installation media, and open it. Click on **setup.exe** as per the screenshot below.

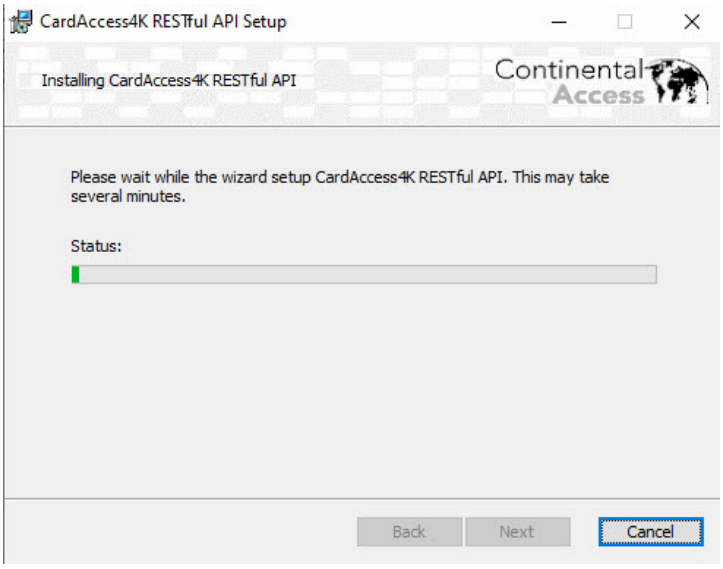



Note: The **RESTful API Demo code** can also be found in the above two locations.

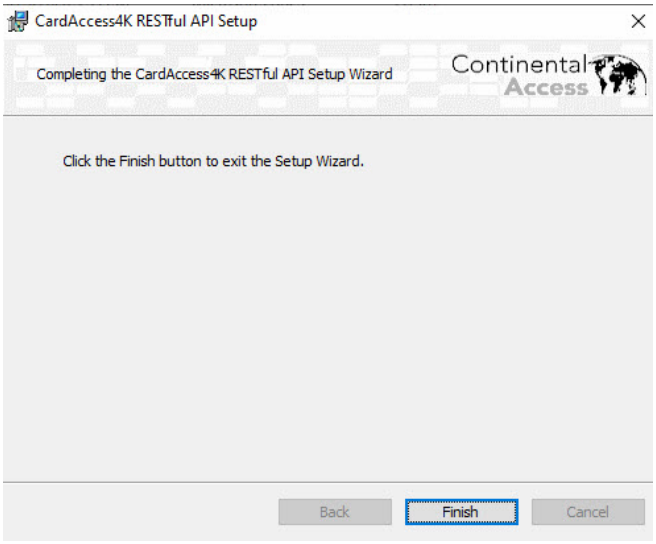
2. Upon launching the setup.exe, the **Ready to Install** screen will display. For our demonstration, leave the **Web Site name** and the **Virtual Directory** name at the defaults. Click **Install**.



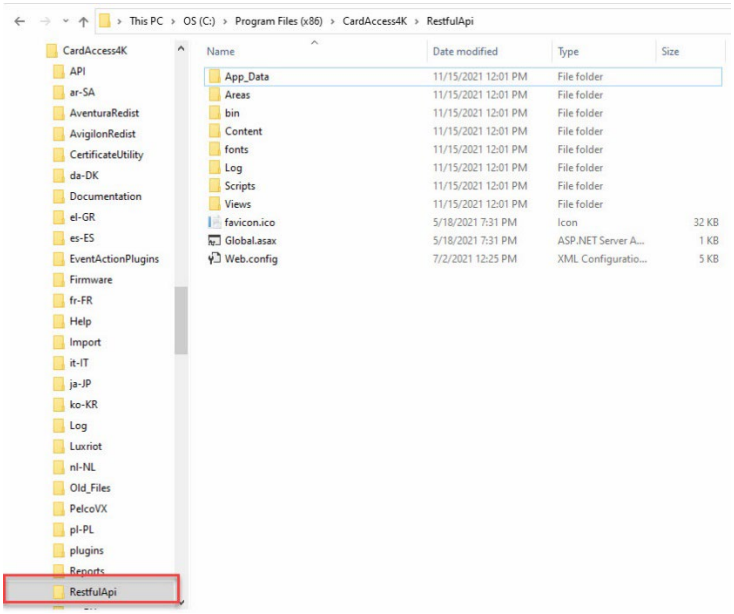

3. Upon clicking Install, a progress bar will display while setup.exe is installing files.



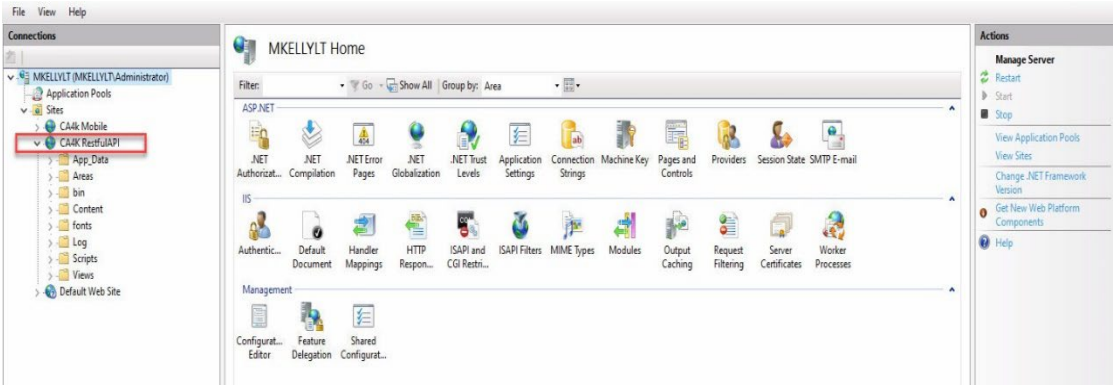
4. Upon the installation completing, click **Finish**.




5. After the RESTful API installation is complete, verify the RESTfulAPI folder was created under the CardAccess4K folder.



6. In addition to the RestfulApi folder being created in the CardAccess4K folder, launch IIS. In IIS, verify the **CA4K RestfulAPI site** is displaying. Right click on the **CA4K RestfulAPI site**, click **Manage Website**, and verify the site is Running.



# RESTful API Programming Tips

Note: As previously mentioned, you must be an experienced Web developer to use the RESTful API and the Demo code. The CA4K RESTful API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. This API was built as a wrapper WebAPI from .NET CardAccess.API and returns mostly JSON data. The RESTful API demo code is written in C# and Javascript. It can be found in the same folders as the Setup.exe files. The Demo Code zip file name is **CardAccess.API.Restful.Demo.zip**.

1. Once the RESTful API is installed, the IIS website name will be **CA4K RESTfulAPI**.

1. By default, the RESTful API will be setup to use **port 8081**.

2. All functions use **Authorization**.

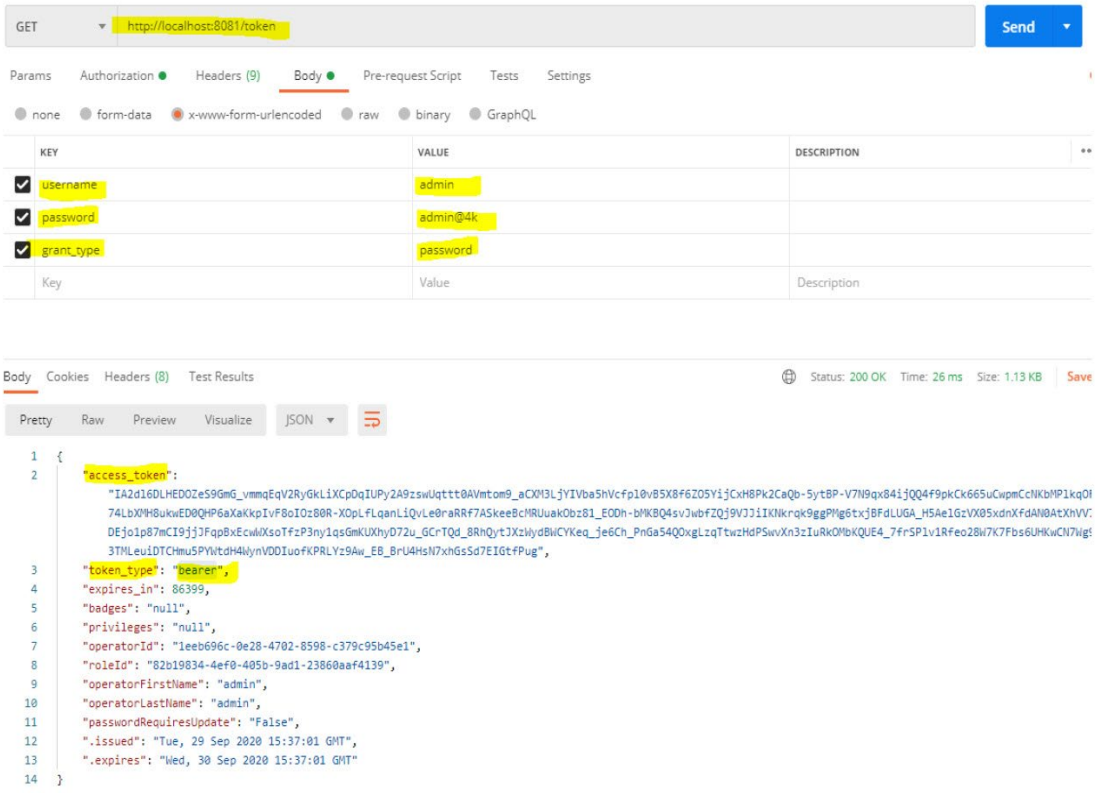
4. First get the **access token** by calling the url http://<server ip>:8081/token
   a. Body keys
      i. Username <- ca4k operator name
      ii. Password <- ca4k operator password
      iii. Grant_type = password

5. Use the access_token returned and token type = bearer to call all other functions.

6. **Example 1- (To Get badgeinfo)**
   a. http://localhost:8081/Badges/BadgeInfo?badge=1234&facility=0
   b. Body keys
      i. Authorization = bearer XXXXXXXXXXXXXXXXXX… , is the access_token return from token request call.

7. **Example 2** – (**Register to receive events using signalr).**

```
@section scripts{
   @*test signalr*@
   @Scripts.Render("~/bundles/signalR")
  <script src="/signalr/hubs"></script>
   <script>
```

```
$(function () {
    var alertHubService = $.connection.alertHubService;
     $.connection.hub.start().done(()
     =>alertHubService.server.connect());
    alertHubService.client.alertReceived = function (sde) {
    console.log(sde);
    };
    alertHubService.client.hcsStatusReceived = function (message) {
    console.log(message);
    };
  });
</script>
```

## Example using POSTMAN to get Token