



Real-Time Predictive Analytics Scoring in Bluemix

Table of Contents

1	Document Purpose	3
2	Prerequisites	4
2.1	Bluemix	4
2.2	Cloud Foundry	4
2.3	Node.js	5
2.4	Download Sample 2	6
3	IBM SPSS Modeler	7
4	Bluemix Application and Service	8
4.1	Application Creation	8
4.2	Service Creation	10
5	File Customization	13
5.1	\app\manifest.yml	13
5.2	\app\app.js	14
5.3	\app\public\js\app.js	14
5.4	\app\public\js\srv.js	15
6	Porting Source Code	16
6.1	Node Package Manager	16
7	Uploading SPSS Models	18
8	Real-Time Scoring the Sample	19

1 Document Purpose

This document will outline how to configure and run the SPSS real-time scoring sample in Bluemix named Sample 1;

<https://github.com/pmservice/predictive-modeling-samples>

- This document is meant to compliment the current documentation associated with Sample 2.
- This document interfaces with an IBM SPSS Modeler stream but the focus is on Bluemix. It is assumed that the reader has a basic understanding of predictive analytics used in IBM SPSS Modeler.
- While there are many supported runtimes in Bluemix (Liberty for Java, Python, Ruby, ASP.NET, Swift, Tomcat, etc.), this document will use SDK for Node.js.
- Other operating systems are supported but this example uses MS Windows.

2 Prerequisites

2.1 Bluemix

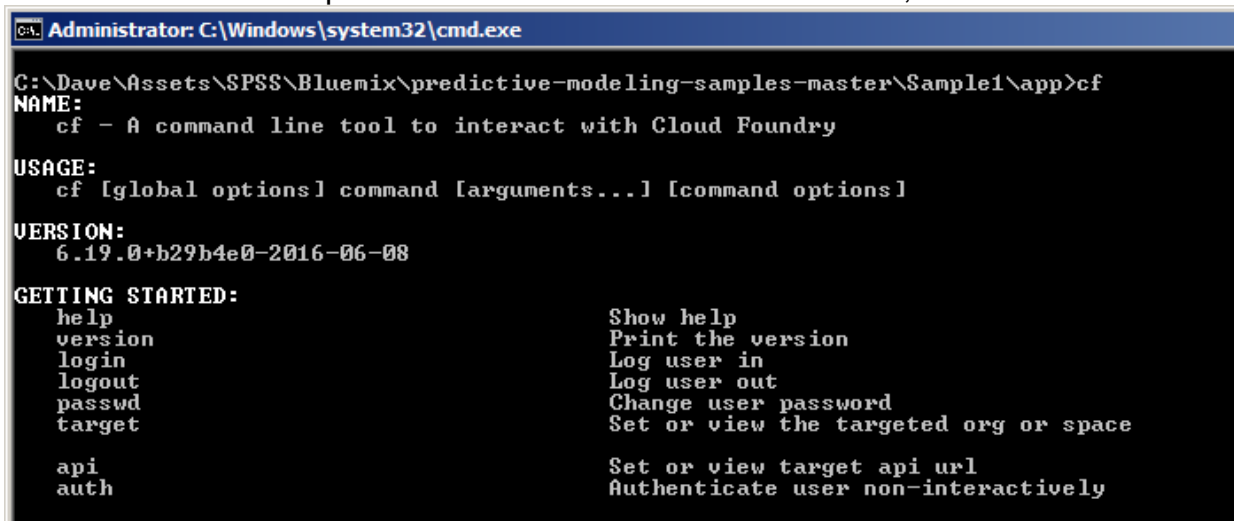
1. Ensure that you have a valid [Bluemix](#) account.

2.2 Cloud Foundry

2. Install the Cloud Foundry (CF) Command Line Interface;
<https://github.com/cloudfoundry/cli/releases>
3. Ensure that the tool is running properly by typing the following from a command prompt;

cf

You should receive output that looks similar to the screen below;



```
C:\Administrator: C:\Windows\system32\cmd.exe

G:\Dave\Assets\SPSS\Bluemix\predictive-modeling-samples-master\Sample1\app>cf
NAME:
  cf - A command line tool to interact with Cloud Foundry

USAGE:
  cf [global options] command [arguments...] [command options]

VERSION:
  6.19.0+b29b4e0-2016-06-08

GETTING STARTED:
  help                Show help
  version             Print the version
  login              Log user in
  logout            Log user out
  passwd            Change user password
  target            Set or view the targeted org or space

  api                Set or view target api url
  auth              Authenticate user non-interactively
```

Note: you may have to reboot your system in order for it to locate the “cf” executable. Do not proceed if the “cf” command does not run.

2.3 Node.js

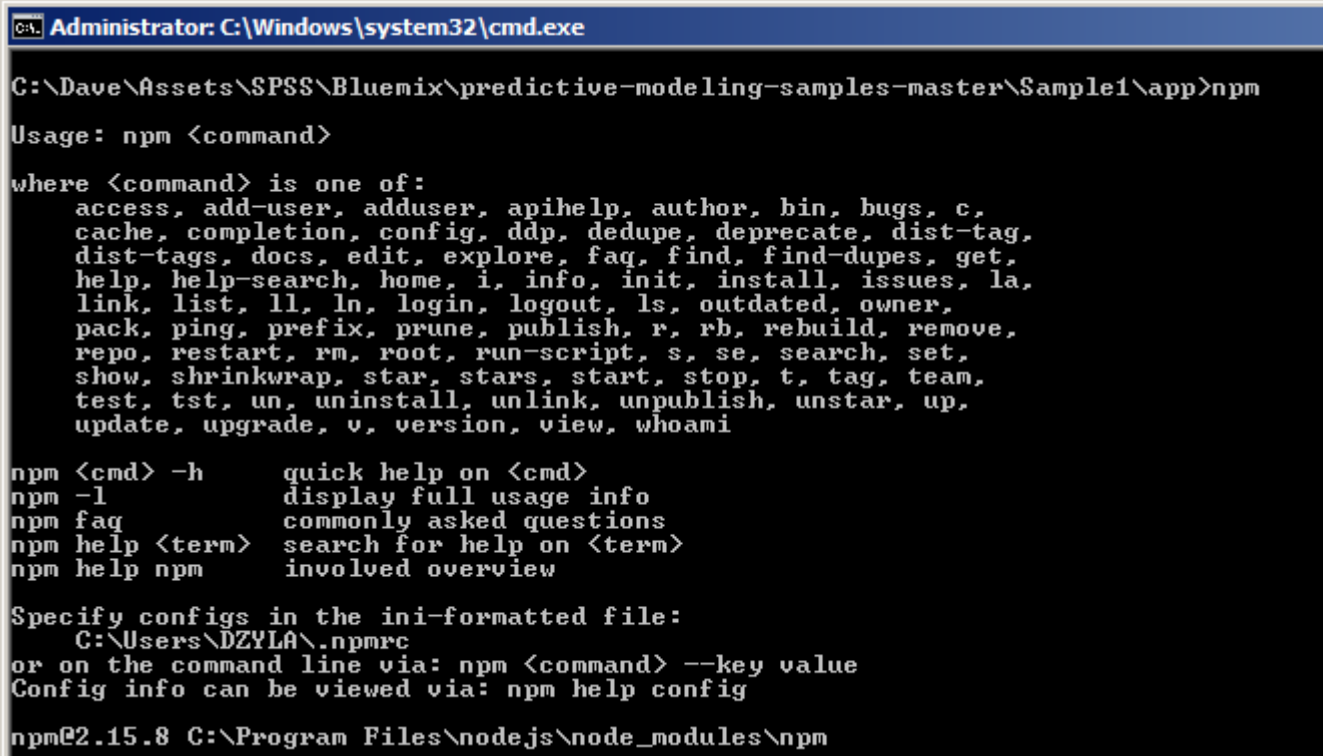
Node.js is required to download supporting code from Bluemix to the local development environment. Additional information can be found at;

<https://nodejs.org/en/about/>

1. Download and install Node.js from;
<https://nodejs.org/en/download/>
2. Validate that Node.js is running properly by typing the following from a command prompt;

npm

You should receive output that looks similar to the screen below;



```
Administrator: C:\Windows\system32\cmd.exe
C:\>C:\Dave\Assets\SPSS\Bluemix\predictive-modeling-samples-master\Sample1\app>npm
Usage: npm <command>

where <command> is one of:
  access, add-user, adduser, apihelp, author, bin, bugs, c,
  cache, completion, config, ddp, dedupe, deprecate, dist-tag,
  dist-tags, docs, edit, explore, faq, find, find-dupes, get,
  help, help-search, home, i, info, init, install, issues, la,
  link, list, ll, ln, login, logout, ls, outdated, owner,
  pack, ping, prefix, prune, publish, r, rb, rebuild, remove,
  repo, restart, rm, root, run-script, s, se, search, set,
  show, shrinkwrap, star, stars, start, stop, t, tag, team,
  test, tst, un, uninstall, unlink, unpublish, unstar, up,
  update, upgrade, v, version, view, whoami

npm <cmd> -h      quick help on <cmd>
npm -l           display full usage info
npm faq          commonly asked questions
npm help <term>  search for help on <term>
npm help npm     involved overview

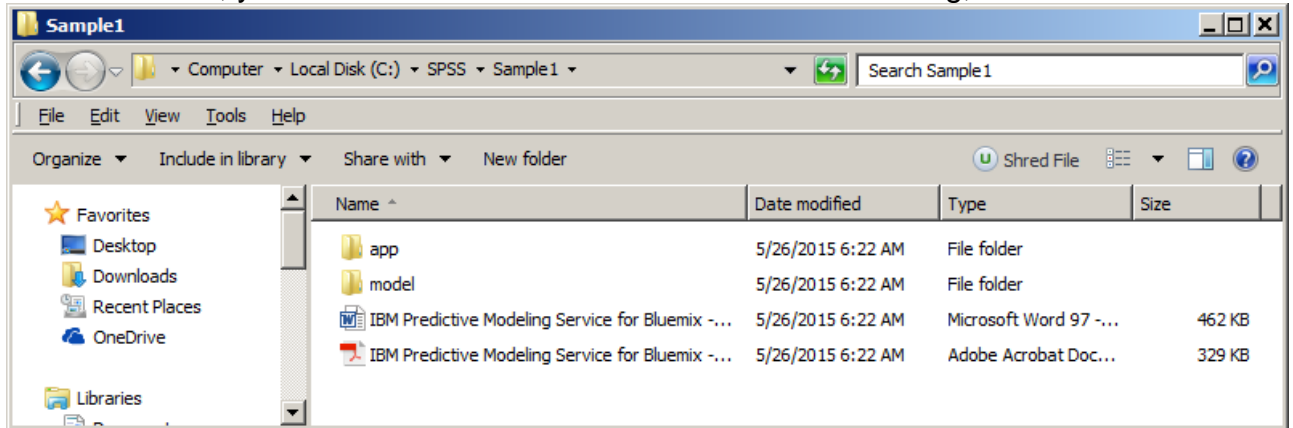
Specify configs in the ini-formatted file:
  C:\Users\DZYL\A\npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@2.15.8 C:\Program Files\nodejs\node_modules\npm
```

Note: you may have to reboot your system in order for it to locate the “npm” executable. Do not proceed if the “npm” command does not run.

2.4 Download Sample 2

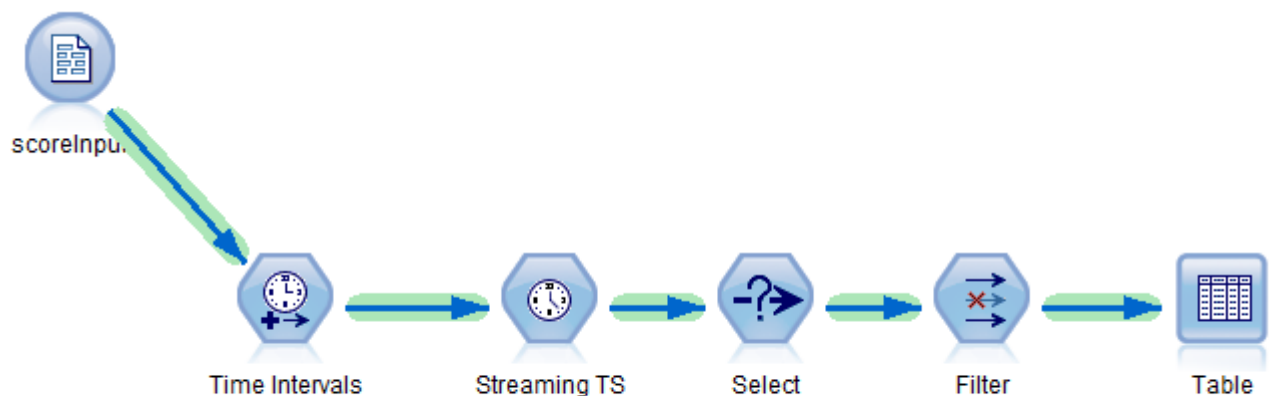
1. Download Sample 2 from;
<https://github.com/pmservice/predictive-modeling-samples>
2. Extract the contents to a working directory. This example will use;
C:\SPSS
3. Once extracted, you should have a folder structure like the following;



3 IBM SPSS Modeler

The sample comes with an SPSS Modeler stream which can be found at;
C:\SPSS\Sample1\model\Drug1n.str

It is important to note that a scoring branch has been set in the stream. Any model used in the Bluemix Predictive Analytics node must have a scoring branch set. This can simply be done by right clicking the terminal node and selecting “Use as scoring branch”.

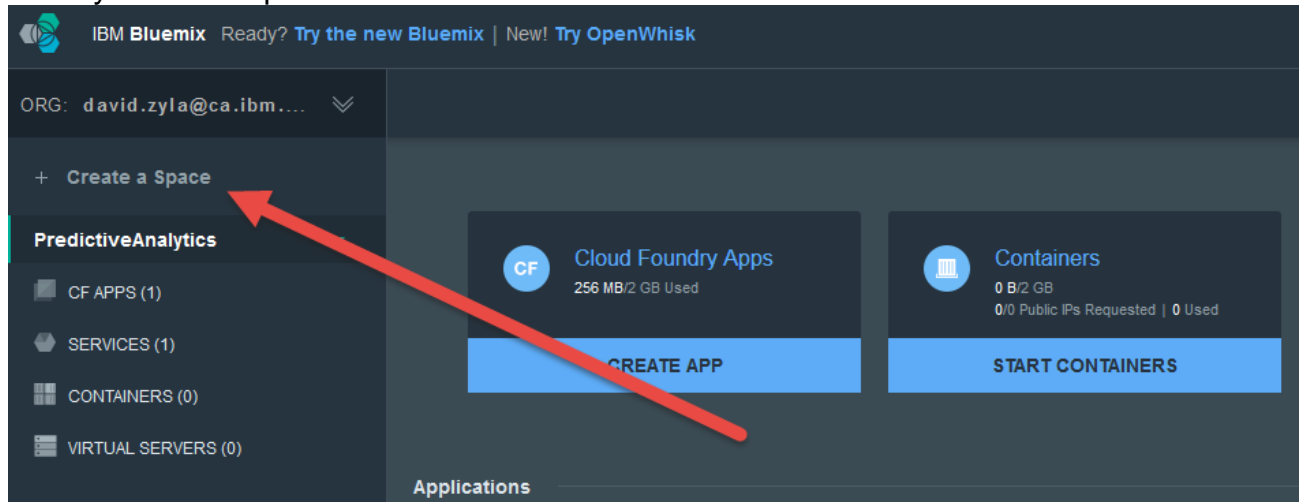


The Bluemix application will have to provide the same data items used in the input node. In this case it is the “scoreInput” node.

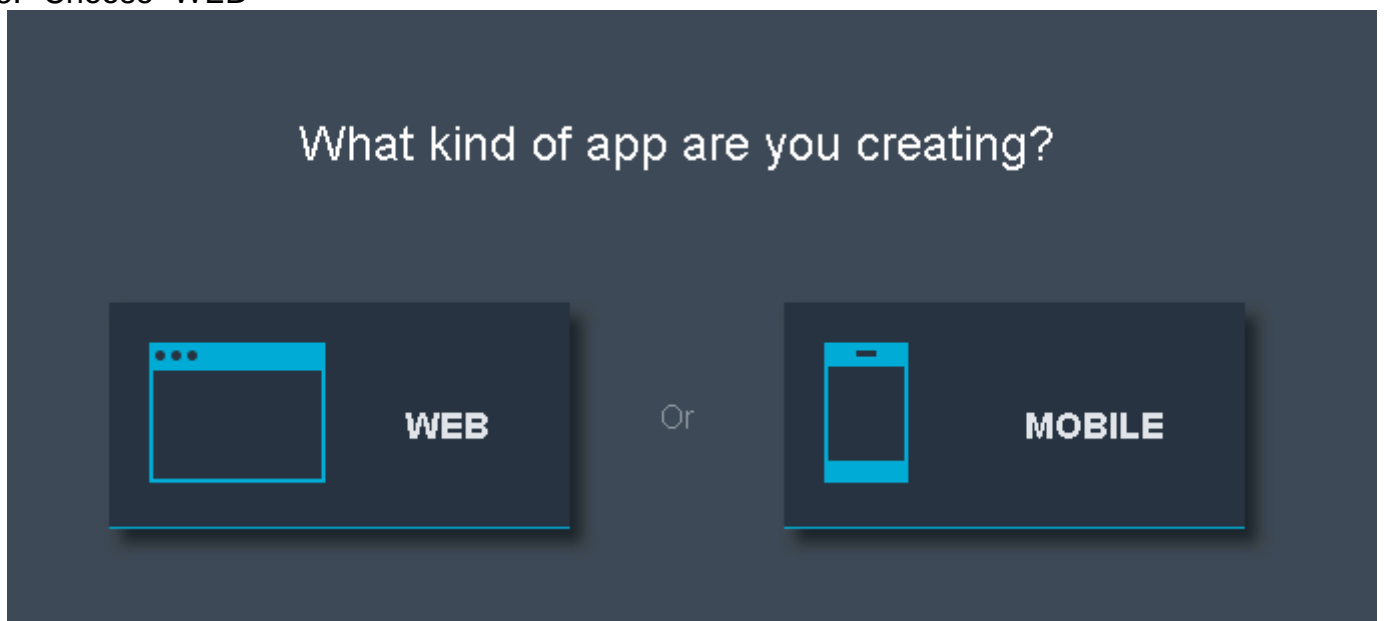
4 Bluemix Application and Service

4.1 Application Creation

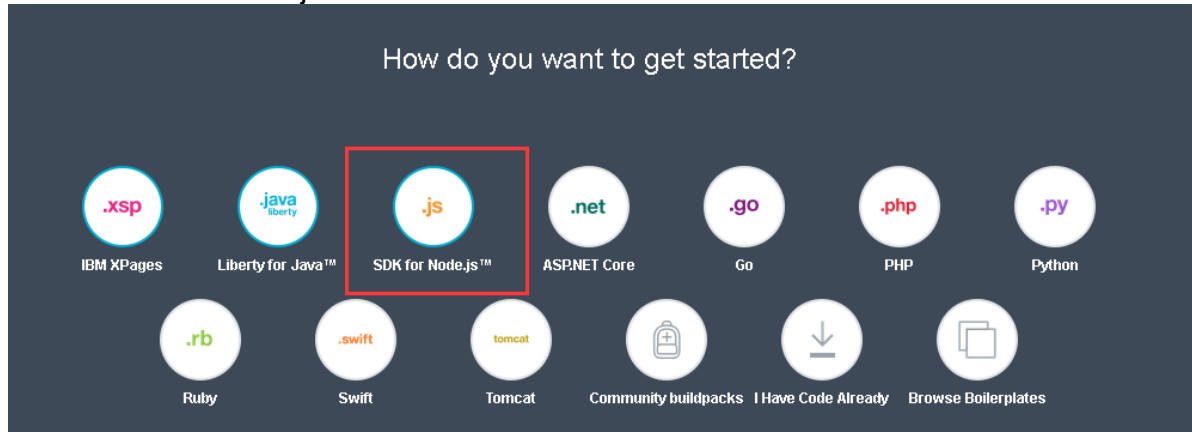
1. Login to Bluemix.
2. If you don't already have a Space created, create one now. The space can be whatever name you like but please make note of it for future reference.



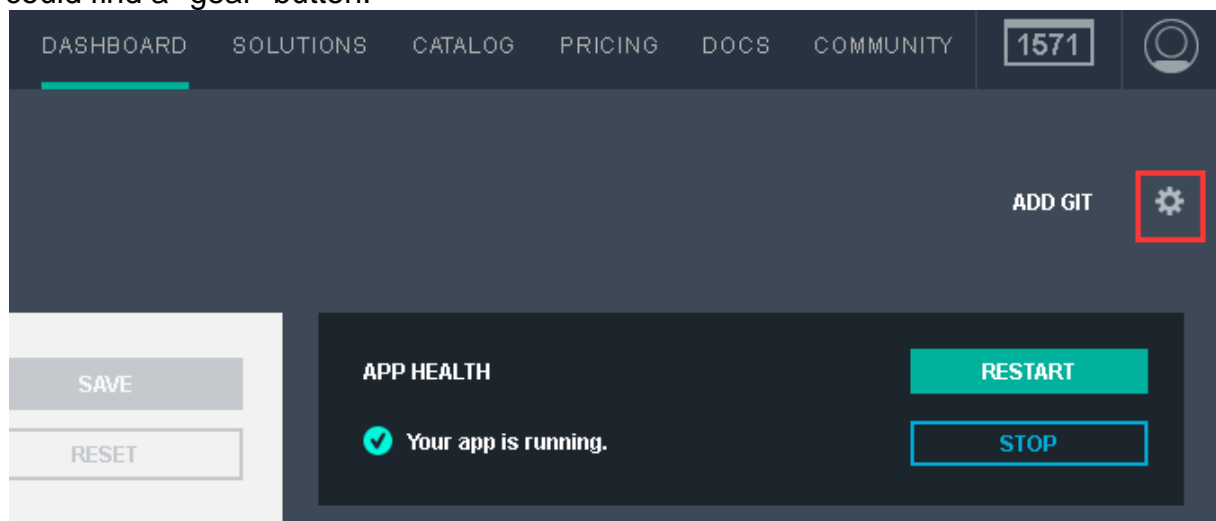
3. Once you have created the new space, click on it on the left hand side of your screen.
4. On the top of the page click on "Create app"
5. Choose "WEB"



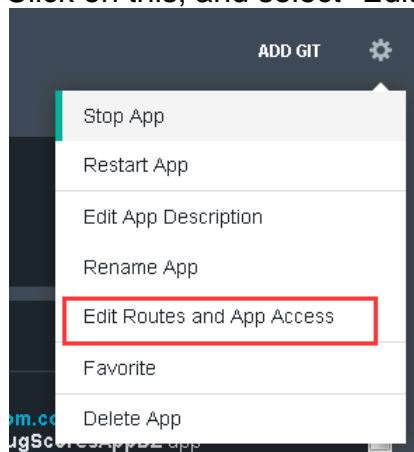
6. Select SDK for Node.js



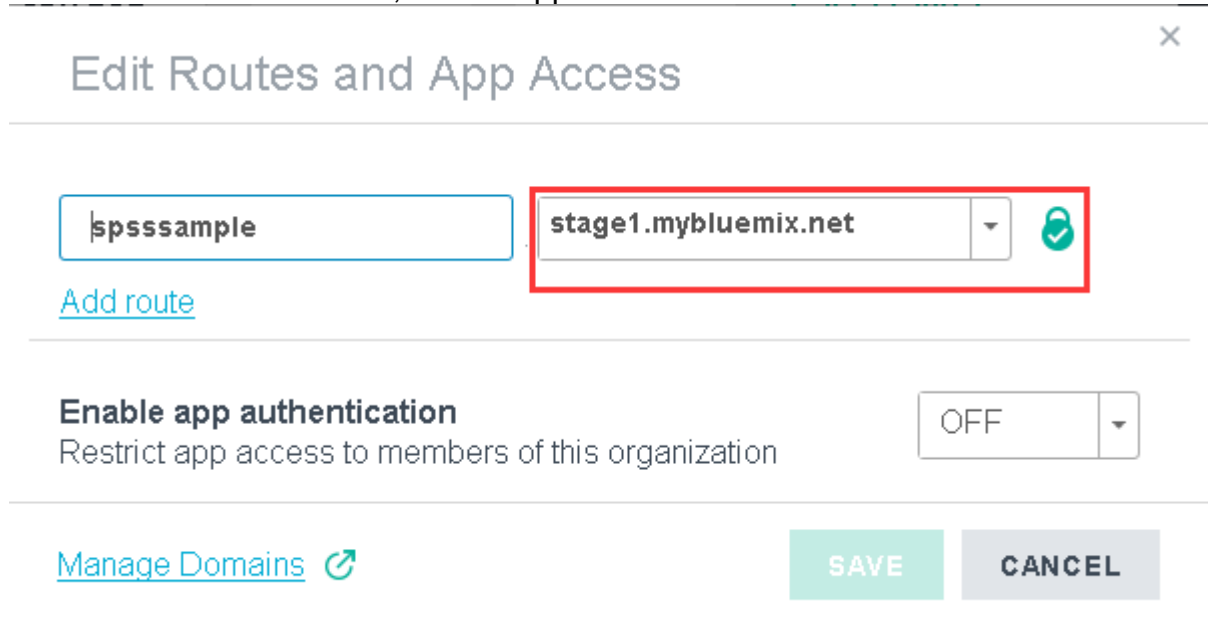
7. Name the new application. Note that this name can be whatever you want. This example will use "SPSSSample". (Or any other name you like, but it can't be too long)
8. Click Create. This will stage and create the application. This process may take a minute or two to complete.
9. After the application is created, go back to the dashboard, you could see the application "SPSSSample".
10. Click on the application, and go to the application overview page. On the top right corner, you could find a "gear" button.



11. Click on this, and select "Edit Routes and App Access"



12. In the second selection box, it's this app's domain name.



Close button (X)


Edit Routes and App Access

spsssampl stage1.mybluemix.net

[Add route](#)

Enable app authentication
Restrict app access to members of this organization

OFF

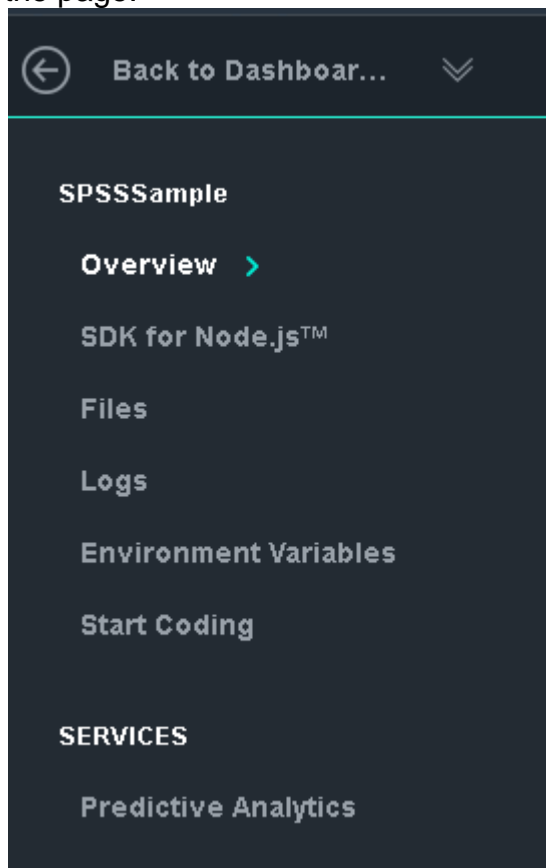
[Manage Domains](#) 

SAVE **CANCEL**

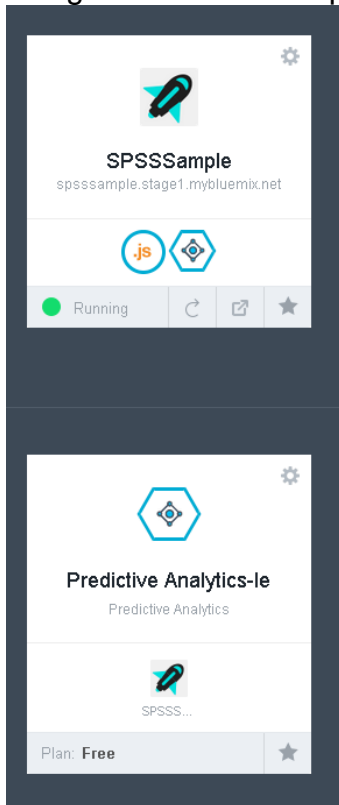
13. Make note of the Space, App Name, Host and Domain as they will be referred to later on.

4.2 Service Creation

1. Ensure that the SDK for Node.js application is running and is selected on the left hand side of the page.

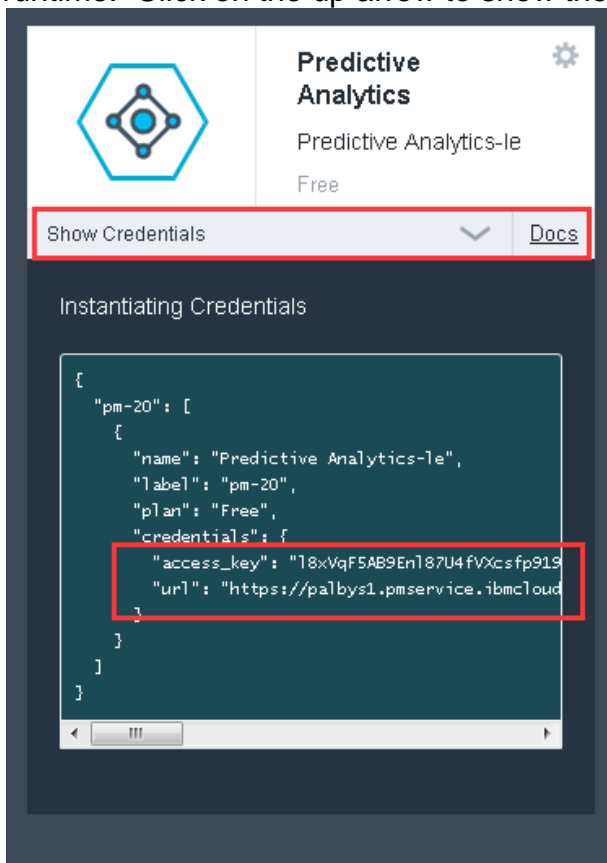


2. Click on Catalog from the upper toolbar.
3. To filter the number of services displayed, select “Data and Analytics” under Services on the left hand side of the screen.
4. Click on Predictive Analytics.
5. On the Add Service screen, ensure that the App drop down is changed from Leave Unbound to the application that you created previously. This will bind the service to the application.
6. Make note of the App and Service names.
7. Click Create. If prompted to re-stage, select yes.
8. Navigate back to the top Dashboard page to display both the application and service.



9. Click on the application (SPSSSample).

10. When we score the input data in this example, credentials have to be supplied to Bluemix at runtime. Click on the up arrow to show the credentials.



In the fly-out window, make note of the the access key and URL, these will needed later on.

11. At this point in time the application and service containers are ready.

5 File Customization

This section deals with customizing the files that were downloaded to the SPSS Working directory; C:\SPSS\Sample1

5.1 \app\manifest.yml

Manifest.yml contains information about the deployment and is typically used to reduce the number of deployment details that you must specify every time that you deploy an application to Bluemix.

1. Open the following file in a text editor;
C:\SPSS\Sample2\app\manifest.yml
2. Change the value of the **host** variable to reflect what was used in the Application Creation section above. In this example, SPSSSample was used
3. Change the value of the **name** variable to reflect what was used in the Application Creation section above. In this example, SPSSSample was used
4. Change the value of the **domain** variable to reflect what was used in the Application Creation section above. In this example, mybluemix.net was used
5. Change the value of the **memory** variable to 256M.
6. Add the following line;
disk_quota: 1024M
7. Save and close the file.

5.2 `lapp\app.js`

1. Open the following file in a text editor;
C:\SPSS\Sample2\app\app.js
2. Navigate to line 21 and update the defaultBaseUrl that was recorded in the Service Creation section above. Note that the <> must be removed and the entire string value be encapsulated in single quotes '.

```
var defaultBaseUrl = 'https://palbyp.pmservice.ibmcloud.com/pm/v1';
```

3. Similarly to the step above, update the defaultAccessKey value key on line 22.
4. Save and close the file.

5.3 `lapp\public\js\app.js`

1. Open the following file in a text editor;
C:\SPSS\Sample2\app\public\js\app.js
2. Navigate to line 8.
3. Note that the value of the \$scope.context variable is ['catalogTS', 'catalogSTS'] . While no change is required in this file, these steps are here to outline that every SPSS model scored in Bluemix must have a unique context ID. This will be discussed further in the Uploading SPSS Models section.
4. Close the file, no change is required.

5.4 `\\app\\public\\js\\srv.js`

1. Open the following file in a text editor;
C:\\SPSS\\Sample1\\app\\public\\js\\srv.js
2. Navigate to line 14. Note that the tablename variable matches the name of the source node in the SPSS Modeler model.



scoreInput

3. Close the file, no change is required.

6 Porting Source Code

6.1 Node Package Manager

The Node Package Manager utility is used to download supplemental files from Bluemix. The Start Coding section of the Bluemix application has some of the commands that will help you with connecting to your Bluemix environment. It may be easier to use the commands there than to customize the steps below.

1. Launch a command prompt.
2. Change directory to;
C:\SPSS\Sample2\app

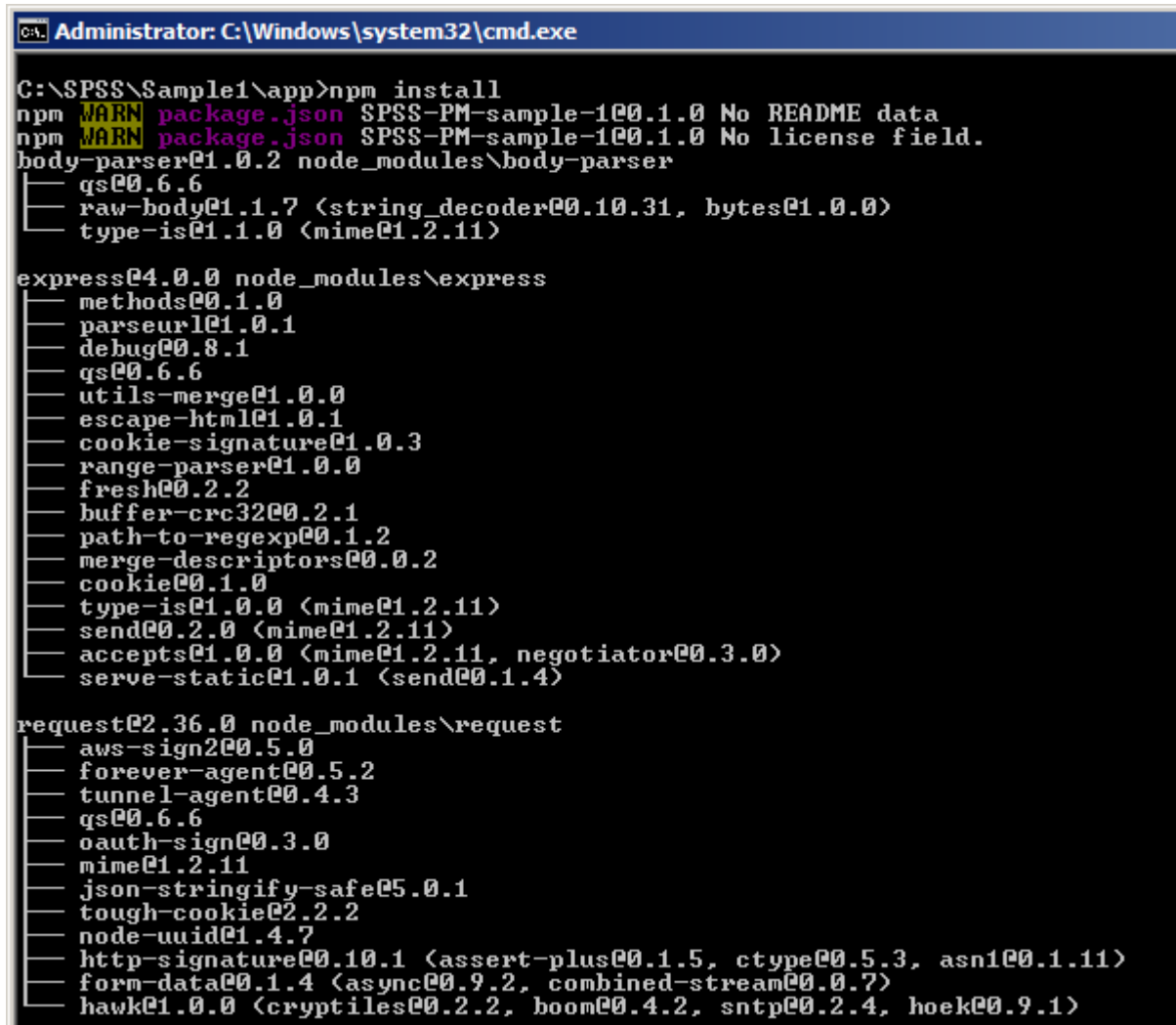


```
c:\SPSS\Sample2\app>
```

3. Connect to IBM® Bluemix® by typing the following command into the command prompt window.
bluemix api <https://api.ng.bluemix.net>
4. Login to Bluemix by typing the command below into the command prompt window.
bluemix login -u david.zyla@ca.ibm.com -o david.zyla@ca.ibm.com -s SPSS_RT_Scoring
5. Type your password when prompted.

- Download the supplemental files by typing;
npm install

This will create the following directory;
C:\SPSS\Sample2\app\node_modules



```
Administrator: C:\Windows\system32\cmd.exe

C:\SPSS\Sample1\app>npm install
npm WARN package.json SPSS-PM-sample-1@0.1.0 No README data
npm WARN package.json SPSS-PM-sample-1@0.1.0 No license field.
body-parser@1.0.2 node_modules\body-parser
├── qs@0.6.6
├── raw-body@1.1.7 <string_decoder@0.10.31, bytes@1.0.0>
└── type-is@1.1.0 <mime@1.2.11>

express@4.0.0 node_modules\express
├── methods@0.1.0
├── parseurl@1.0.1
├── debug@0.8.1
├── qs@0.6.6
├── utils-merge@1.0.0
├── escape-html@1.0.1
├── cookie-signature@1.0.3
├── range-parser@1.0.0
├── fresh@0.2.2
├── buffer-crc32@0.2.1
├── path-to-regexp@0.1.2
├── merge-descriptors@0.0.2
├── cookie@0.1.0
├── type-is@1.0.0 <mime@1.2.11>
├── send@0.2.0 <mime@1.2.11>
├── accepts@1.0.0 <mime@1.2.11, negotiator@0.3.0>
└── serve-static@1.0.1 <send@0.1.4>

request@2.36.0 node_modules\request
├── aws-sign2@0.5.0
├── forever-agent@0.5.2
├── tunnel-agent@0.4.3
├── qs@0.6.6
├── oauth-sign@0.3.0
├── mime@1.2.11
├── json-stringify-safe@5.0.1
├── tough-cookie@2.2.2
├── node-uuid@1.4.7
├── http-signature@0.10.1 <assert-plus@0.1.5, ctype@0.5.3, asn1@0.1.11>
├── form-data@0.1.4 <async@0.9.2, combined-stream@0.0.7>
└── hawk@1.0.0 <cryptiles@0.2.2, boom@0.4.2, sntp@0.2.4, hoek@0.9.1>
```

- To push all files along with the customizations that were made, run the following command;
cf push <Application Name>
cf push SPSSSample2
- The application has now been uploaded to Bluemix.

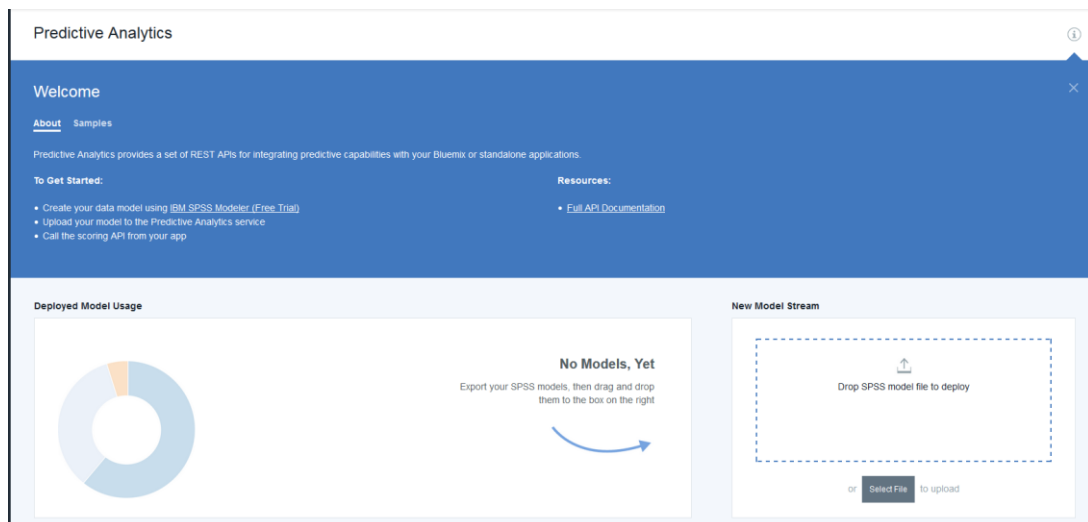
7 Uploading SPSS Models

At this point in time we have moved our custom application to IBM Bluemix but we still need to provide Bluemix with the SPSS models that we want to use.

1. Launch a browser and login to Bluemix.
2. From the Dashboard, click on the predictive analytics service that was created previously.
3. In the bottom right hand corner note that there is a New Model Stream section. The SPSS model that we are going to provide can be found in;
C:\SPSS\Sample2\model\catalog_timeseries.str
C:\SPSS\Sample2\model\ catalog_streaming_timeseries.str

Upload the model by either dragging and dropping it or by using the Select File option.

4. You will be prompted to provide a Context Id. While this ID can be anything, for this example, . catalog_timeseries.str as "catalogTS", catalog_streaming_timeseries as " catalogSTS "



5. Click Deploy.
6. The model will be listed in the bottom section of the screen.

Manage Models - Status: Active				
Context Id	File	Date Created	Date Updated	Action
catalogTS	catalog_timeseries.str	7/12/16	7/12/16	
catalogSTS	catalog_streaming_timeseries.str	7/12/16	7/12/16	

8 Real-Time Scoring the Sample

1. Within Bluemix, navigate to your Dashboard.
2. Click on the application that you created.
3. At the top of the screen you will be presented with the URL to access your application.




4. Click on the URL.
5. You will now be presented with your real-time scoring application.

 IBM SPSS Predictive Modeling service **"Time Series"** scoring applicaiton

A screenshot of the IBM SPSS Predictive Modeling service "Time Series" scoring application interface. The background is a solid teal color. At the top, the text 'Catalog Sales:' is followed by two radio buttons. The first radio button is selected and is labeled 'Men's Clothing'. The second radio button is unselected and is labeled 'Women's Clothing'. Below this, there are four input fields, each preceded by a label: 'Sales Input: 1', 'Sales Input: 2', 'Sales Input: 3', and 'Sales Input: 4'. The input fields contain the values '10000', '20000', '30000', and '40000' respectively. At the bottom left, there is a button with a small icon of a person and the text 'Score Now'.

6. You can change any of the values and set "men" or "woman" to simulate data being entered by a user of this application.

7. Click the Score Now button



\$TI_TimeIndex	\$TI_TimeLabel	\$TI_Week	\$TI_Day	\$TS-men	\$TSLCI-men	\$TSUCI-men
5	1388880000000	1	7	10915.129201	602.356084	21227.902317
6	1388966400000	2	1	11111.526551	798.753434	21424.299668
7	1389052800000	2	2	8711.233632	-1601.539484	19024.006749
8	1389139200000	2	3	11458.923238	1146.150121	21771.696355

Close

8. In this example the input along with the predicted drug and predicted confidence are being displayed to the screen.

You can see now that this exercise has walked through the process of;

- Setting up the necessary development tools
- Customizing the sample code
- Uploading an SPSS model
- Scoring the SPSS model with data provided in real-time