TechU

IBM

Intelligence at work

David Spurway
IBM Power Systems CTO, UK & Ireland

Session Objectives

- You don’t need to be an expert to do this. Anyone can cook with these ingredients!
- What started me on this journey and initial learning you can also use.
- My first creation was to answer a Frequently Asked Question. What FAQs do you see?
- The initial ingredients I used - Python, Beautiful Soup, Jupyter and Anaconda
- Lots of options available, so choose what works for you
- Node-RED is a powerful mixing bowl and IBM Watson Assistant can be the icing on the cake
- A quick taste of IBM Sales Manual Assistant
- My next bake brings in AIX, Db2 and James Bond!
- My first “Showstopper” is still to come
- What could you cook up?

TechU

IBM

Baking with IBM Watson Assistant, Node-RED, Bond Films and more!

David Spurway
IBM Power Systems CTO, UK & Ireland

2019 IBM Systems Technical University
Wednesday 23rd October 2019
Florenc 2-Mezzanine
c109178

What is the Great British Bake Off?

- “So we’ve got this fat Scouser and we’ve got this posh lady from down south and we’ve got a couple of comedians that haven’t been on the TV for a while and we stick them in a tent with 12 bakers.”
- “a show so popular that only England football matches compete with it for viewing figures”
- “a peak of 14.5m people, the largest TV audience of 2015, tuned in to watch Nadiya Hussain win”
- “It appeals to children, grannies, workers, everybody, because it’s gentle.”
- “sales of flour and cooking chocolate boom each time a new series starts”

I am not Paul Hollywood

"**Paul John Hollywood** (born 1 March 1966) is an English celebrity chef and television presenter, best known for being a judge on *The Great British Bake Off* since 2010.

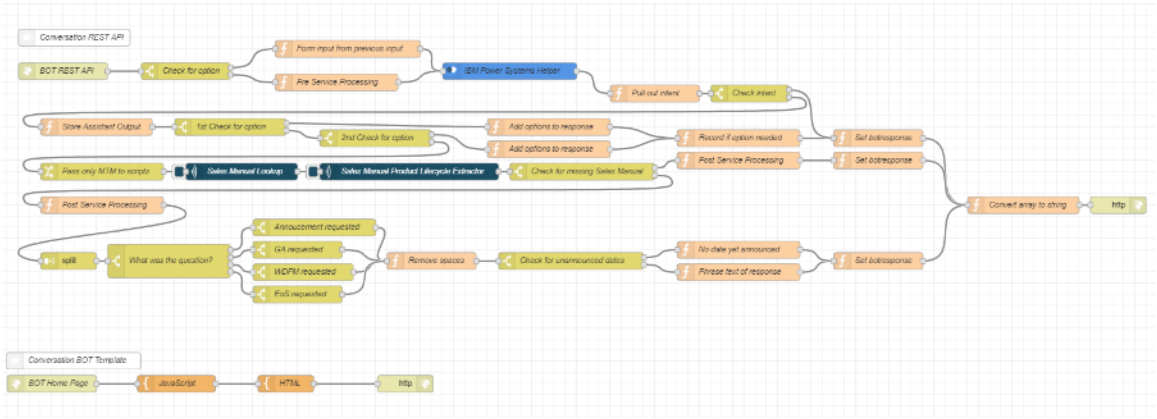
He began his career at his father's bakery as a teenager and went on to serve as head baker at a number of hotels around Britain and internationally.

- *100 Great Breads* (2004) Cassell, London ISBN 978-1-8440-3700-1
- *How to Bake* (2012) Bloomsbury ISBN 978-1-4088-1949-4
- *Paul Hollywood's Bread* (2013) Bloomsbury, London ISBN 978-1-4088-4069-6
- *Paul Hollywood's Pies and Puds* (2013) Bloomsbury, London ISBN 978-1-4088-4643-8
- *Paul Hollywood's British Baking* (2014) Bloomsbury USA ISBN 1408846489 ISBN 978-1408846483
- *The Weekend Baker* (2016) Michael Joseph, London ISBN 978-0-718-18401-8

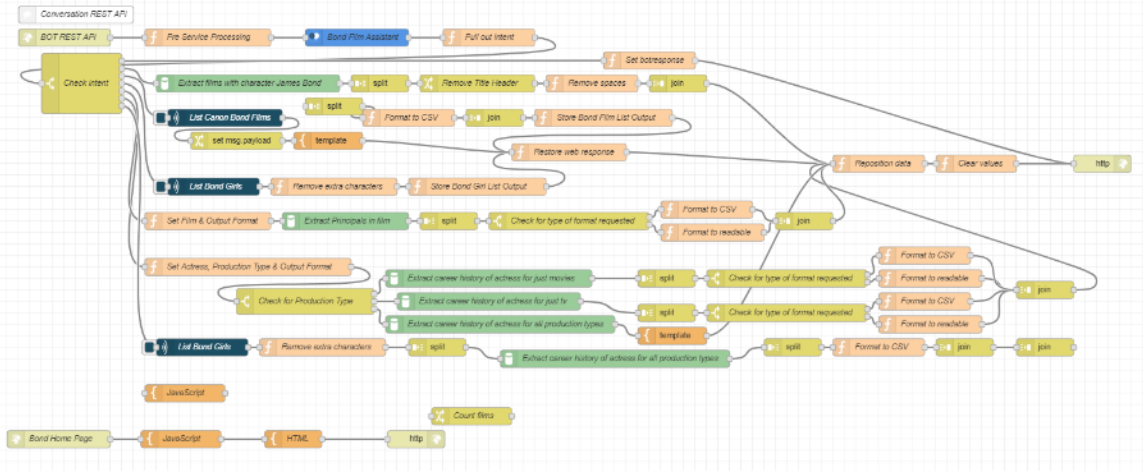
Anyone can bake

What have I baked so far?

IBM Sales Manual Assistant



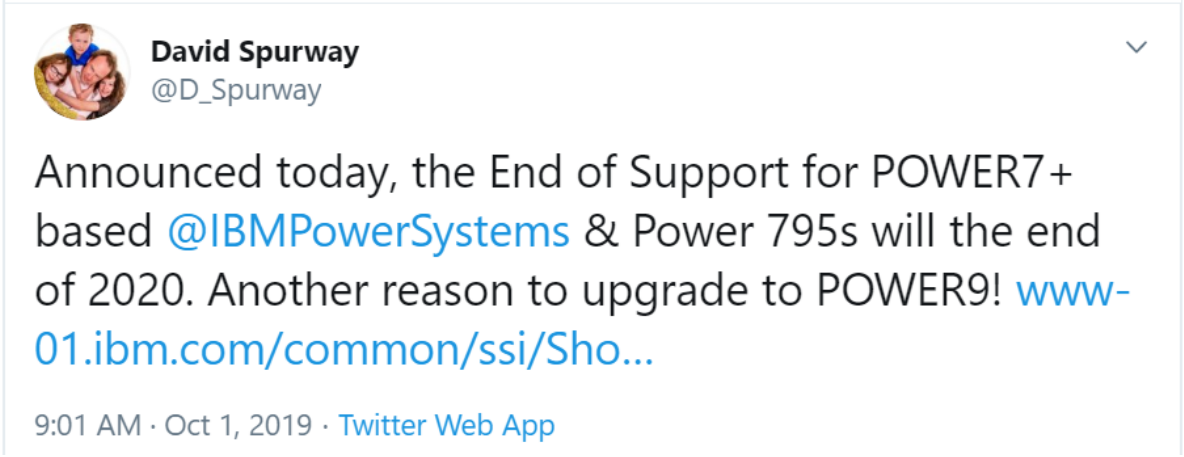
James Bond Assistant



In my first “bake”, answering an FAQ

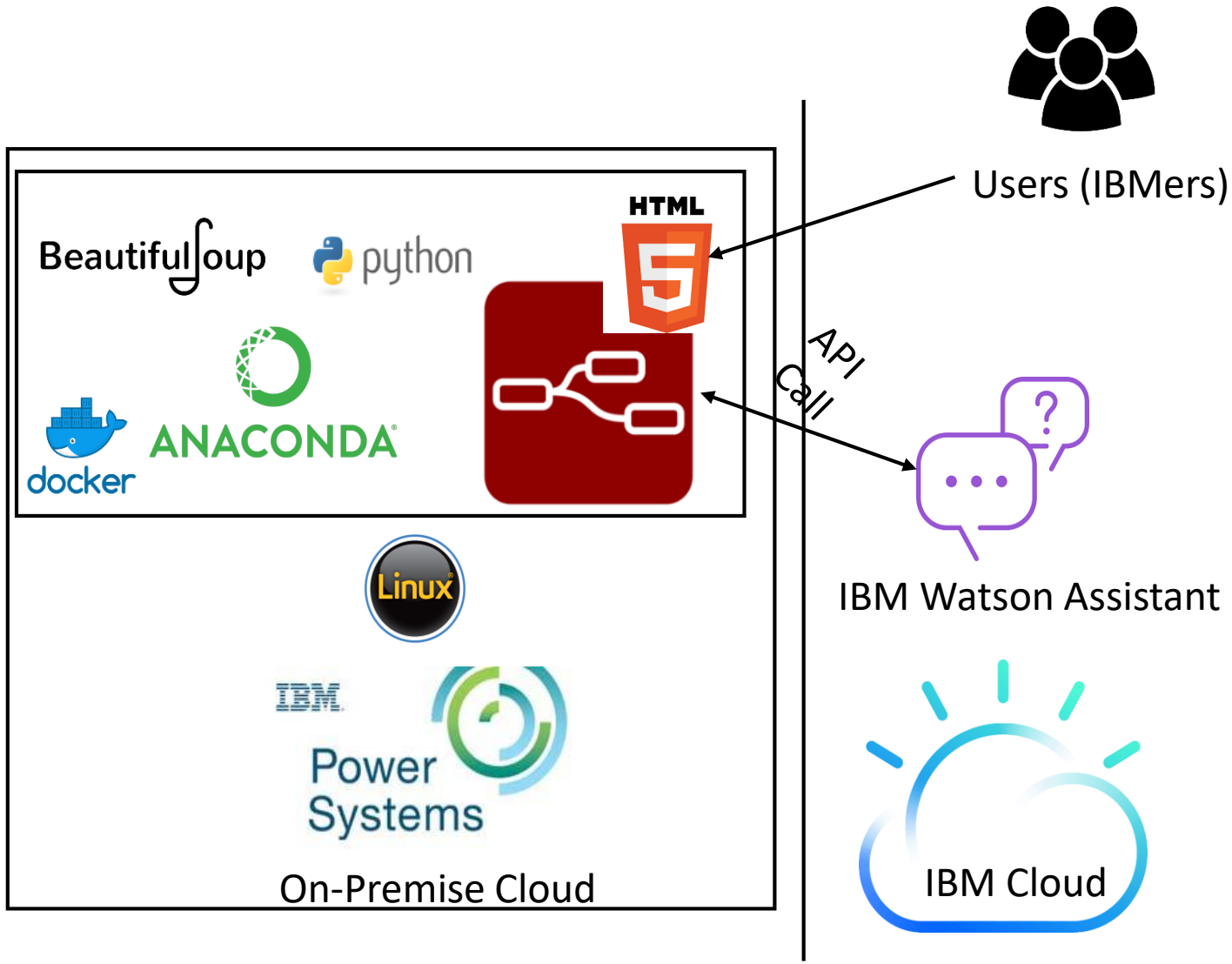
- I wrote a blog entitled “Those older IBM Power Systems were great, but the time has come to move up to POWER9!”
- https://www.linkedin.com/pulse/those-older-ibm-power-systems-were-great-time-has-come-david-spurway/

  - Majority of POWER5-based IBM iSeries & pSeries servers will reach End of Service 31/12/2019
  - Majority of POWER6-based IBM System i & System p servers reached End of Service 31/03/2019
  - Majority of POWER7-based IBM Power System servers will reach End of Service 30/09/2019



The high level recipe for the IBM Sales Manual Assistant
A Hybrid Cloud solution with AI

- Running in a Docker Container on Centos on IBM Power
- Anaconda to handle compatibility issues
- Node-RED calls
  - Watson Assistant
  - Python
    - Beautiful Soup
    - Urllib
- Users interact with a webpage, which is part of Node-RED



My first new ingredients

“Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.”

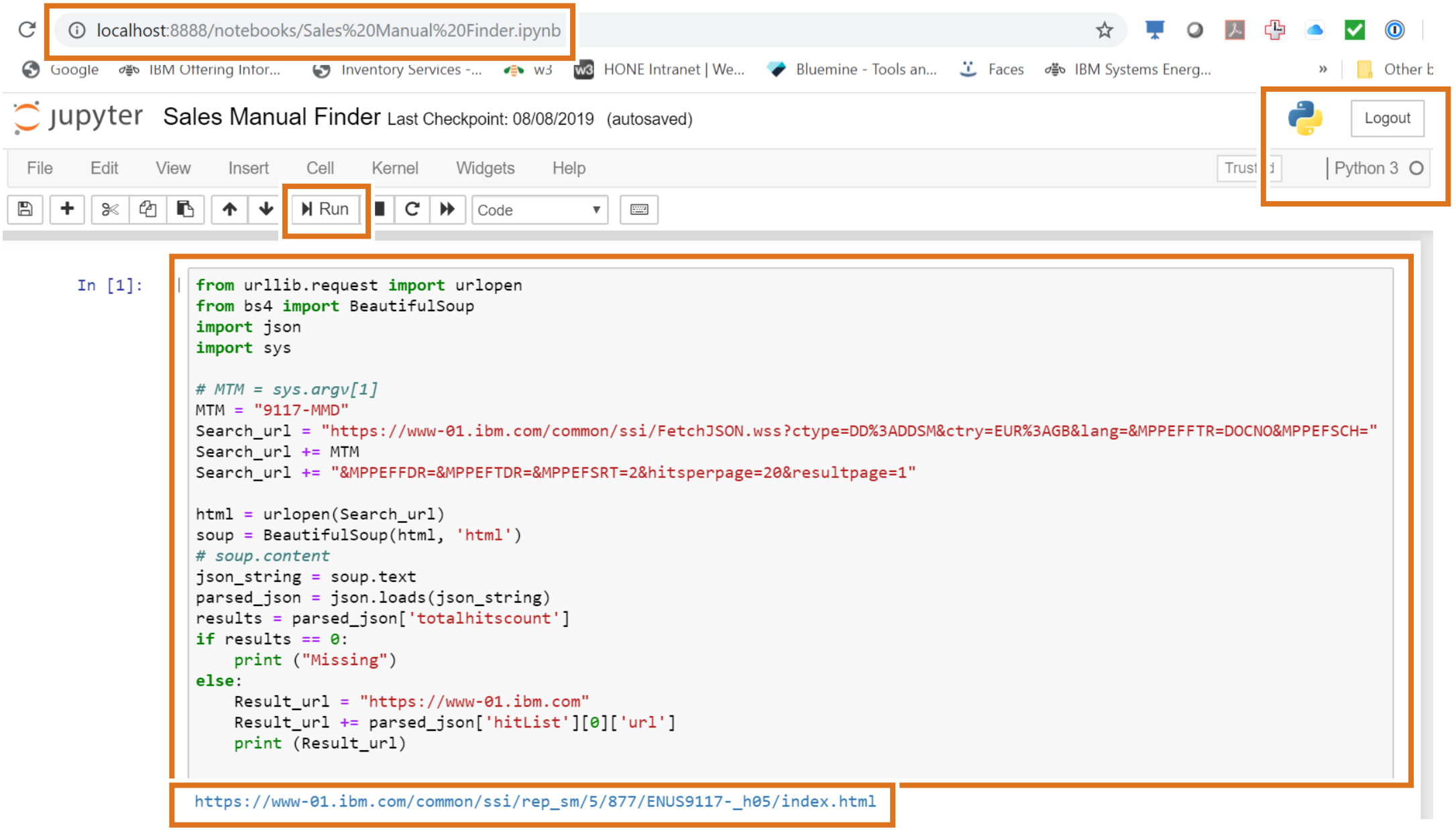
“Beautiful Soup is a Python library for pulling data out of HTML and XML files.”

Dough needs a rest, Python is easy with a Notebook

“Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.”

“Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.”

Get ready to bake with Jupyter Notebook



What I baked first combined Technical and Signature

A little bit of Python, calling Beautiful Soup:

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import sys

url = sys.argv[1]
if url == "Missing":
print("Missing")
else:
html = urlopen(url)
soup = BeautifulSoup(html, 'html')
[illegible]
Product_Life_Cycle_Table = Product_Life_Cycle_T[illegible].find_next('table')
[illegible]next('td')
[illegible]
Available = Announce.find_next('td')
WDFM = Available.find_next('td')
EOS = WDFM.find_next('td')
[illegible]t_text(),"\n",Available.get_text(),"\n",WDFM.get_text(),
"\n",EOS.get_text())
```

Work out which server is being talked about

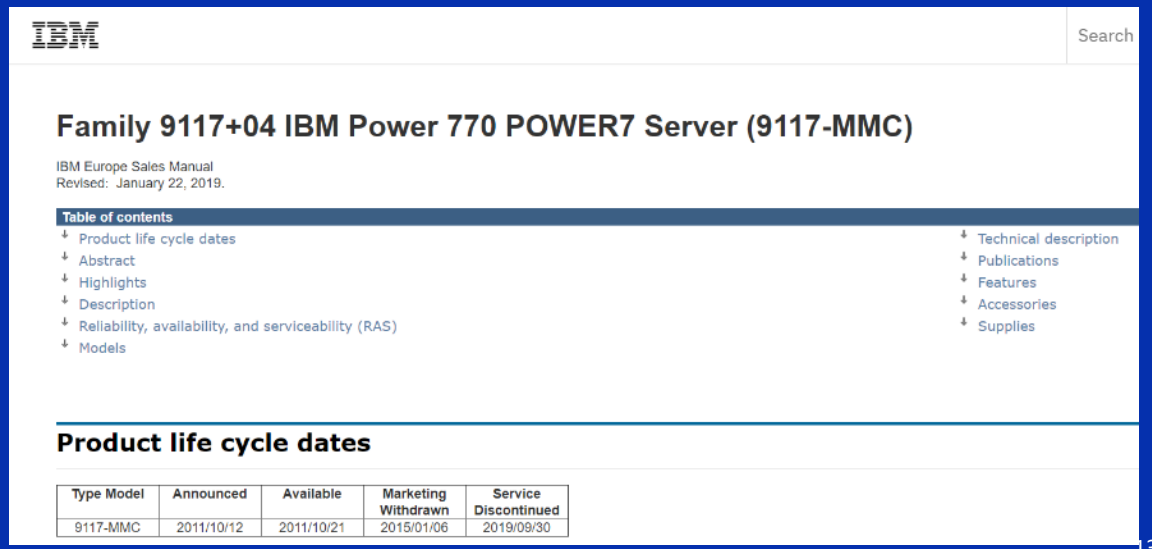
Example:

Maybe we are talking about the IBM Power 770, but there are three different versions of that server with different dates

Find the right “Sales Manual” page for the server

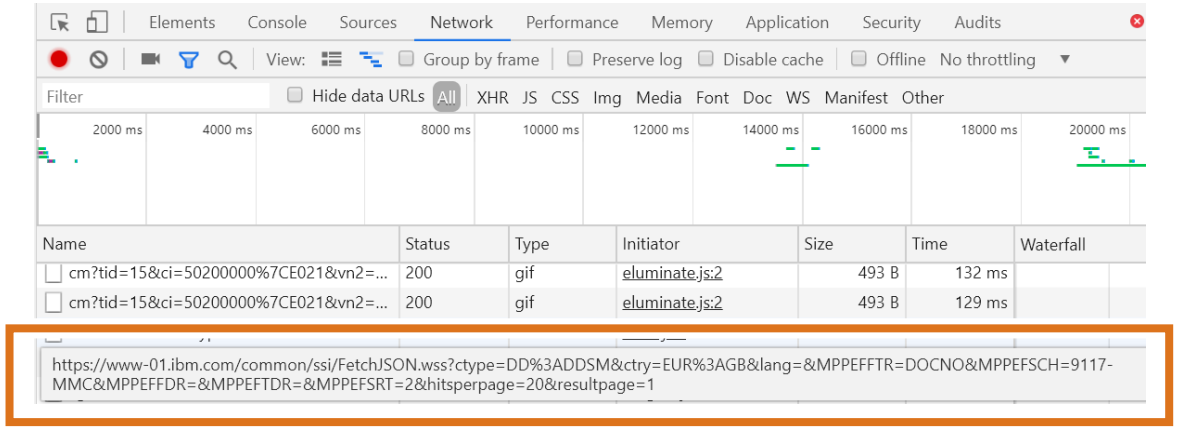
- Europe - IBM Power 770 Server MMB
- Europe - IBM Power 770 POWER7 Server (9117-MMC)
- Europe - IBM Power 770 POWER7+ Server (9117-MMD)

Read the dates from the table in the Sale Manual



A complication to the bake

Thanks to Ross Cruickshank for pointing me at the JSON being sent by the search, which can be accessed directly:



Unfortunately, the URLs for the Sale Manuals can change, there are sometimes duplicates and can even be missing

Oops — that's not right!

The page you are trying to access has either been moved or deleted.

The page used to search is dynamic, to Beautiful Soup can't directly parse the results

```
</div>
<div class="ibm-rule"></div>
<div class="ibm-fluid" id="hitlist">
</div>
<div class="ibm-fluid">
<div class="ibm-col-12-5">
</div>
```

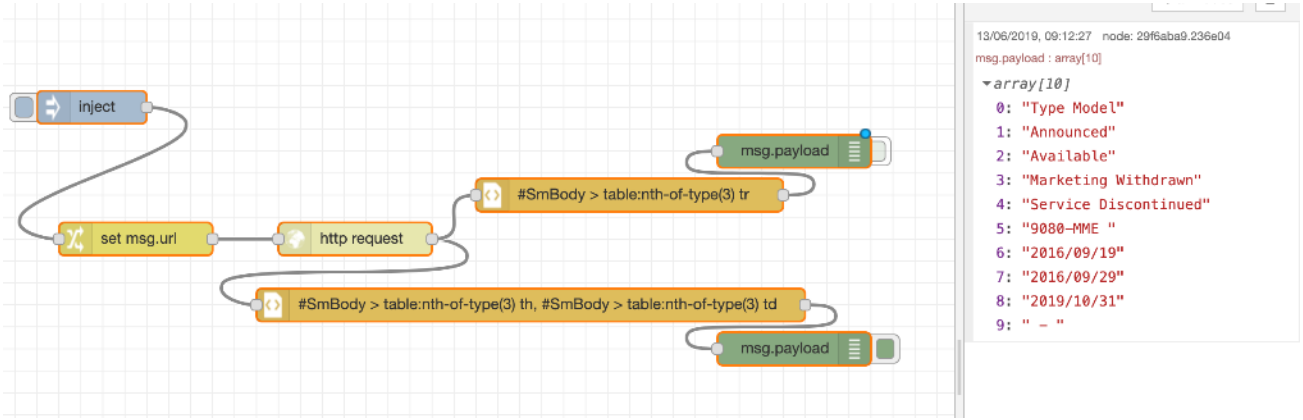
Inspecting my page, I can see my result, but…

```
▼<div class="ibm-fluid" id="hitlist">
  ▼<div class="ibm-col-                esult">
    ▼<a class="ibm-h5 ibm-bold" target="_blank" href="/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_sm/2/877/ENUS9080-_h02/index.html&request_locale=en">
        <span>EMEA - IBM Power E870C Enterprise server Model MME
        </span> == $0
      </a>
      <br>
      <span class="ibm-small ibm-bold ibm-textcolor-gray-30">
      Published: </span>
      <span class="ibm-date-time ibm-small ibm-bold ibm-textcolor-gray-50">21 May, 2019</span>
      <span class="ibm-small ibm-bold ibm-textcolor-gray-30"> |
      Document Number: </span>
      <span class="ibm-small ibm-bold ibm-textcolor-gray-50">M9080+02
      </span>
```


You choose the ingredients you want to use

I could have used pure Node-RED, without Python, as shown by Ross. That could have been usable directly in the IBM Cloud, as a Service.

But, the page I am working with has unhelpful HTML and it was a lot less clear how to do it and maintain.



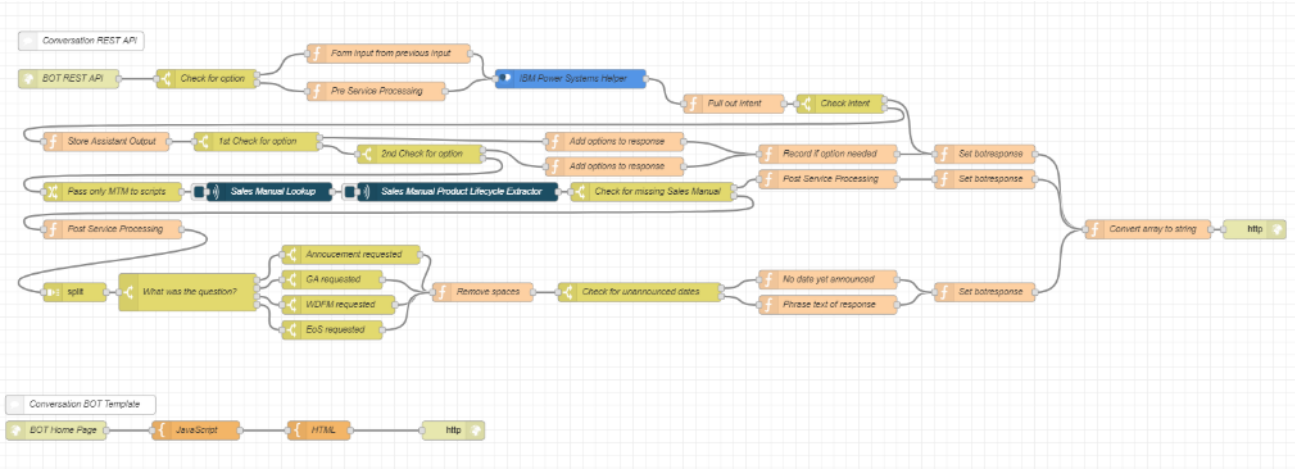
The ingredients I liked the taste of were Python and Json

My little Python script to find the Sales Manual(s) for a given Machine Type/Model, if it exists.

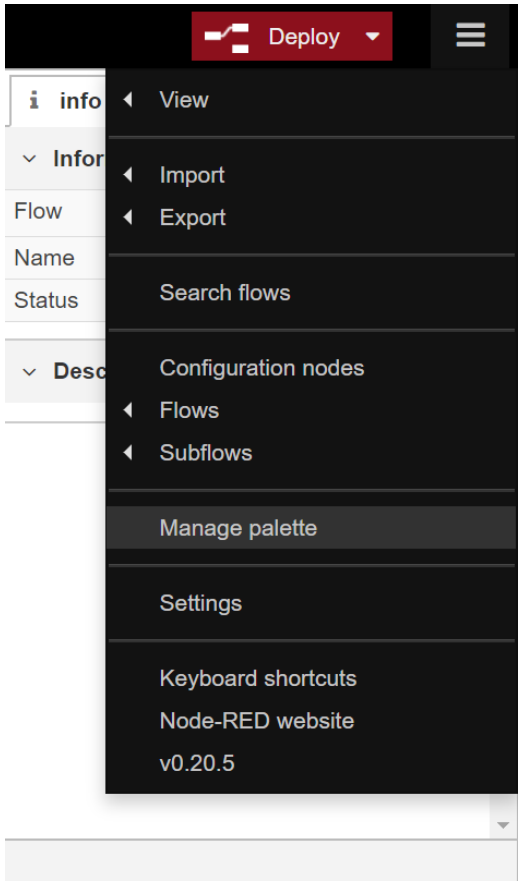
```
#!/usr/bin/env python
# coding: utf-8

from urllib.request import urlopen
from bs4 import BeautifulSoup
import json
import sys

MTM = sys.argv[1]
# MTM = "9080-MME

Search_url = "https://www-
01.ibm.com/common/ssi/FetchJSON.wss?ctype=DD%3ADSM&ctry=EUR%3AGB&lang=&MPPEFFTR=DOCNO&MPPEFSCH="

Search_url += MTM
"&MPPEFTDR=&MPPEFSRT=2&hitsperpage=20&resultpage=1"

html = urlopen(Search_url)
soup = BeautifulSoup(html, 'html')

json_string = soup.text
parsed_json = json.loads(json_string)
['totalhitscount']

print ("Missing")

Result_url = " https://www-01.ibm.com"
Result_url += parsed_json['hitList'][0]['url']
print (Result_url)
```

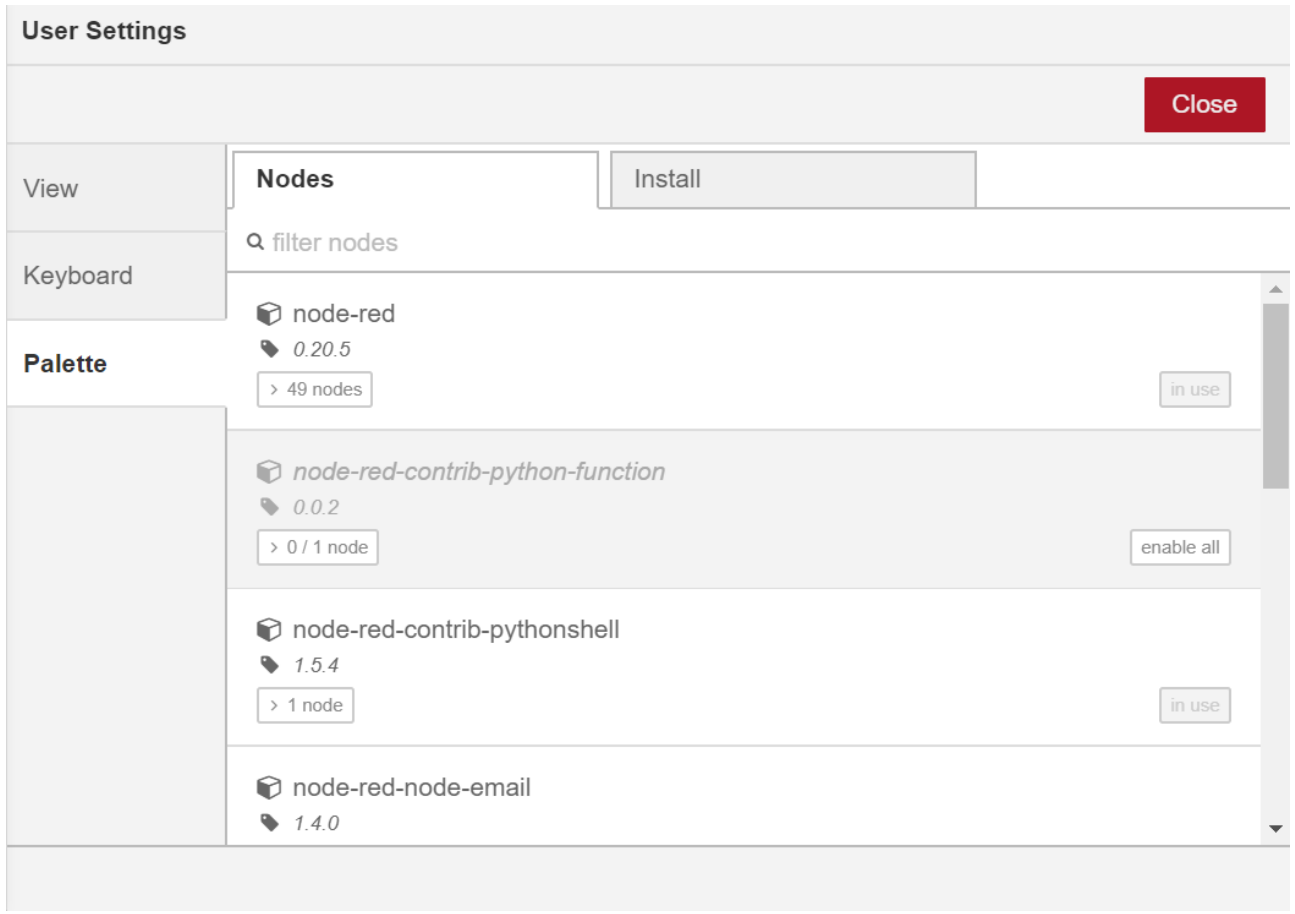

Move to the Node-RED Mixing Bowl

- Simple flow at the bottom creates the webpage
- Top flow detects input on the page, passes it to IBM Watson Assistant, which works out the MTM and the question
- Python scripts find the answer, and we return the result



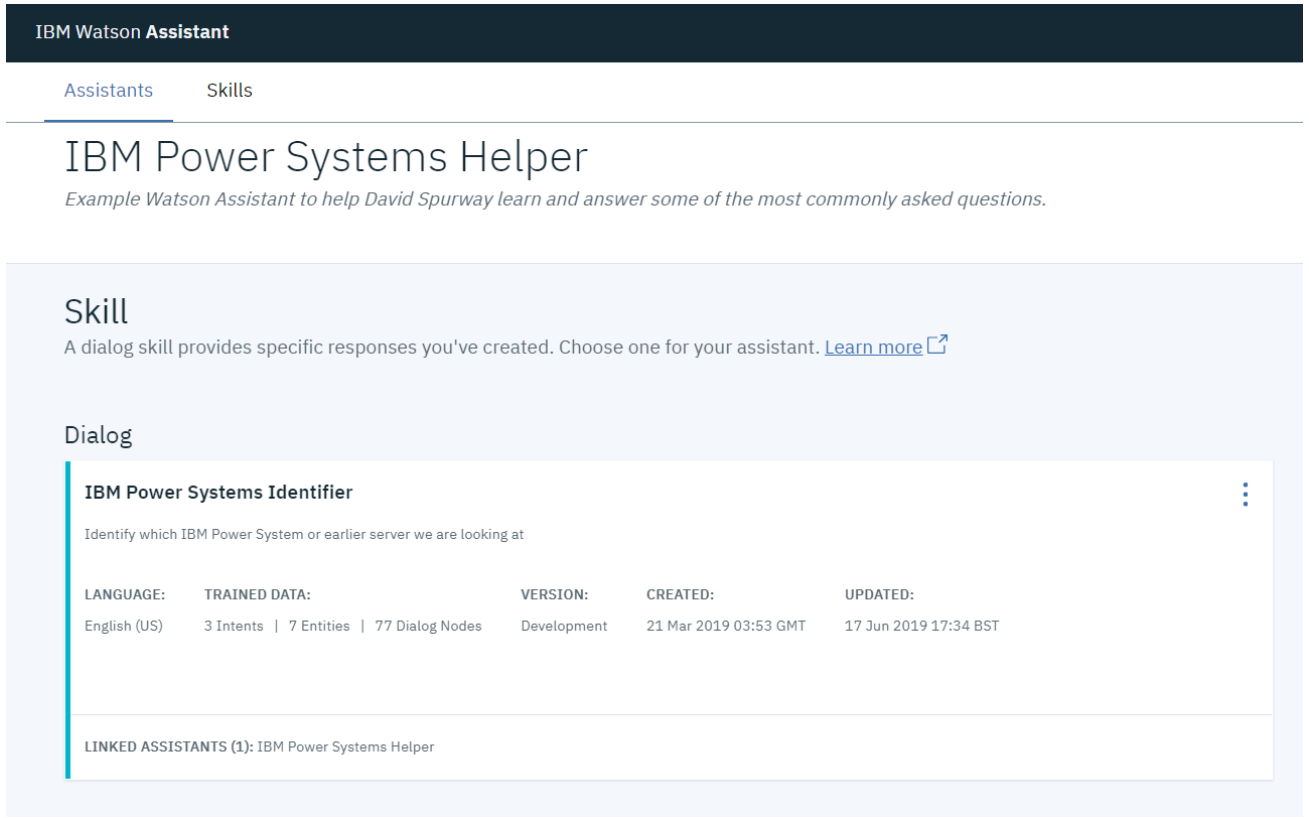
Adding my Python ingredient to Node-RED





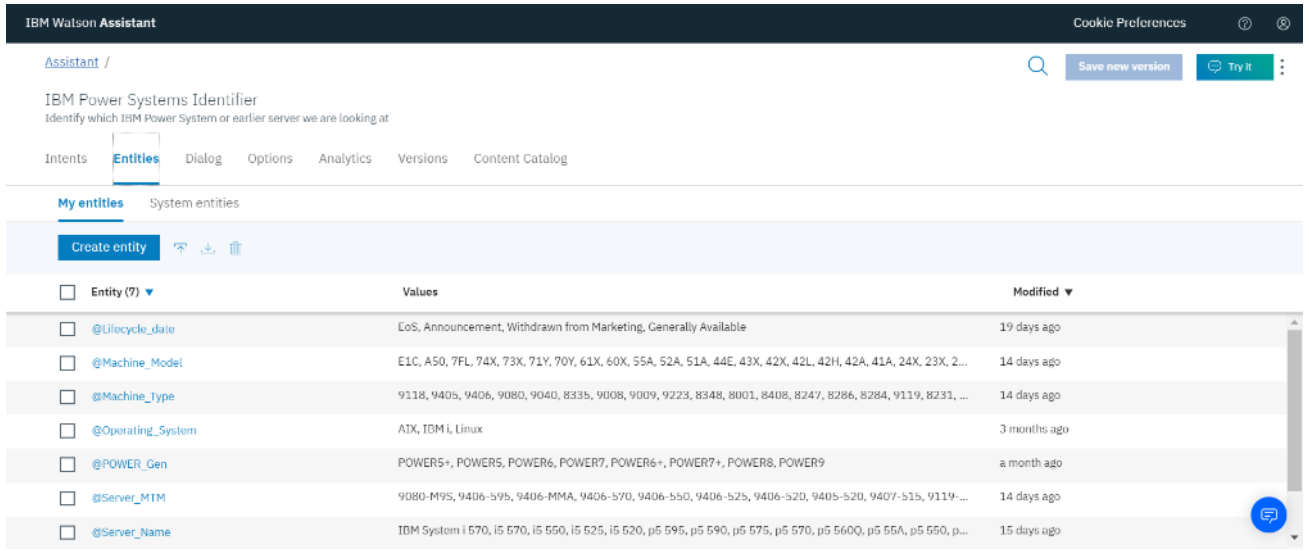
The icing on the cake is IBM Watson Assistant

- I built an IBM Watson Assistant to identify which server we are talking about.
- 10,000 Free Messages/Month



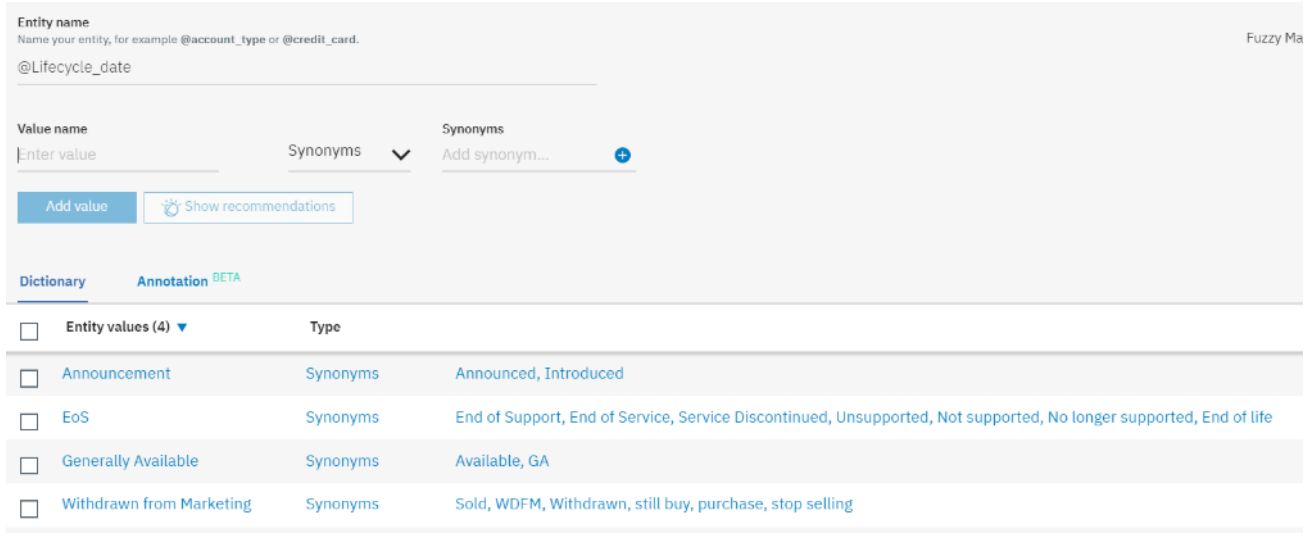
Not all decoration needs to be done by hand

Entities can be imported using CSV files and/or manually added.

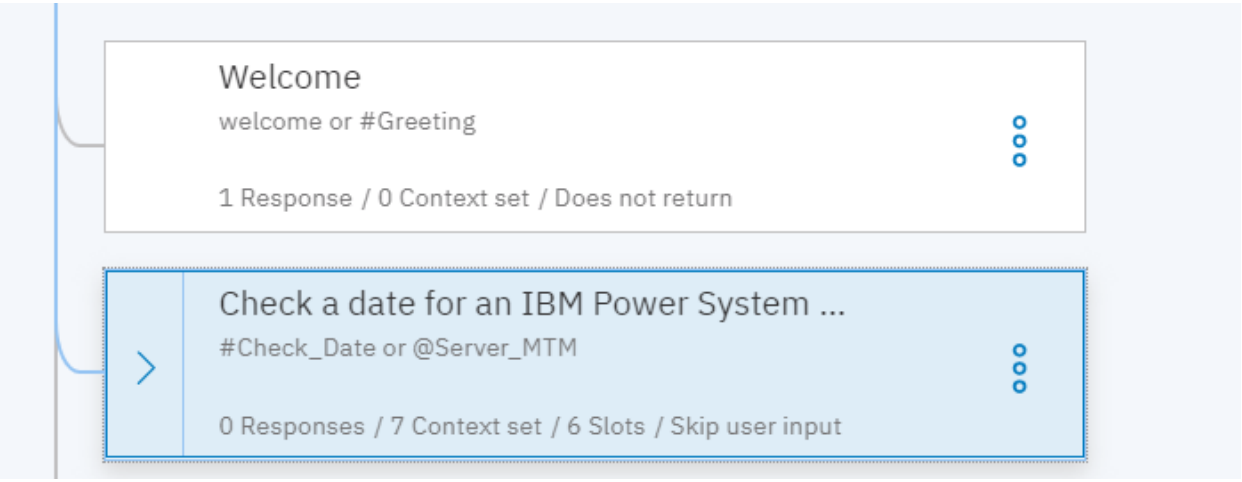


Catering for different tastes

- I came up with a range of ways to say the same thing.
- Then I had my friend test it.
- It broke, as she used other words that I had not thought of.
- So I added them.



Powerful flexibility to allow eating in many ways



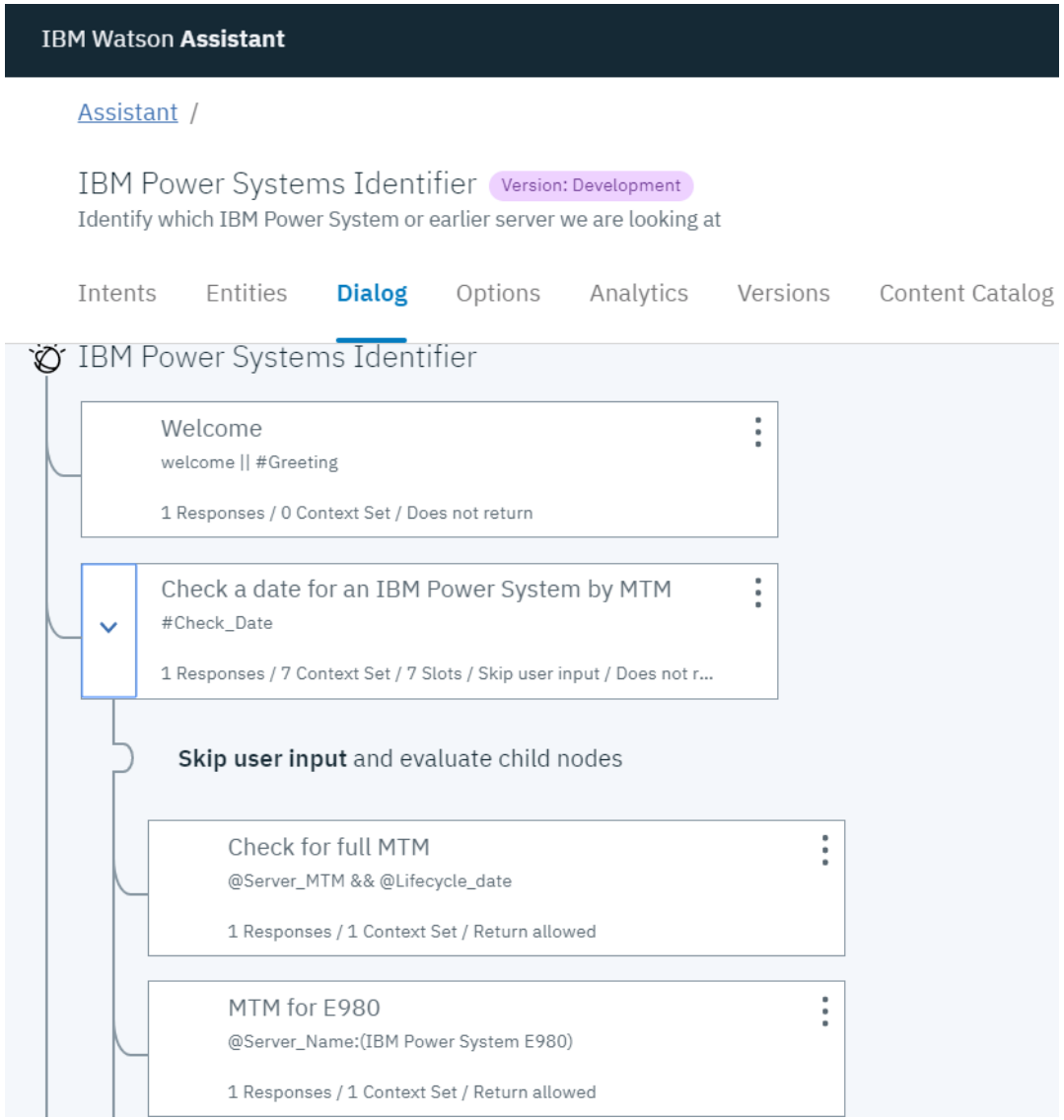
Check a date for an IBM Power System by MTM

If assistant recognizes:

#Check_Date or @Server_MTM

1	@Lifecycle_date	$Lifecycle_date	Which kind of Lifec	Required		
2	@Server_Name	$Server_Name	Enter a prompt	Optional		
3	@POWER_Gen	$POWER_Gen	Enter a prompt	Optional		
4	@Machine_Type	$Machine_Type	Enter a prompt	Optional		
5	@Machine_Model	$Machine_Model	Enter a prompt	Optional		
6	@Server_MTM	$Server_MTM	Please tell me the I	Required		

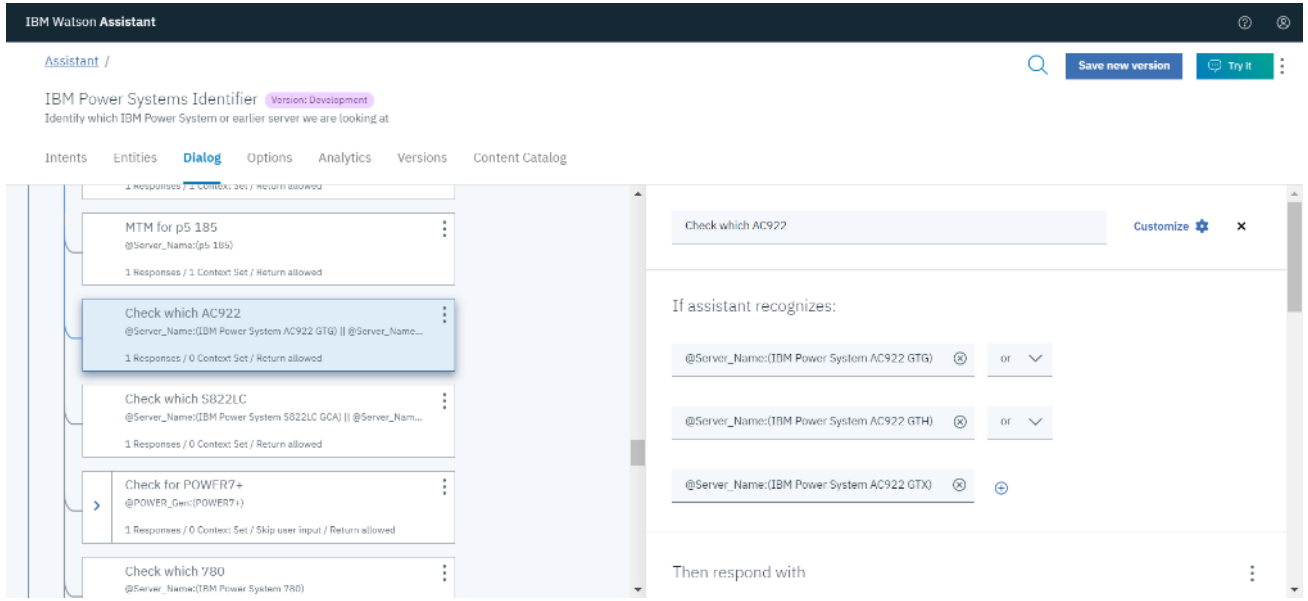
Layers in IBM Watson Assistant



- In IBM Watson Assistant, we step down through the nodes until a condition is met, then we process that node.
- To begin, a welcome, where I explain what the bot can do so far.
- Then, capture the kind of date we are looking for, and a range of other details like MTM and/or Server Name.
- If the MTM is provided, we just pass that to Node-RED. Or, if the name is unique, translate that to the right MTM, and pass that through.

Some more challenges in the bake

- If the name could be more than one MTM, we need to ask the question back to find out which one.
- Then, on the second time back into Watson Assistant, the MTM is presented and we go through nice a easily.

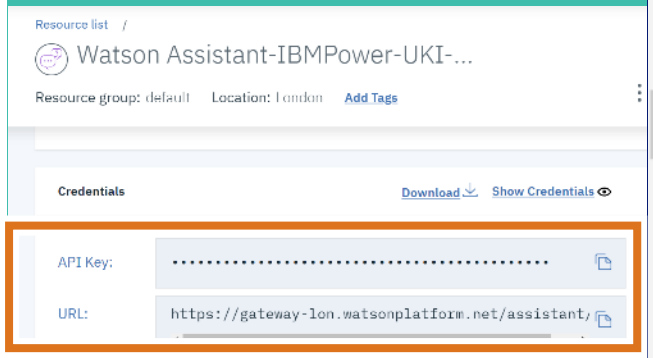


Some ingredients mix together easily

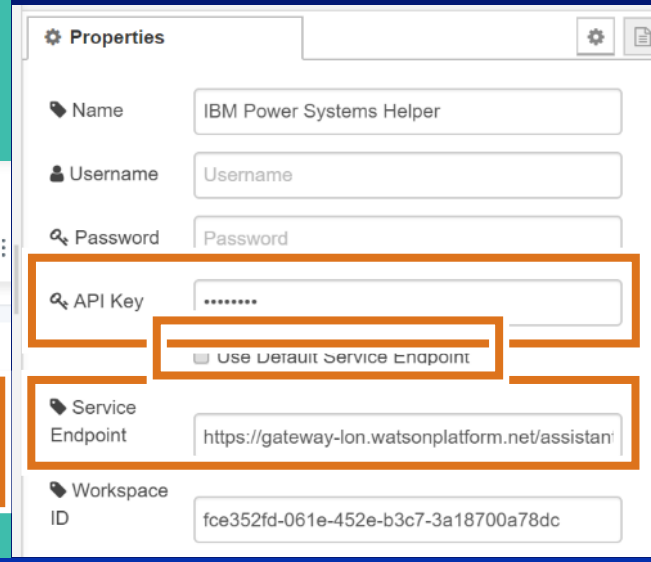
Linking the Watson Assistant Node, through the API Key and URL, results in the Intents, Context and text responses from IBM Watson Assistant just flowing into Node-RED.

One mistake I made was leaving the “Use Default Service Endpoint” clicked, but once past that, this part was very easy!

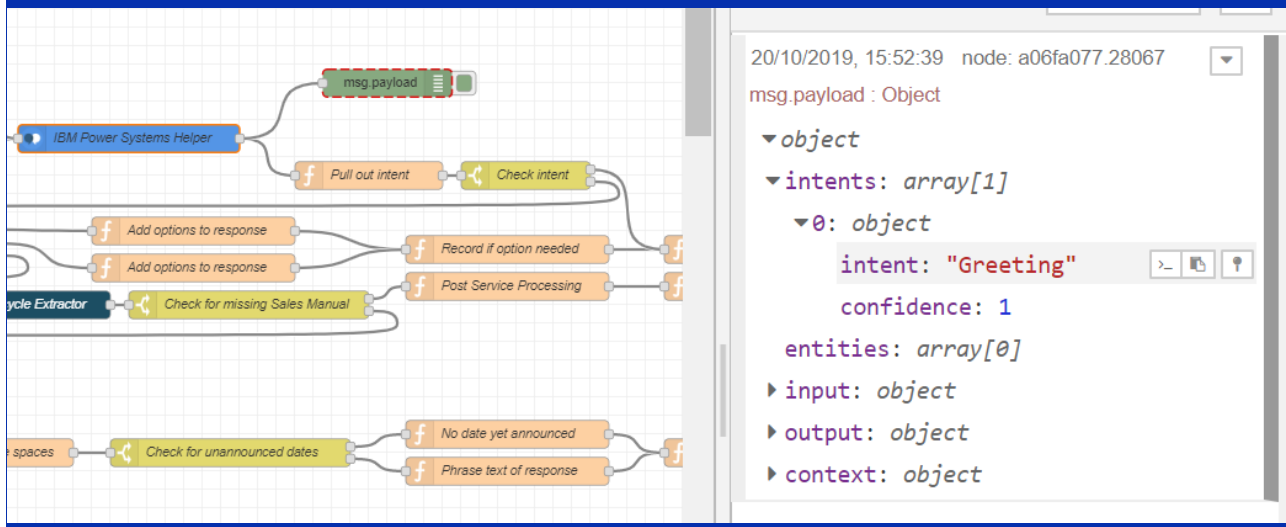
Get the API Key and URL from Resource List of IBM Watson Assistant



Paste into node



Context and Intents just flow into Node-RED



A quick taste of IBM Sales Manual Assistant

Find out more by checking on Github here:

https://github.com/DSpurway/DIS-IBM-Power-Hybrid-Cloud

My next bake brings in AIX, Db2 and James Bond!

The question I thought I would look at this time is The Curse Of The Bond Girls

https://jamesbond.fandom.com/wiki/Bond_girl_curse

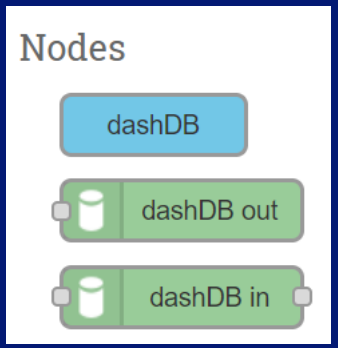
"My agent told me, 'If you take that role you'll never work again'. But what movie could you do that they'd still be interviewing you for 20 years later?"

— Lynn-Holly Johnson

We want to include AIX & IBM i in the mix!

SQL on AIX accessed with DashDB Node

https://flows.nodered.org/node/node-red-nodes-cf-sqldb-dashdb

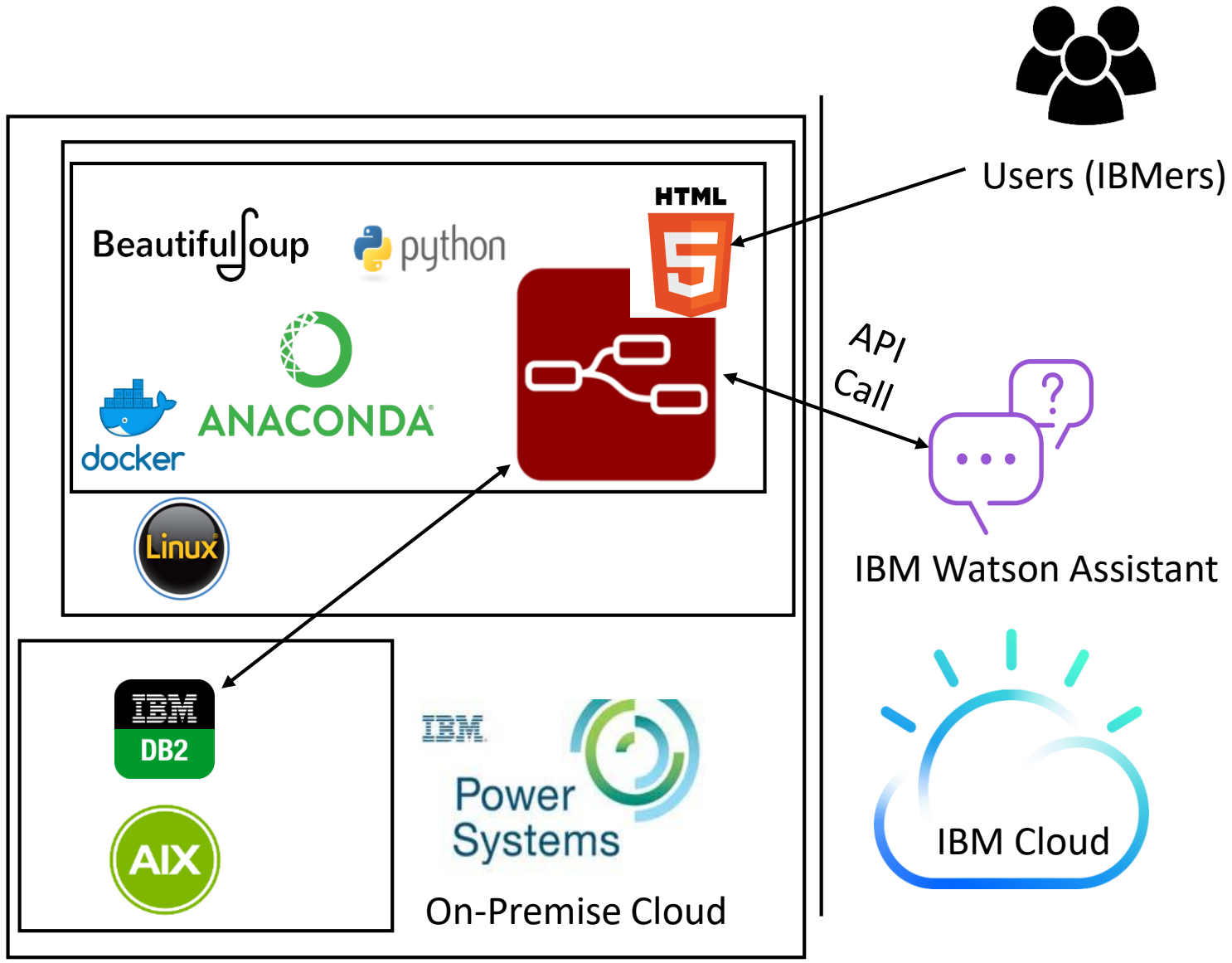


Some fun data from the Internet Movie Database

https://www.imdb.com/interfaces/

The high level recipe for the James Bond Assistant
Another Hybrid Cloud solution with AI

- IBM Db2 running in AIX
- Docker Container on Centos on IBM Power
- Anaconda to handle compatibility issues
- Node-RED calls
  - Watson Assistant
  - Python
    - Beautiful Soup
    - Urllib
  - DashDB
- Users interact with a webpage, which is part of Node-RED



Bringing in a key ingredient

Another new area to me, but standard for most AIX and IBM i implementations, was to get my database installed in the server, populated and accessible from my Container.

This may be a new technical skill for to me, but the idea is that almost all existing installs of IBM Power Systems have data like this already running. That data could be added to this kind of Hybrid Cloud solution.

What could you bake by mixing your existing data with new functions and other data sources?

Load the TSV files from IMDB initially into Db2 in the IBM Cloud, as it can help predict the schema details.

Then, LOAD, and find a lot of truncations. So, make the tables bigger and repeat!

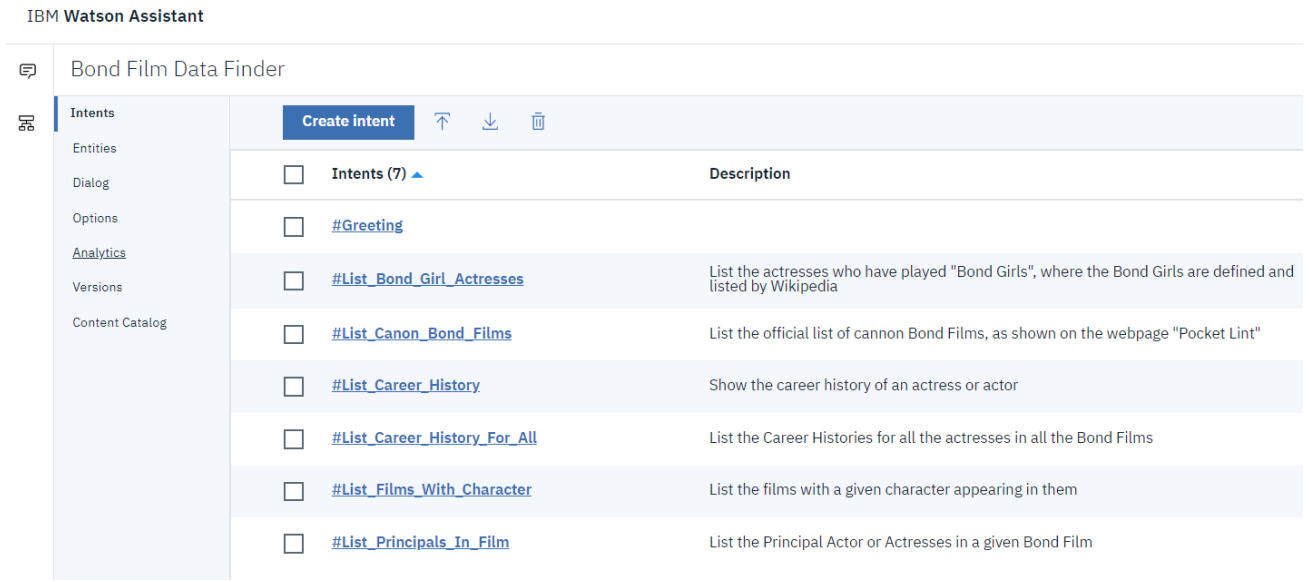
Quite a lot of challenge getting the prerequisites for the Db2 CLI installed in the Container. Ross Cruickshank and Stuart Cunliffe helped me around those issues

```
FROM ppc64le/centos

RUN yum -y install libstdc++ make gcc-c++ numactl-devel
RUN curl -sL http://public.dhe.ibm.com/software/server/POWER/Linux/xl-compiler/eval/ppc64le/rhel7/libxlc-16.1.1.3-190404a.ppc64le.rpm > libxlc-16.1.1.3-190404a.ppc64le.rpm \
    && yum -y install ./libxlc-16.1.1.3-190404a.ppc64le.rpm
RUN cd /usr/local \
    && curl -sL https://nodejs.org/dist/v8.16.1/node-v8.16.1-linux-ppc64le.tar.xz > node-v8.16.1-linux-ppc64le.tar.xz \
    && tar --strip-components 1 -xf node-v8.16.1-linux-ppc64le.tar.xz
```


Simple icing this time

This version of IBM Watson Assistant is really simple.

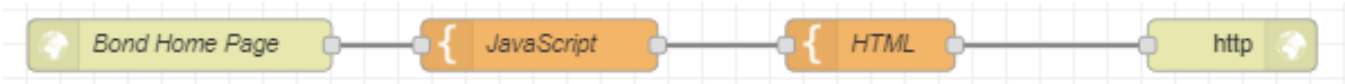
I created an Intent for each “function”, with minimal Dialog nodes to capture basic Entities/variables

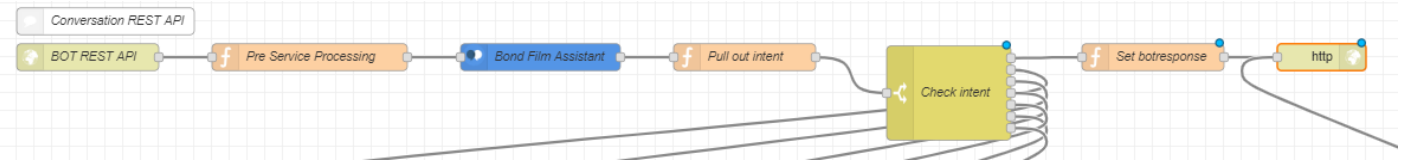


Repeated ingredients

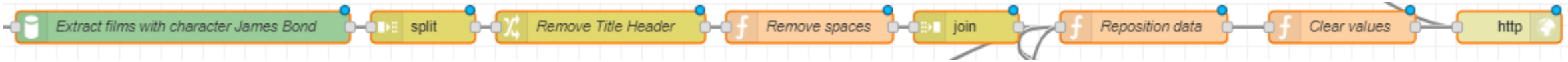
I mostly reused the same couple of nodes to give me my web page.

Then, grab the input, save some values, pass to IBM Watson Assistant to work out the Intent. Check for that Intent, and return to the webpage if just saying “Hello”…





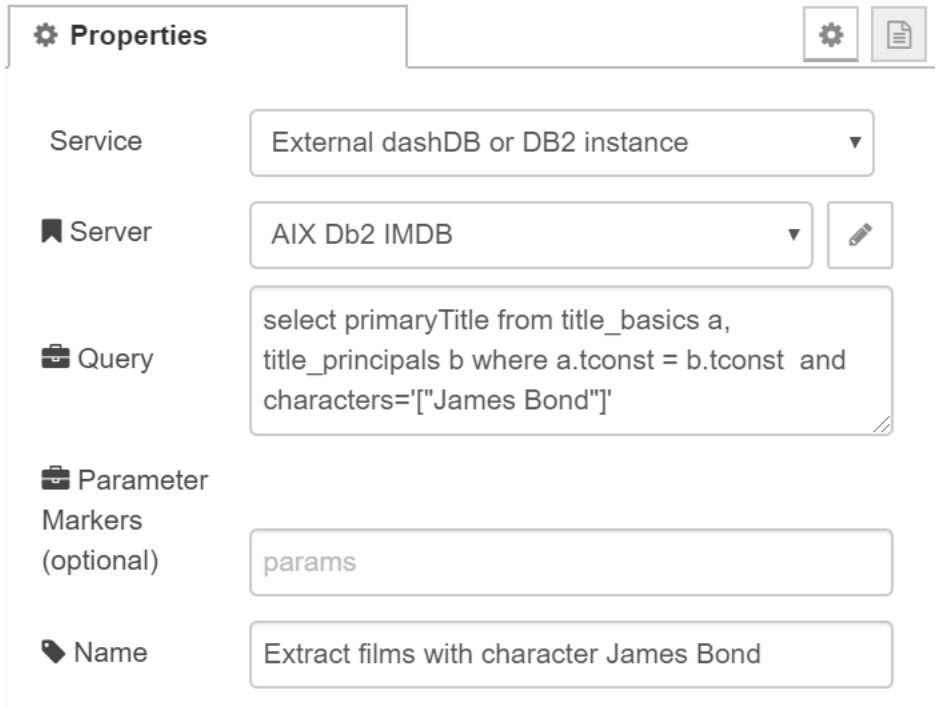
Cooking up the first “question” I want to ask



For AIX, using the DashDB node, I can pass SQL queries very easily.

The connection details for the DB are entered, then SQL can be called.

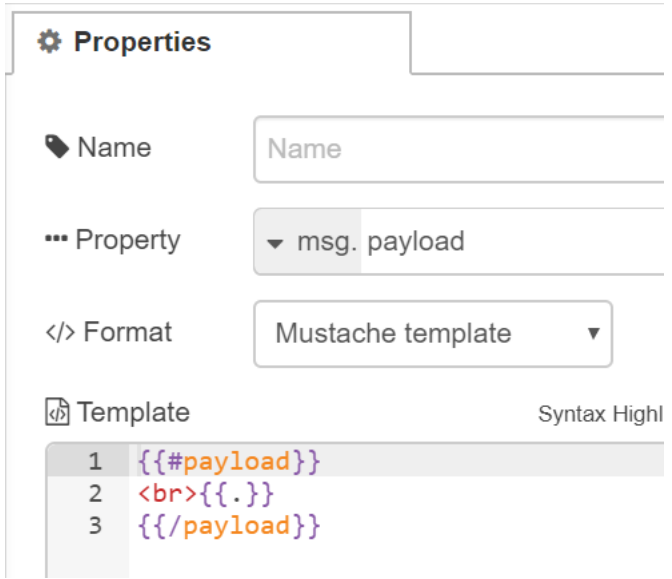
For Db2 for i, you need to pass the SQL to the Db2 for i node, which is very similar.



Using Python and Beautiful soup again to enrich existing data

Using the Intent from IBM Watson Assistant, I reused some Python and Beautiful Soup to scrape a list of the official Bond Films.

Also, a recent addition is to use a Template node to pass HTML back to my page, to simply format my results.



Trying out new methods

Ross Cruickshank recently showed me some new ways to format the results, which look much better.

In my Javascript template node, I had this:

```
// creates div for interaction with bot      function createNewDiv(who, message) {    var txt = who + ' : ' + message;    return $('<div></div>').text(txt);
```

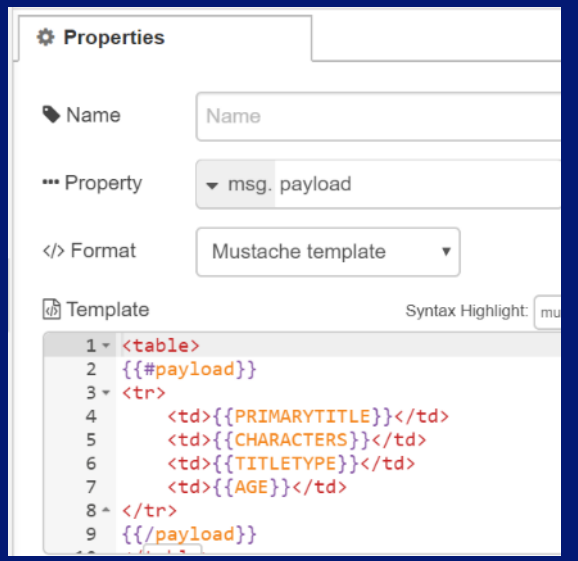
Which we changed to:

```
// creates div for interaction with bot      function createNewDiv(who, message) {    var txt = who + ' : ' + message;    return $('<div></div>').html(txt);
```

Initially, the template for the webpage just accepted text.

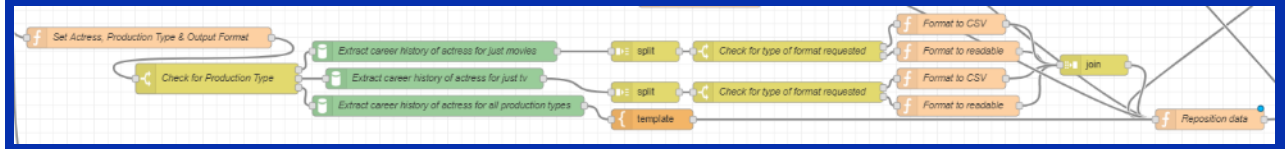
But, by changing to accept HTML, both text and HTML can be passed and displayed in the results

Passing a table format



Using Slots in IBM Watson Assistant, I can change if we want to display career histories for just Films, just TV, or both, which is the default. I was also checking for CSV or “human readable” in the question.

Just text, a table, or a CSV file are all possible.



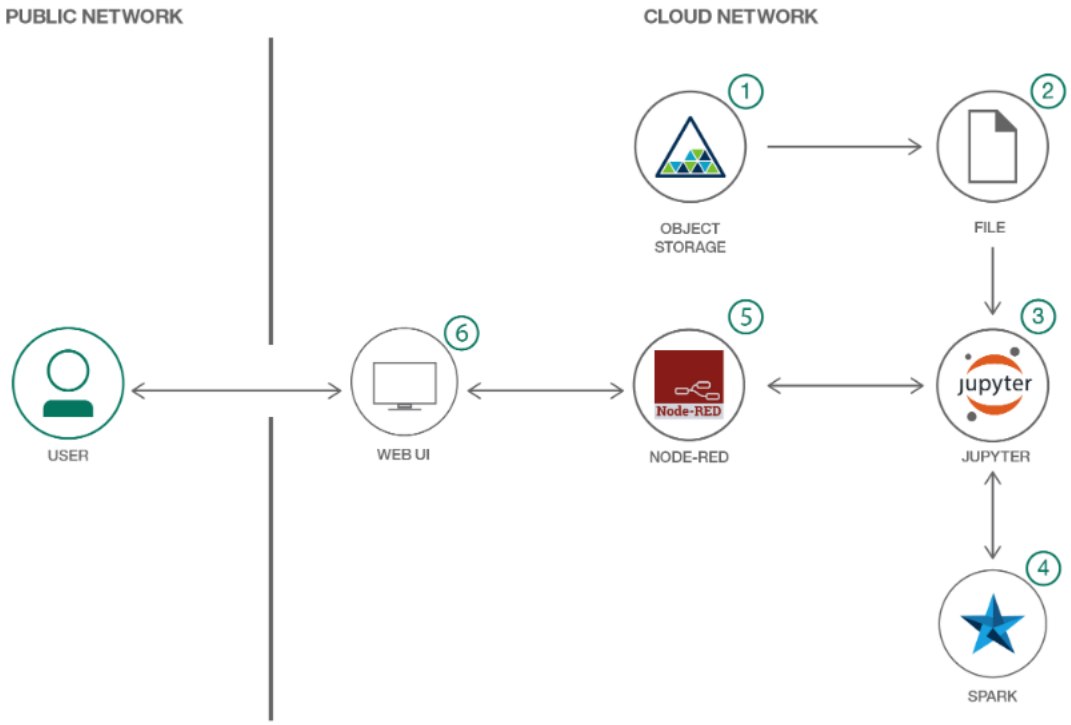
A slice of James Bond Assistant

If you want to know more about this one, ask me!

More ingredients I want to work in

"Orchestration of the analytics workflow in IBM Watson Studio using a custom web user-interface built with Node-RED"

https://github.com/IBM/node-red-dsx-workflow

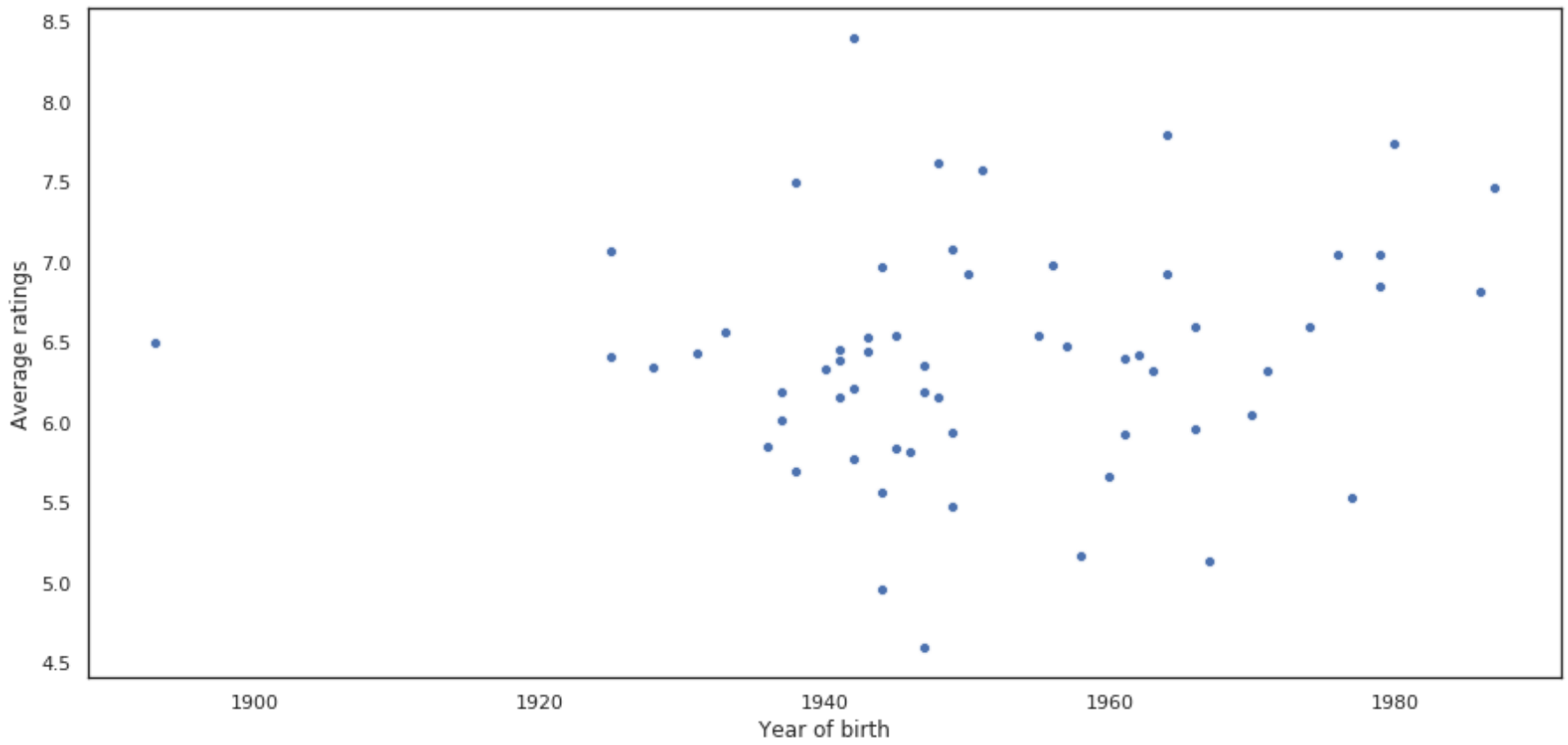


First taste of Data Science

My friend Mandie Quartly looked at whether Date of Birth had a correlation to rating of the film.

This was done with IBM Watson Studio on IBM Cloud, which could be linked to directly in a Hybrid Cloud solution.

On-premise also an option.



Trying something new can lead to more ideas

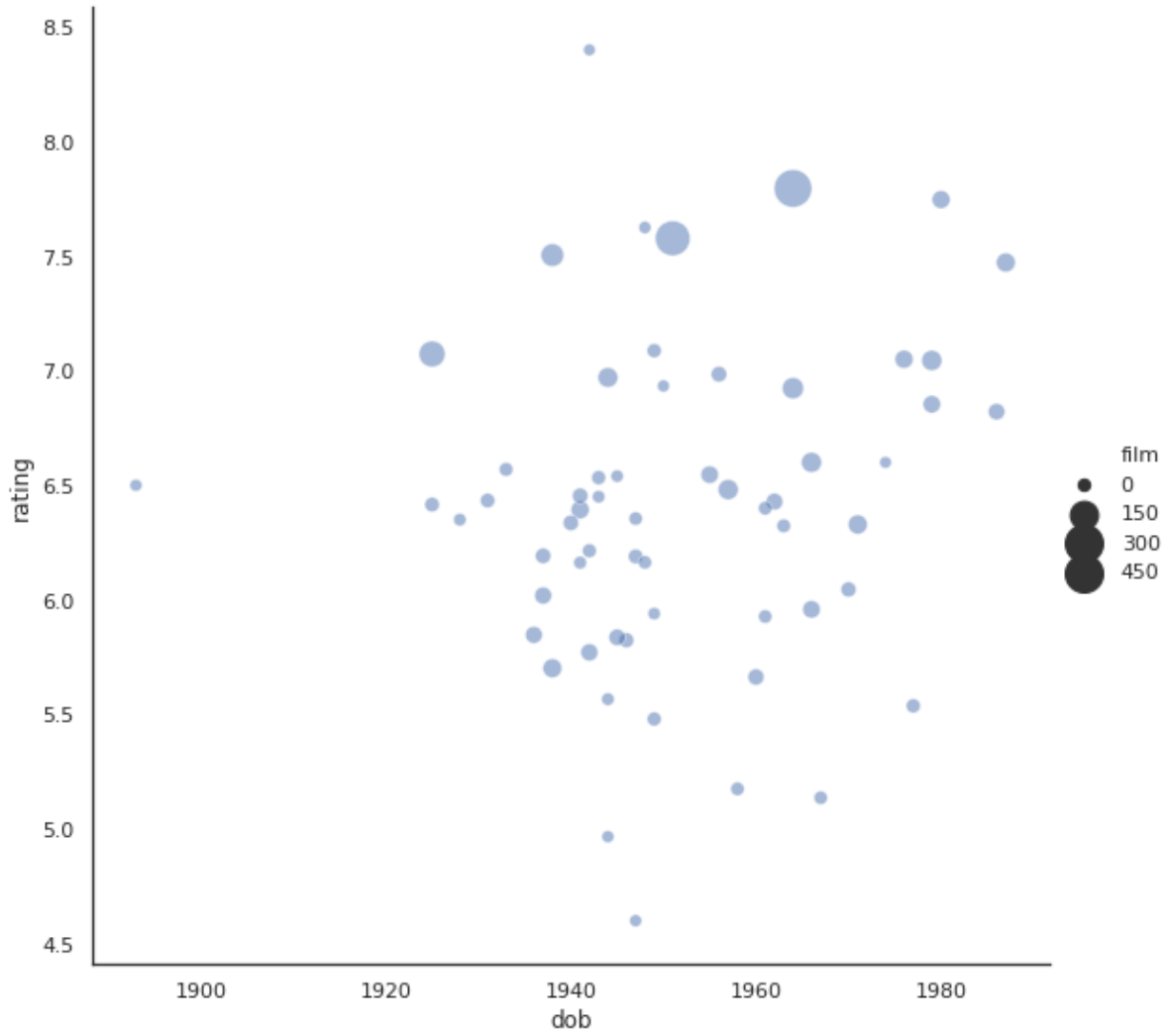
- One initial interesting result is that the more productions a “Bond Girl” has been in, the higher the “rating”.
- So, that might be a reflection on established experience, maybe?

```
In [63]: # Have Jane Seymour and Teri Hatcher really done that many films?
         ratings2[(ratings2.film > 200)]
```

Out[63]:

	name	dob	film	rating
26	Jane Seymour	1951.0	255	7.577647
55	Teri Hatcher	1964.0	302	7.795364

- Interesting catch: two of the actresses who had played “Bond Girls” had done a lot of other work
- This is because TV shows are also part of the data
- Teri Hatcher is in multiple episodes of Superman, Desperate Housewifes and many more.
- Jane Seymour is similar, with Dr. Quinn, Medicine Woman after being a “Bond Girl”.



My first “Showstopper” is still to come

- My first creation is going down well. It answers a question I get asked, and so do others, so I plan to expand it and make it more widely available.
- Time can be saved!
- Next, the IBM Inventory Records is in a DB today…
- If I can take a version of that, combine my work with the Sales Manuals, I could tell which customers have servers which are going out of support and we could help with that issue.
- I have previously worked on TCO models. I may be able to combine that with the Inventory data, to show the benefits of upgrading, for a given customer.
- Maybe, going deeper into the Feature Codes, which Features have been announced to be Withdrawn from Marketing, so we can help there too.
- Other possibilities include integrating with Slack, so BPs and maybe others could benefit from my creation.

What could you cook up?

Could you create something that enhances existing AIX or IBM i solutions?

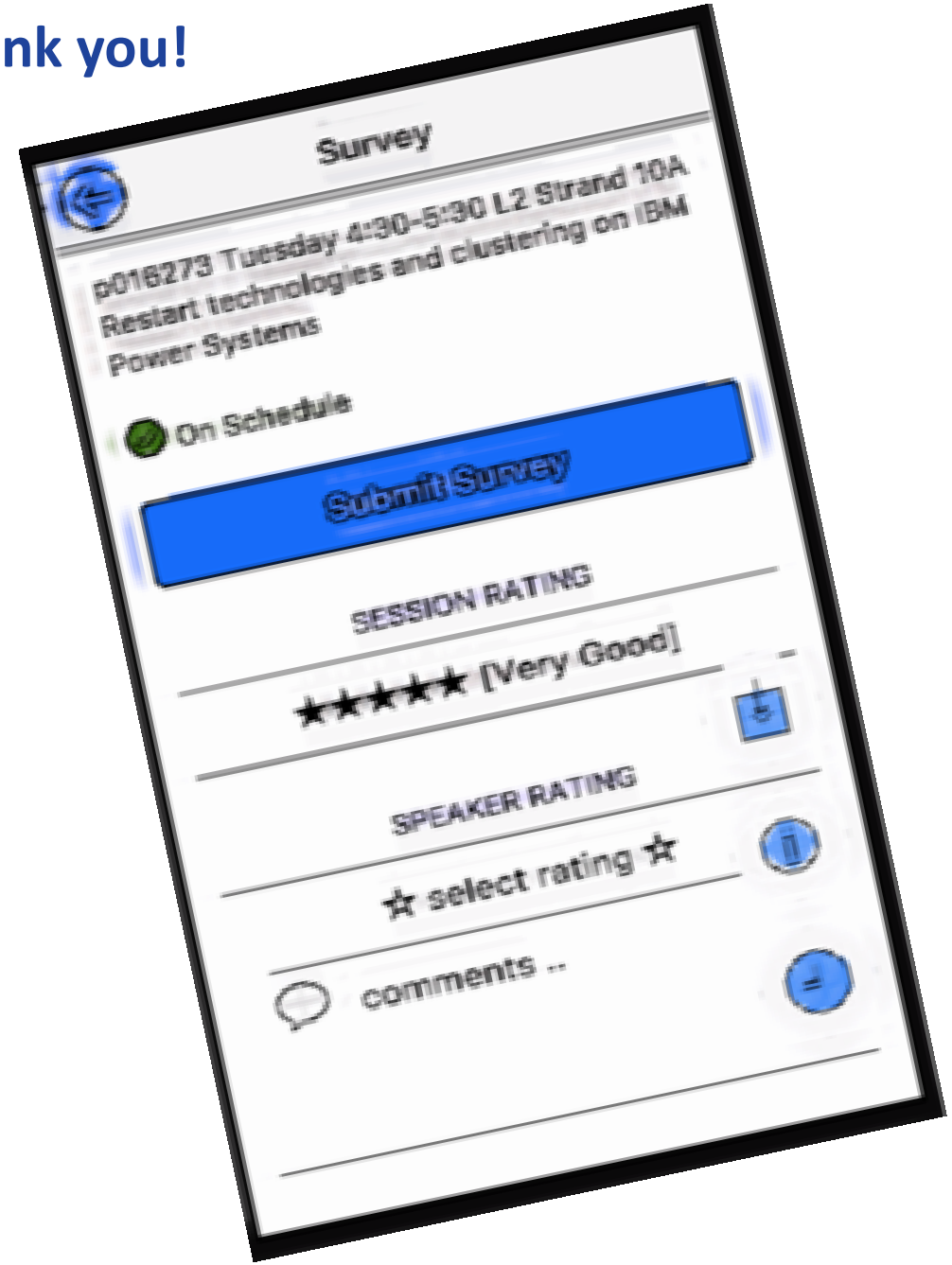
Don’t let existing data become legacy!

You don’t need to be an expert to combine AIX or IBM i data with a Conversational Interface built with AI, and create tasty results!

Session summary

- You don’t need to be an expert to do this. Anyone can cook with these ingredients!
- What started me on this journey and initial learning you can also use.
- My first creation was to answer a Frequently Asked Question. What FAQs do you see?
- The initial ingredients I used - Python, Beautiful Soup, Jupyter and Anaconda
- Lots of options available, so choose what works for you
- Node-RED is a powerful mixing bowl and IBM Watson Assistant can be the icing on the cake
- A quick taste of IBM Sales Manual Assistant
- My next bake brings in AIX, Db2 and James Bond!
- My first “Showstopper” is still to come
- What could you cook up?

Thank you!



David Spurway
IBM Power Systems CTO, UK & Ireland

Mobile: +44 7717 892 896

Email: david.Spurway@uk.ibm.com

Please complete the Session Evaluation!

Notices and disclaimers

- © 2019 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those
- customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml

