# Allianz REST Course Exercises

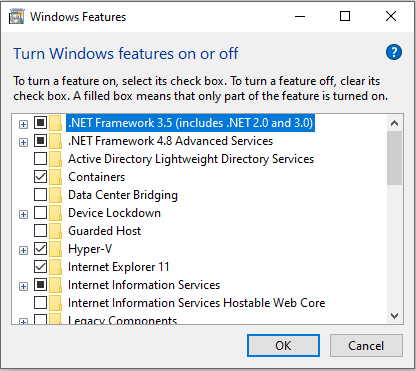
## Exercise 1a - Set up IIS

In this exercise, you will set up IIS on your computer, ready to use with JADE REST services.

If you already have IIS set up on your computer, great! Just go through and double-check everything looks like it does in the pictures.

### Enabling Windows Features.

* Search for **Turn Windows Features on or off** in windows search.



Within the Windows Features dialog:

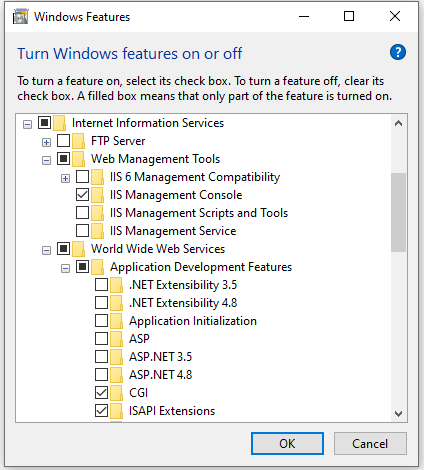
* Navigate through and tick **Internet Information Services -> Web Management Tools -> IIS Management Console**.

*This will enable the IIS Manager which is a GUI for editing all the IIS settings.*

* Navigate through and tick **Internet Information Services -> World Wide Web Services -> Application Development Features -> CGI.**
* From the same location, also tick **ISAPI Extensions.**

*(These allow IIS to use a Dynamic Link Library file (dll) to handle web requests. In our case, that will be jadehttp.dll.)*

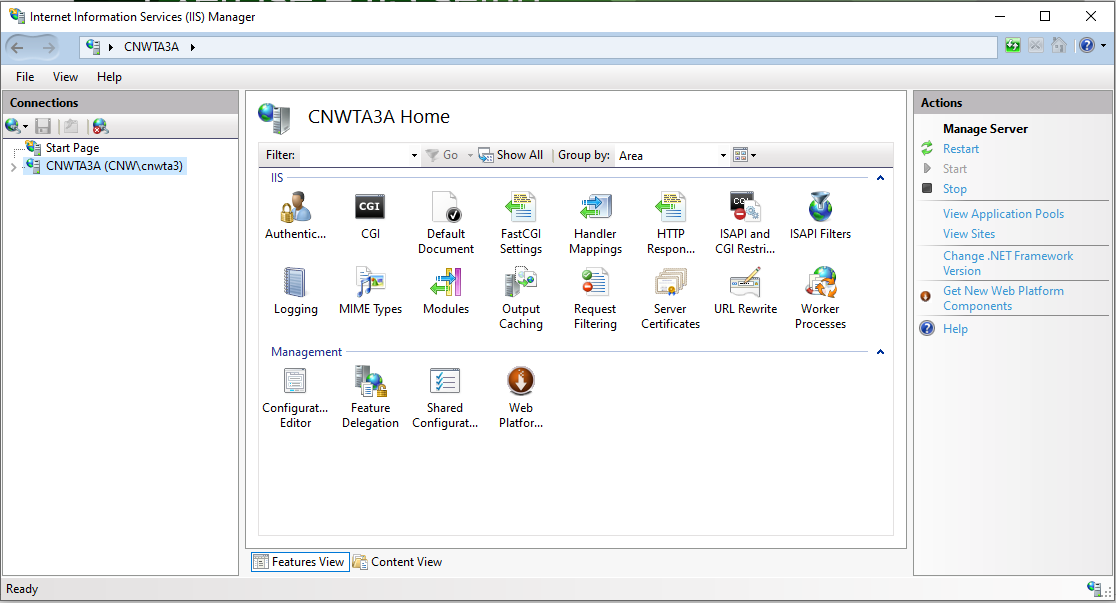
It should look something like this:



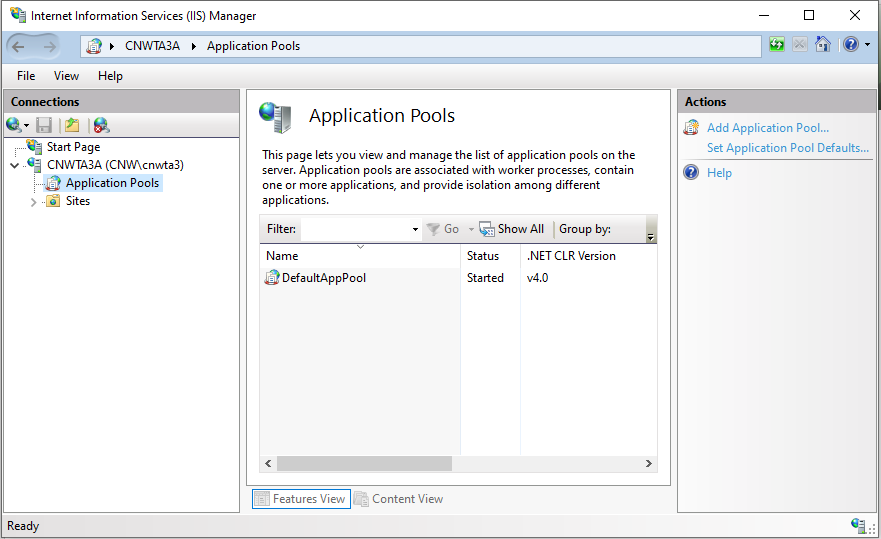
Now that the features are enabled, we can use them to configure IIS:

* Search for IIS Manager in Windows search.

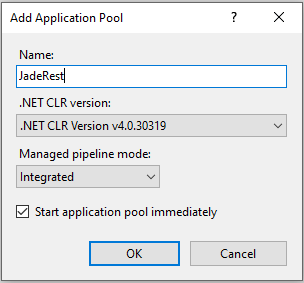
You will see something like this, though you may have different icons in the middle (and your computer will have a different name than me 😊)



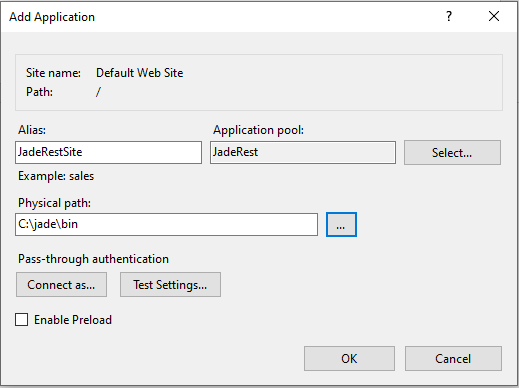
* On the left-most pane, click on your computer name to expand it, then **Application Pools**.



* On the rightmost pane, click **Add Application Pool…**



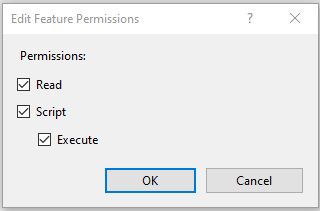
* Call it something sensible like **JadeRest** and click OK.
* Back to the left-hand pane, expand Sites and right-click on **Default Web Site**, selecting **Add Application** from the context menu.
  + For **Alias**, enter **JadeRestSite**
  + For **Application pool**, press the **Select** button and select the pool you made.
  + For **Physical Path**, select the **bin** directory of your JADE install.



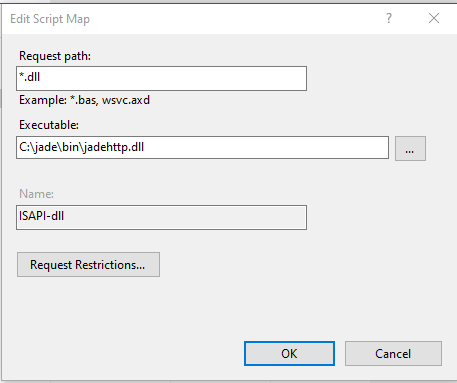
* From the left-hand pane, select the **JadeRestSite** application you just created.

There are two things we need to do here; we need to enable CGI-exe to execute code (like the code in **jadehttp.dll**) and we need to configure ISAPI-dll to know to use **jadehttp.dll**.

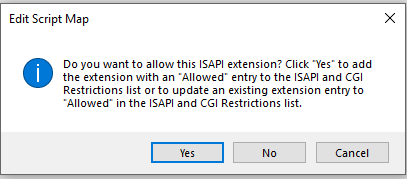
* Double-click on **Handler Mappings** in the center pane.
* Right-click **CGI-exe** from the center pane, and from the context menu select **Edit Feature Permissions**.
* Tick **Execute** and press OK.



* Double-click **ISAPI-dll**.
* For the **Executable**, select the **jadehttp.dll** from your JADE install’s **bin** directory.

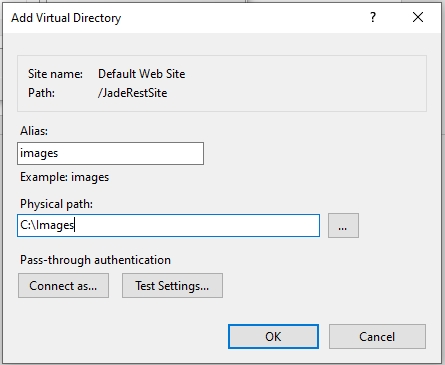


* Click **Yes** to the popup:



Finally, we need to set up a virtual directory.

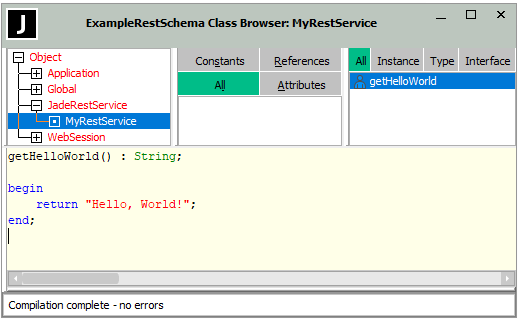
* From the leftmost pane, right-click **JadeRestSite** and from the context menu select **Add Virtual Directory**.
* For **Alias**, enter **images**.
* For **Physical path**, select (or create) a convenient, empty directory on your filesystem, e.g. **c:\images**.



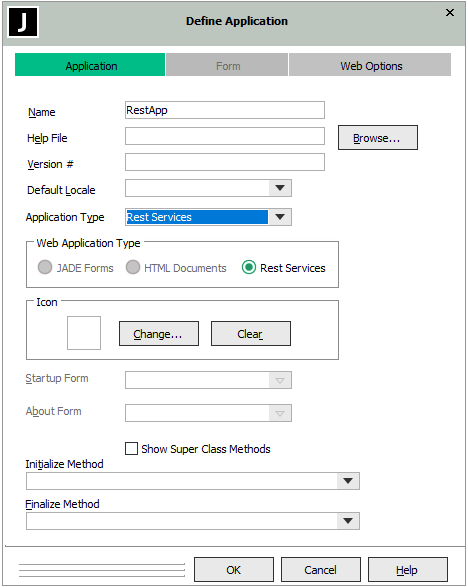
## Exercise 1b – JADE REST API Setup.

In this exercise, you will create a \*very\* simple REST API in JADE so we can make sure that everything is wired up correctly.

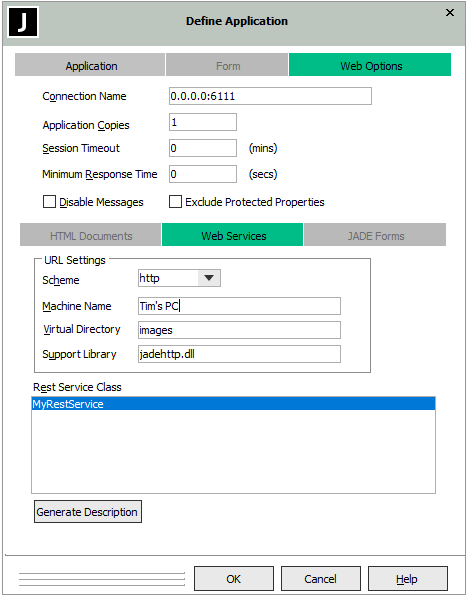
* Open up your JADE, create a new Schema and find the **JadeRestService** class (e.g. with F4)
* Subclass it with a user class, call it whatever you want (but I’ll refer to it as **MyRestService**).
* To the **MyRestService** class, add a new method called **getHelloWorld** which returns a string, “Hello, World!”.



* Open the Applications browser, and add a new application:



* Set the **Name** to **RestApp**
* Set the **Application Type** to **Rest Services**.
* Click the **Web Options** tab:



* For the **Connection Name**, enter **0.0.0.0:6111**

**Tip:** The **Connection Name** is expecting an IP address and a port. 0.0.0.0 means “I don’t care what the client thinks my IP address is.” You could put localhost or 127.0.0.1 for loopback, which is what we’ll be using for the exercises, but then it would \*only\* work for loopback traffic. The **6111** for the port number doesn’t matter so long as it’s consistent with what you set later and so long as it’s not already used on your machine.

* For the **Machine Name**, enterwhatever you want. You probably should put your actual machine name, but it doesn’t matter.
* For the **Virtual Directory**, enter the name of the Virtual Directory you created in IIS, e.g., **images**.
* For the **Rest Service Class**, select the class you created earlier, e.g., **MyRestService**.
* Click OK.

You now have a JADE REST Service ready to accept requests, and an IIS set up to forward the requests to **jadehttp.dll**, the last link we need to wire up is the connection between **jadehttp.dll** and your REST app.

## Exercise 1c – jadehttp.ini setup

In this exercise, you will configure the **jadehttp.ini** file to forward IIS requests to your JADE application.

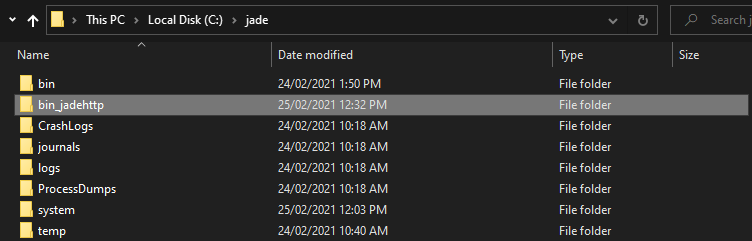
When **jadehttp.dll** gets a request, it looks for **jadehttp.ini** to see what port to forward the request to, and what application name it should be expecting to be listening on that port. We don’t yet have a **jadehttp.ini** though. There’s an easy way and a hard way to get one. We could create it ourselves, but let’s be lazy and do it the easy way.

* In your web browser, go to <http://localhost/JadeRestSite/jadehttp.dll/HelloWorld?RestApp>

You should see the following error:



This is what we want, see how it mentions a JadeHttp ini file? We have one of those now. Navigate to your JADE install directory and you should see a new folder that wasn’t there before.



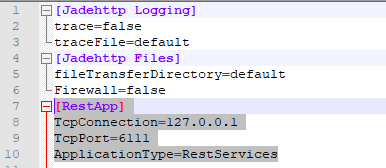
* Within this folder, there is a subfolder **ini**, and within that there is a file **jadehttp.ini**. Open that file.
* Add the following text to the file.

[RestApp]

TcpConnection=127.0.0.1

TcpPort=6111

ApplicationType=RestServices



* Save the file and try refreshing the <http://localhost/JadeRestSite/jadehttp.dll/HelloWorld?RestApp> in the web browser.

It should now work, and show the following:



## Exercise 2 – A more complex REST API

In this exercise, you will add new REST API methods to an existing (Erewhon) REST API while working around “gotchas”. Two files have been provided to make testing this exercise, **ErewhonClient.html** and **ErewhonAgent.html**.

Both are JavaScript REST Clients, where the **ErewhonClient.html** calls existing methods so you can see how it should go, and the **ErewhonAgent.html** should fail all its calls until you implement the new methods.

First, we’ll load the Erewhon System:

* From the MS Teams Channel for this meeting, download **Allianz-REST-Erewhon.rar** if you haven’t already.
* Extract it, then load the **ErewhonInvestments.mul** file.
* From **ErewhonInvestmentsModelSchema**, run the **initializeData** JadeScript method and give it the **DataFiles** folder from your Erewhon directory.

Next, we’ll add the Erewhon REST App to your **jadehttp.ini** file.

* Navigate back to your ***<JADE\_Install\_Directory>*\bin\_jadehttp\ini\jadehttp.ini** file and add the following to the end of the file:

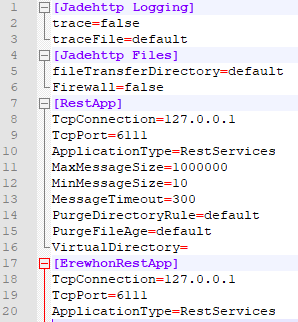
[ErewhonRestApp]

TcpConnection=127.0.0.1

TcpPort=6111

ApplicationType=RestServices

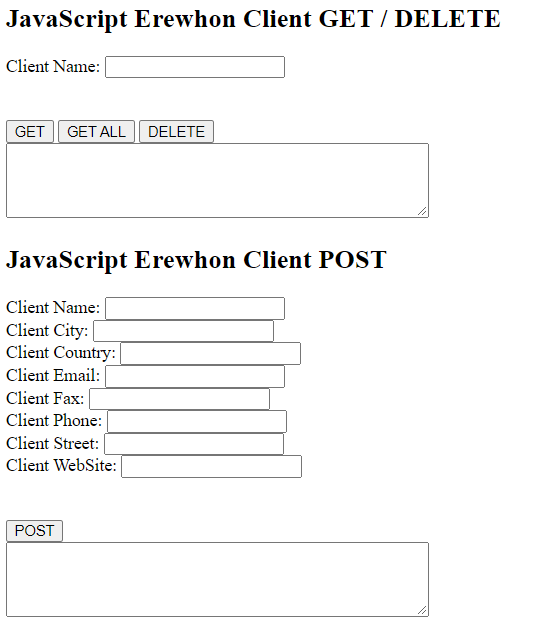
* It should now look something like this:



* (Don’t forget to save)

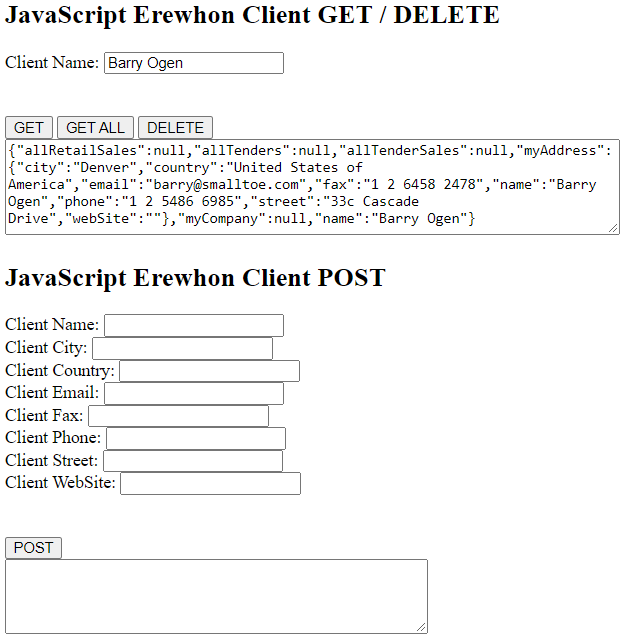
Thirdly, we’ll add the Erewhon Client JavaScripts to the bin directory of your JADE install (because that’s where your IIS application is pointing to) so that we can access them through IIS.

* Grab the **ErewhonClient.html** and **ErewhonAgent.html** from MS Teams and put them in your bin directory.
* In a web browser, navigate to [**http://localhost/jadeRestSite/ErewhonClient.html**](http://localhost/jadeRestSite/ErewhonClient.html)
* You should see the following page:

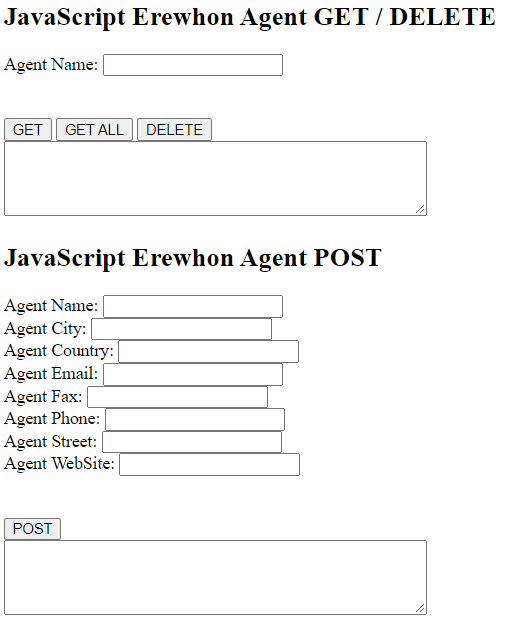


*NOTE: This is a simple JavaScript REST client that I whipped up so you can experience connecting JADE’s REST with a variety of other technologies. It’s not exactly pretty and it doesn’t handle errors so well but will give you a basic idea of how RESTing works in JavaScript.*

* In JADE, in the **ErewhonInvestmentsViewSchema**, start the **ErewhonRestApp** if you haven’t already.
* In the JavaScript client, set the Client Name (in the GET / DELETE section to **Barry Ogen**
* Click GET

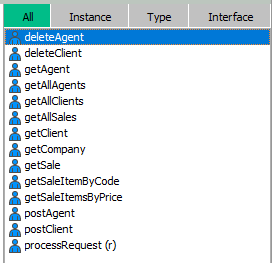


* Try the other operations – try POSTing a new client, GETting it, and then DELETEing it.
* Now, in your browser, navigate to <http://localhost/jadeRestSite/ErewhonAgent.html>



If you try to use this one, you will see nothing works.

* In JADE, in ErewhonInvestmentsViewSchema, navigate to ErewhonRestService.
* You will see that there are a bunch of methods:



The Agent methods have no source yet, while the Client ones do.

* Implement the Agent methods such that the JavaScript Erewhon Agent works like the Client one. You will need to implement the following methods.
  + deleteAgent
  + getAgent
  + getAllAgents
  + postAgent

**TIP:** Use the corresponding Client methods as inspiration – if you’re feeling confident try to code them mostly yourselves, or if you’re getting a bit more stuck lean on the existing ones more. If you really want to cheat, copy-paste the Client method and replace Client with Agent and that should work 😊

* As you do this, try to answer the following questions:
  + Why are we cloning things?
  + Why are we using an AddressProxy rather than an Address?
  + What extra could we do to improve the GET methods?
  + What is funny about the DELETE and POST methods?

## Exercise 3 – REST Client (Optional)

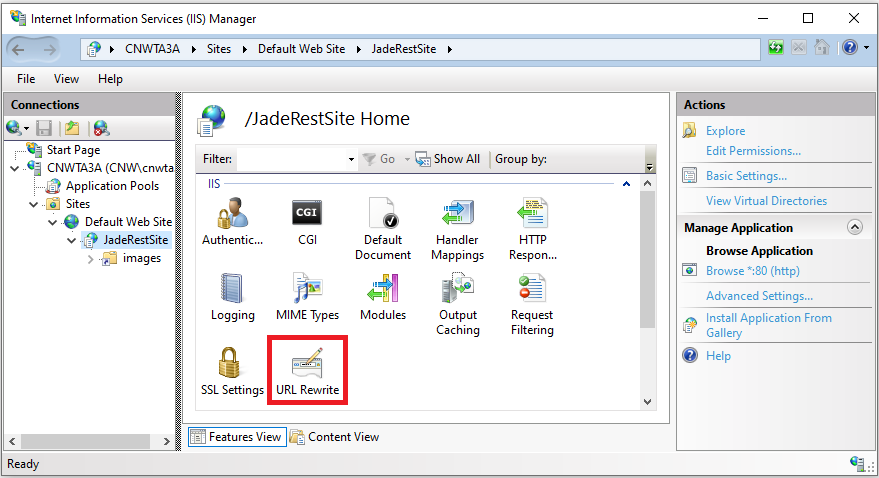
There is officially no exercise for topic 3, REST Clients.

If you would like to do something in this area, you could consider reimplementing the JavaScript Erewhon Client in JADE.

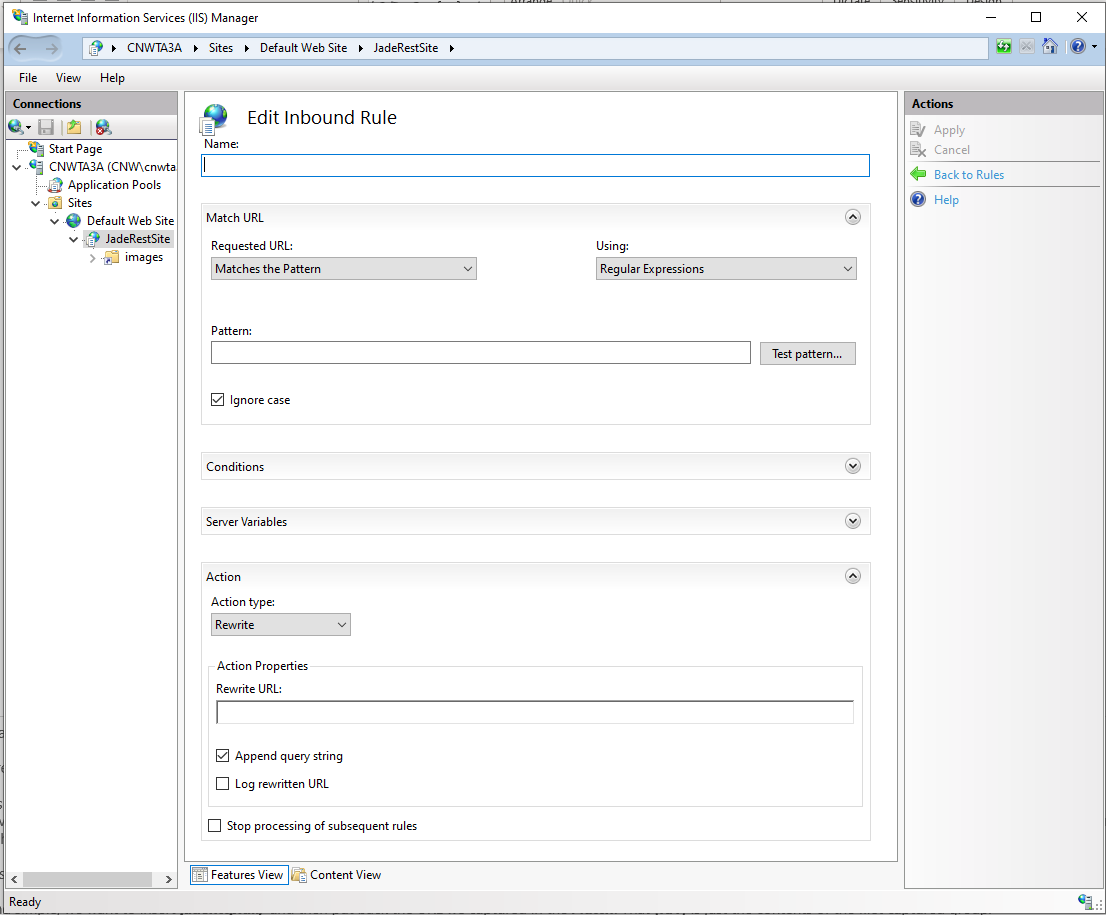
## Exercise 4 – URL Rewrite

In this exercise, you will download and install URL Rewrite, then use it to make your JADE Erewhon REST Service have a more standard-style URL.

* In a web browser, go to <https://www.iis.net/downloads/microsoft/url-rewrite>
* Click the **Install this extension** button to install IIS rewrite.
* Run the downloaded **urlrewrite2.exe**.
* Once it’s installed, check that URL Rewrite is present in your IIS manager:



* Double-click **URL Rewrite** for your **JadeRestSite** application in IIS manager.
* From the rightmost pane, click **Add Rule(s)…** and select **Blank rule**.



* Come up with a witty name for your rule and enter it in the **Name** textbox.
* For the **Pattern**, we can use a simple regex expression: **(.\*)**

***(.\*)*** *is a regex expression that means match any character any number of times (e.g., the whole string) and save it to a regex group.*

* For the **Rewrite URL:** in the **Action** section, enter **jadehttp.dll/{R:1}?ErewhonRestApp**

***{R:1}*** *represents the group matched in the Pattern, i.e., the whole string – so the new URL will have* ***jadehttp.dll/*** *prepended to it and* ***?ErewhonRestApp*** *appended to it. Remember we’re already in the* ***JadeRestSite****, so the* ***http://localhost/JadeRestSite*** *part has already been processed and removed by IIS.*

* Click **Apply** in the righthand pane.
* Now it’s time to make sure we haven’t broken anything 😊.
* *(Make sure the Erewhon REST Application is running)*
* In a Web Browser, navigate to <http://localhost/JadeRestSite/AllClients>
* You should see a JSON string with all the clients.