XML & JSON For Total Beginners

Katie Kodes

Database Jill-Of-All-Trades

@KatieKodes

katiekodes.com/forcelandia-2019/

KEEP CODE WEIRD

FORCELANDIA

< />

PUT A CLOUD ON IT

What are XML & JSON?

Punctuation standards
for giving structure and meaning
to data, using
plain text

*** The “CSV” file standard is, too!*

Data Complexity

Which do you prefer? Why?

Table?

ID	Date	Vendor	Category	Price
0001	01/01/2010	Apple	Office Supplies	$60,000.00
0002	01/01/2010	Applebee’s	Meals	$50.87
0003	01/09/2010	Apple	Repairs	$928.20
0004	01/24/2010	Steelcase	Office Supplies	$20,289.98

```
"ID","Date","Vendor","Category","Price"
"0001","01/01/2010","Apple","Office Supplies","$60,000.00"
"0002","01/01/2010","Applebee's","Meals","$50.87"
"0003","01/09/2010","Apple","Repairs","$928.20"
"0004","01/24/2010","Steelcase","Office Supplies","$20,289.98"
```

Bulleted List?

- **Record: 0001**
  - **Date:** 01/01/2010
  - **Vendor:** Apple
  - **Category:** Office Supplies
  - **Price:** $60,000.00
- **Record: 0002**
  - **Date:** 01/01/2010
  - **Vendor:** Applebee’s
  - **Category:** Meals
  - **Price:** $50.87
- **Record: 0003**
  - **Date:** 01/09/2010
  - **Vendor:** Apple
  - **Category:** Meals
  - **Price:** $50.87
- **Record: 0004**
  - **Date:** 01/24/2010
  - **Vendor:** Steelcase
  - **Category:** Office Supplies
  - **Price:** $20,289.98

Which do you prefer? Why?

Table?

Name	Bday	Kid 1	Kid 2	Job	Food 1	Food 2	Collection
Hani	Nov. 8	Johnny (4)	Matilda (2)	nurse			
Dan	Jan. 27				wine	pickles	
Ridhi	Sep. 16						Frogs

```
"Name","Bday","Kid 1","Kid 2","Job","Food 1","Food 2","Collection"
"Hani","Nov. 8","Johnny (4)","Matilda (2)","nurse","","",""
"Dan","Jan. 27","","","","wine","pickles",""
"Ridhi","Sep. 16","","","","","","Frogs"
```

Bulleted List?

- **Hani**
  - **Bday:** Nov. 8
  - **Kids:**
    - Johnny (4)
    - Matilda (2)
  - **Job:** Nurse
- **Dan**
  - **Bday:** Jan. 27
  - **Foods:**
    - Wine
    - Pickles
- **Ridhi**
  - **Bday:** Sep. 16
  - **Collection:** Frogs

Choose a standard for the shape

Table-Shaped Data

- **Consistent** “keys”
- **Flat**

Optimal: CSV / spreadsheets

List-Shaped Data

- **Varied** “keys”
- **Nested**

Optimal: XML / JSON / bullets

My friends as XML: *one* possibility

```
<AllFriends>
    <friend name="Hani" bday="Nov. 8">
         <kid name="Johnny">
            <age>4</age>
         </kid>
         <kid name="Matilda">
            <age>2</age>
         </kid>
         <job>
            nurse
         </job>
    </friend>
    <friend name="Dan" bday="Jan. 27">
         <food>
            wine
         </food>
         <food>
            pickles
         </food>
    </friend>
    <friend name="Ridhi" bday="Sep. 16">
         <collection>
            frogs
         </collection>
    </friend>
</AllFriends>
```

My friends as JSON: *one* possibility

```
[
   {
      "Name" : "Hani",
      "Bday" : "Nov. 8",
      "Kids" :
         [
            {
               "Name" : "Johnny",
               "Age" : 4
            },
            {
               "Name" : "Matilda",
               "Age" : 2
            }
         ],
      "Job" : "nurse"
   },
   {
      "Name" : "Dan",
      "Bday" : "Jan. 27",
      "Food" : ["wine","pickles"]
   },
   {
      "Name" : "Ridhi",
      "Bday" : "Sep. 16",
      "Collection" : "frogs"
   }
]
```

- Hani
  - **Bday:** Nov. 8
  - **Kids:**
    - Johnny (4)
    - Matilda (2)
  - **Job:** Nurse
- Dan
  - **Bday:** Jan. 27
  - **Foods:**
    - Wine
    - Pickles
- Ridhi
  - **Bday:** Sep. 16
  - **Collection:** Frogs

Line breaks & whitespace ***optional*** (for human convenience only)

XML vs. JSON: Which is “better?”

XML easier for…

- configuration files?
- human eyes? 👀 *(words = “bookmarks”)*

```xml
<friend>
 <name>Dan</name>
 <bday>Jan. 27</bday>
 <food>wine</food>
 <food>pickles</food>
</friend>
```

JSON easier for…

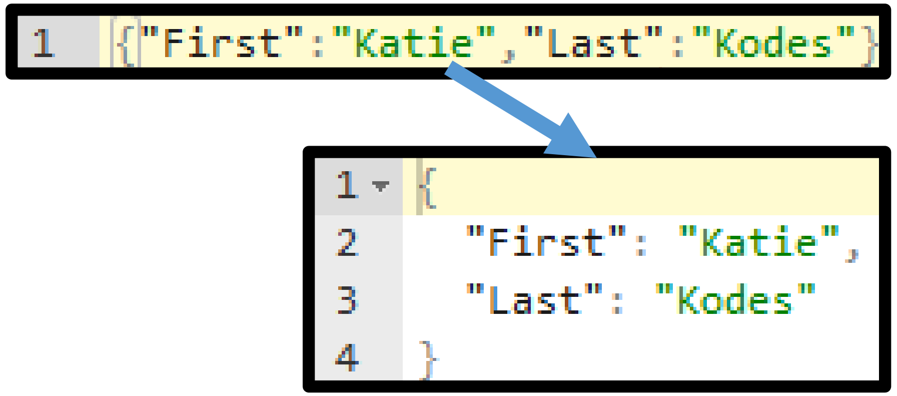
- simple data sets? *(less “clutter”)*
- coding? 💻 *(simpler rules = simpler code)*

```json
{
   "name" : "Dan",
   "bday" : "Jan. 27",
   "food" : ["wine","pickles"]
}
```

“wine” 🍷	JSON	XML
Apex	`myFriend.food[0]`	`myFriend.getChildElement('food',null).getText()`
Python	`myFriend['food'][0]`	`myFriend.find('food').text`

“pickles”	JSON	XML
Apex	`myFriend.food[1]`	*(too many lines to show!)*
Python	`myFriend['food'][1]`	`myFriend.findall('food')[1].text`

♪ I'm So Pretty… ♪



```
{"First":"Katie","Last":"Kodes"}
```

```
{
  "First": "Katie",
  "Last": "Kodes"
}
```

XML or JSON is all on one line?

"Beautify" it!

For XML & JSON you're okay sharing with strangers…

https://codebeautify.org/xmlviewer & https://codebeautify.org/jsonviewer

For your private / corporate XML & JSON…

Notepad++ text editing software plus "Tidy2" (XML) & "JSTool" plugins

Not So Bad!

📖 *(https://katiekodes.com/intro-xml-json-1/)* 📖

So … how does Salesforce use XML & JSON?

Disclaimer

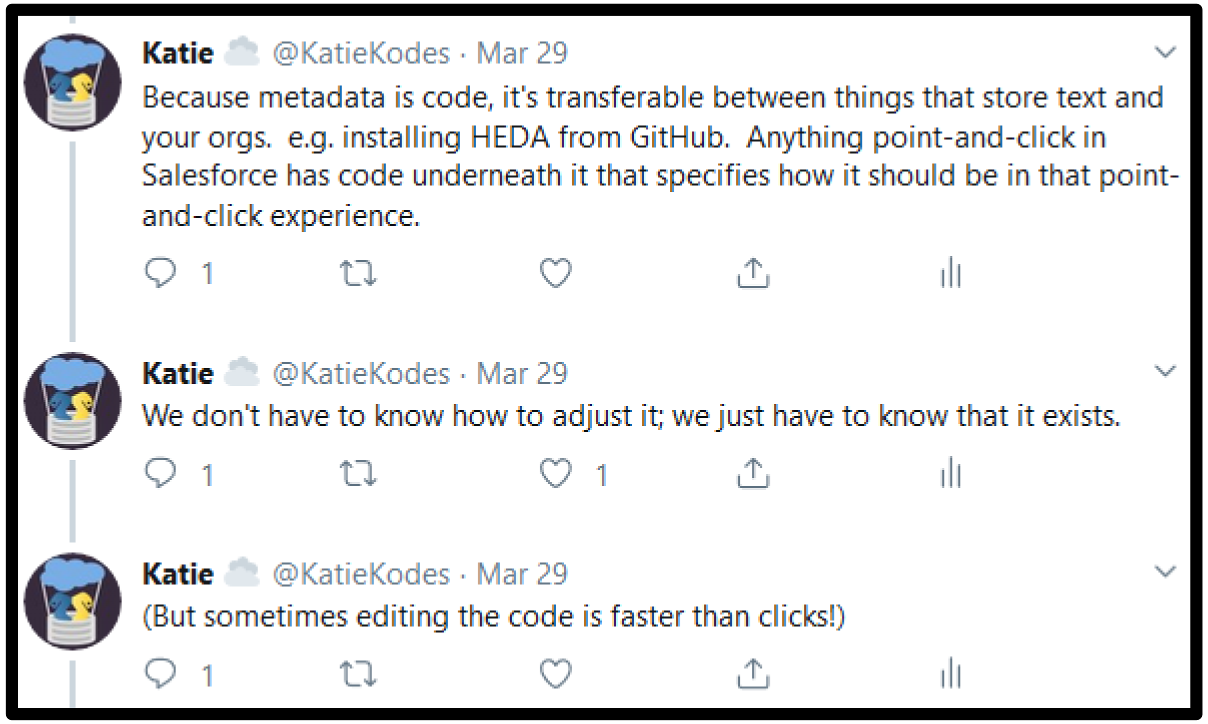
- XML and JSON are ***beginner-friendly*** to read & write!
  - https://katiekodes.com/intro-xml-json-1/

- Doing anything ***useful*** with them in ***Salesforce*** may require ***partnering*** with someone who can program.

#1: Salesforce “Metadata”

Tweak the definition of a...

- **Custom Object**
- **Flow**
- **Workflow**
- **Report**
- **Custom Report Type**
- **...**

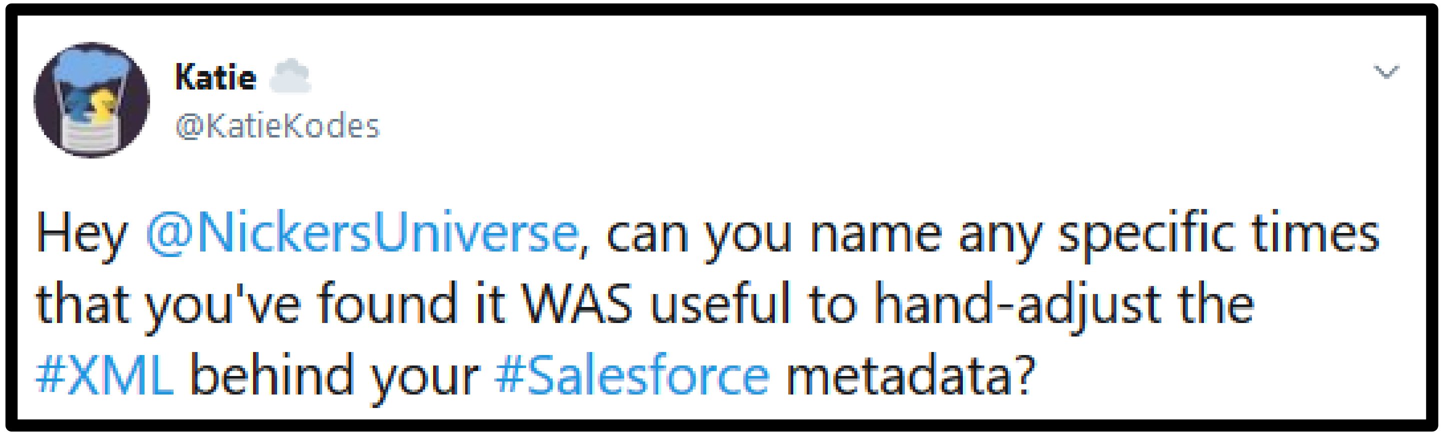


Katie @KatieKodes · Mar 29
Because metadata is code, it's transferable between things that store text and your orgs. e.g. installing HEDA from GitHub. Anything point-and-click in Salesforce has code underneath it that specifies how it should be in that point-and-click experience.
1

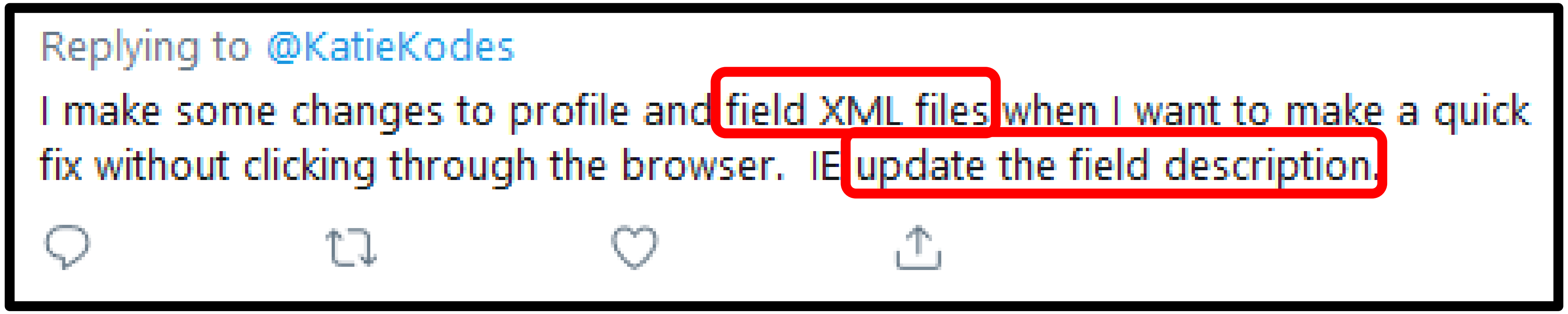
Katie @KatieKodes · Mar 29
We don't have to know how to adjust it; we just have to know that it exists.
1 1

Katie @KatieKodes · Mar 29
(But sometimes editing the code is faster than clicks!)
1

(Live-tweets of a talk by @NickersUniverse)

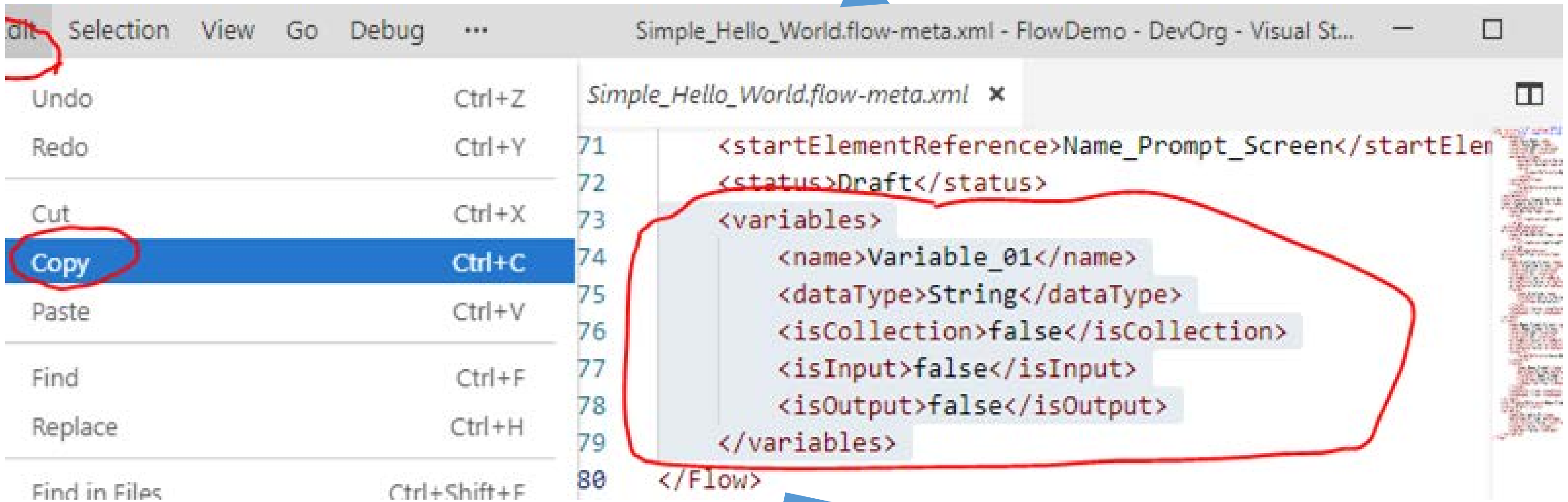

Katie
@KatieKodes

Hey @NickersUniverse, can you name any specific times that you've found it WAS useful to hand-adjust the #XML behind your #Salesforce metadata?


Replying to @KatieKodes

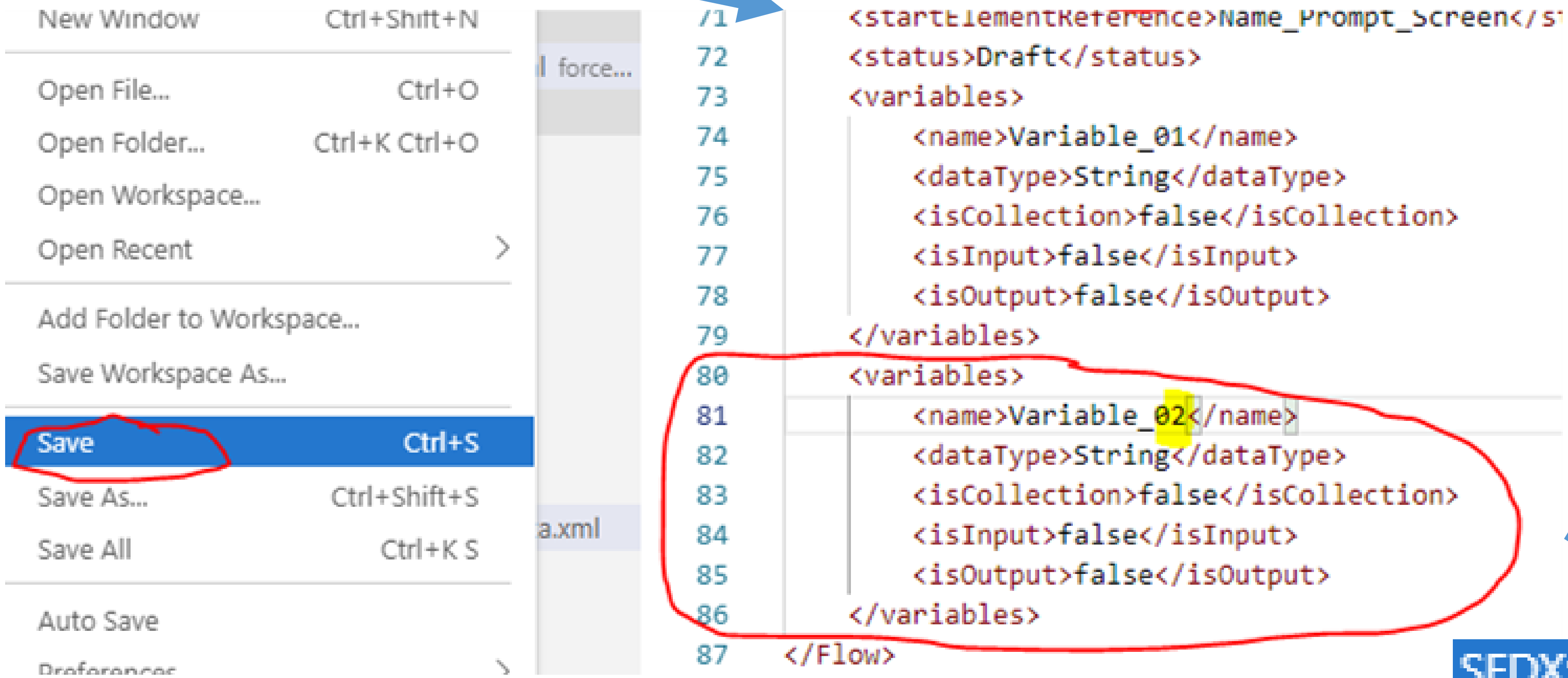
I make some changes to profile and field XML files when I want to make a quick fix without clicking through the browser. IE update the field description.

SFDX: Retrieve Source in Manifest from Org



dit
Selection
View
Go
Debug
···
Simple_Hello_World.flow-meta.xml - FlowDemo - DevOrg - Visual St...
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Find Ctrl+F
Replace Ctrl+H
Find in Files Ctrl+Shift+F
Simple_Hello_World.flow-meta.xml

```
    <startElementReference>Name_Prompt_Screen</startElen
    <status>Draft</status>
    <variables>
        <name>Variable_01</name>
        <dataType>String</dataType>
        <isCollection>false</isCollection>
        <isInput>false</isInput>
        <isOutput>false</isOutput>
    </variables>
</Flow>
```



```
    <startElementReference>Name_Prompt_Screen</s
    <status>Draft</status>
    <variables>
        <name>Variable_01</name>
        <dataType>String</dataType>
        <isCollection>false</isCollection>
        <isInput>false</isInput>
        <isOutput>false</isOutput>
    </variables>
    <variables>
        <name>Variable_02</name>
        <dataType>String</dataType>
        <isCollection>false</isCollection>
        <isInput>false</isInput>
        <isOutput>false</isOutput>
    </variables>
</Flow>
```

KEEP CODE WEIRD
FORCELANDIA
PUT A CLOUD ON IT

SFDX: Deploy Source to Org

Just Add Code – Salesforce Metadata

Dev ideas

1. Re-alphabetize the fields in a **custom report type**
2. Create files for **similar fields**, *and* **profile permissions**, in a single script *(e.g. "Custom18__c" → "Custom32__c")*
3. ______________________________: Admins *and* Devs – Propose an **idea** or ask if it's possible during **Q&A**!

#2: Because Pardot Says So

Want to bulk-edit Pardot Prospects?

There’s no “data loader” that exports/imports CSV.

But there is an “HTTPS API” that exports/imports JSON/XML.

Let’s talk through a case where I want to find addresses that say the word “null” and fix them to actually be blank.

id	**address_one**
0101010101	123 Sunny St
0404040404	null
0505050505	
0707070707	null
0808080808	456 Cloudy
0202020202	

id	**address_one**
0404040404	
0707070707	

Architecture for Admins: ETL with APIs

Instead of this…

id	address_one
0101010101	123 Sunny St
0404040404	null
0505050505	
0707070707	null
0808080808	456 Cloudy
0202020202	

id	address_one
0404040404	
0707070707	

We can design this, now that we know JSON is easy!

```
[
  {"id": 01010101, "address_one": "123 Sunny St"},
  {"id": 04040404, "address_one": "null"},
  {"id": 05050505},
  {"id": 07070707, "address_one": "null"},
  {"id": 08080808, "address_one": "456 Cloudy"},
  {"id": 02020202}
]
```

```
[
  {"id": 04040404, "address_one": ""},
  {"id": 07070707, "address_one": ""}
]
```

(Alas, there's no ***Excel*** *for editing* ***JSON****. Developer not included.)*

#3: Marketing Cloud

Same “data loader” idea as Pardot’s API. *(Mostly XML, some JSON.)*

Admins, YOU CAN HELP proofread the dev’s XML against
<u>your</u> data to get the API working at all!

(Tell a dev “SOAP API” and watch them roll their eyes)

The XML just to *ask* for emails in a list…

Devs need admins who aren’t scared of XML as architects!

```
 ...(more here)...
<s:Body ...(more here)...>
  <RetrieveRequestMsg ...(more here)...>
    <RetrieveRequest>
      <ObjectType>
        DataExtensionObject[2019_ApplicationReminder_020719]
      </ObjectType>
      <Properties>Email</Properties>
      <Properties>Last Name</Properties>
      <Properties>Mailing First Name</Properties>
    </RetrieveRequest>
  </RetrieveRequestMsg>
</s:Body>
 ...(more here)...
```

Together, we deciphered the ~~*spell book*~~ docs & found the good stuff!

My XML+JSON Learning Journey

1. **Salesforce Metadata Edits** (fields, report types…)
    - Reading XML at all
    - VSCode Salesforce plugins
    - Editing XML w/ code
2. **Pardot DIY Data Loader**
    - Reading JSON at all
    - Editing JSON w/ code
    - Talking to "APIs" w/ clicks ("Postman") & code
3. **MarketingCloud Automation Experiments**
    - Teaching admins XML

Lesson Plans

#AwesomeAdmin

1. Set up VSCode, edit a flow or field description as XML
2. Read my full guide to reading & writing XML & JSON
    - https://katiekodes.com/intro-xml-json-1/
3. Share your triumphs with a dev! *(And me! @KatieKodes)*
    1. Celebrate
    2. Speculate what business problems you’re now wondering if XML & JSON might be part of a solution for
    3. If they know XML & JSON, ask them for your questions

#SFDCDevs & #Admineloper

1. Practice **editing** XML and/or JSON files with code
2. Practice downloading and uploading XML and JSON files over **APIs** using **HTTP requests** with code
   - *Use "Postman" software to try it point-and-click first*
   - https://github.com/public-apis/public-apis
3. Repeat steps 2 & 3, only with **real** Salesforce-land APIs
4. Share your triumphs with an admin! *(And me! @KatieKodes)*
   1. Celebrate
   2. Ask if they have relevant business problems

Questions? Ideas?

1. XML & JSON are punctuation standards for ***<u>data</u>***.
2. Both allow for complicated files like “lists of lists.”
3. Complexity → used in complex Salesforce contexts.

https://katiekodes.com/forcelandia-2019/

+ This #SalesforceSaturday on ***Apex Hours***! 🌅 10 *AM* EDT / 7 *AM* PDT 🌅

Thank you!

https://katiekodes.com/forcelandia-2019/

Also, catch me Saturday on Apex Hours!