XML and JSON Programming (Part 2)

Robert M. Dondero, Ph.D.
Princeton University

Objectives

- We will cover:
  - XML
  - XML programming: DOM
  - XML programming: SAX
  - JSON
  - JSON programming
  - XML/JSON and AJAX

Agenda

- **JSON**
- JSON programming
- XML/JSON and AJAX

JSON

- ***JSON (JavaScript Object Notation)***
  - An alternative to XML for data storage and comm
  - Like XML:
    - Textual; human readable
    - Hierarchical
  - Unlike XML:
    - Derived from JavaScript array and object (associative array) notation

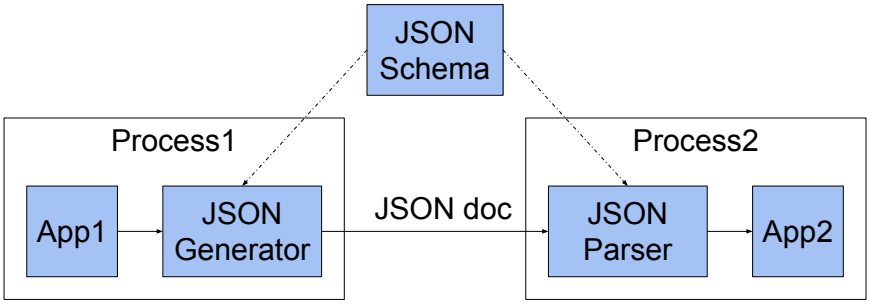
JSON

- ***JSON document***
  - A source code expression of a JavaScript data structure
    - JavaScript data structure can consist of:
      - **Strings**, **Numbers**, **Booleans**, or **null**
      - **Objects** or **arrays** having properties that are Strings, Numbers, Booleans, null, objects, or arrays

JSON: Example

- See **books.json**
  - Note: a JSON doc is a source code expression of a JavaScript data structure

JSON: Data Comm



JSON is convenient for data comm

Can use a **JSON Schema** to define the comm protocol

JSON: Example Programs

- Examples in this lecture:
  - In Python
    - Appropriate for JSON programming in the **server-side** of a Web app
  - In JavaScript
    - Appropriate for JSON programming in the **client-side** of a Web app (i.e., in a browser)

Agenda

- JSON
- **JSON programming**
- XML/JSON and AJAX

JSON Pgmming: Examples

- **writebooksjson** programs
  - The job:
    - Write all books in books.json

JSON Pgmming: Examples

- See **writebooksjson.py**
  - In Python, converting JSON to Python is easy
    - `python_object = json.loads(json_str)`

JSON	Python
array	list
object	dict
Number	int, float
String	str
Boolean	bool
null	None

```
$ python writebooksjson.py
Author: Kernighan
Title: The Practice of Programming
Price: 40.74 dollars

Author: Kernighan
Title: The C Programming Language
Price: 24.99 dollars

Author: Sedgewick
Title: Algorithms in C
Price: 61.59 dollars

$
```

JSON Pgmming: Examples

- See **writebooksjson.js**
  - In JavaScript, converting JSON to JavaScript is easy
    - `javaScriptDs= JSON.parse(jsonStr)`

```
$ node writebooksjson.js
Author: Kernighan
Title: The Practice of Programming
Price: 40.74 dollars

Author: Kernighan
Title: The C Programming Language
Price: 24.99 dollars

Author: Sedgewick
Title: Algorithms in C
Price: 61.59 dollars

$
```

JSON Pgmming: Examples

- **roundtrip** programs
  - The job:
    - Translate from JSON to data structure
    - Translate from data structure to JSON

JSON Pgmming: Examples

- See **roundtripjson.py**
  - Converting Python to JSON is easy
    - `json_str = json.dumps(python_object)`

```
$ python roundtripjson.py
[{"author": "Kernighan", "title": "The
Practice of Programming", "price": 40.74,
"currency": "dollars"}, {"author":
"Kernighan", "title": "The C Programming
Language", "price": 24.99, "currency":
"dollars"}, {"author": "Sedgewick", "title":
"Algorithms in C", "price": 61.59, "currency":
"dollars"}]
$
```

JSON Pgmming: Examples

- See **roundtripjson.js**
  - Converting JavaScript to JSON is easy
    - `jsonStr = JSON.stringify(javaScriptDs)`

```
$ node roundtripjson.js
[{"author":"Kernighan","title":"The Practice of
Programming","price":40.74,"currency":"dollars"
},{"author":"Kernighan","title":"The C
Programming
Language","price":24.99,"currency":"dollars"},{
"author":"Sedgewick","title":"Algorithms in
C","price":61.59,"currency":"dollars"}]
$
```

Aside: JSON vs. XML

- XML pros
  - Appropriate for data comm **and publishing**
- JSON pros
  - Compact
  - Easy to handle in JavaScript
  - Easy to handle in Python, …
  - OK to handle in Java, …

Aside: Data Comm Formats

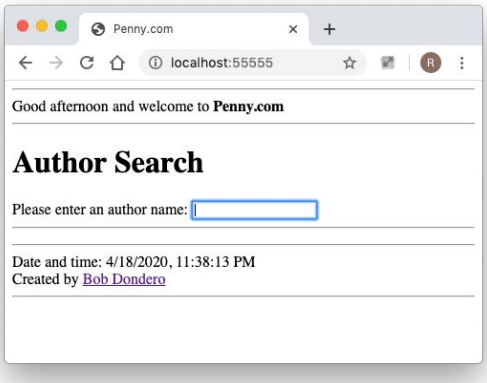
- Data comm formats
  - Binary:
    - Pickled Python objects, …
  - Human readable:
    - plain text, HTML, XML, JSON, …
- See:
  - https://en.wikipedia.org/wiki/Comparison_of_data_serialization_formats

Agenda

- JSON
- JSON programming
- **XML/JSON and AJAX**

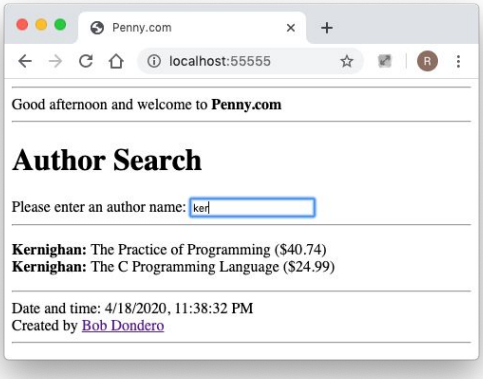
XML/JSON and AJAX

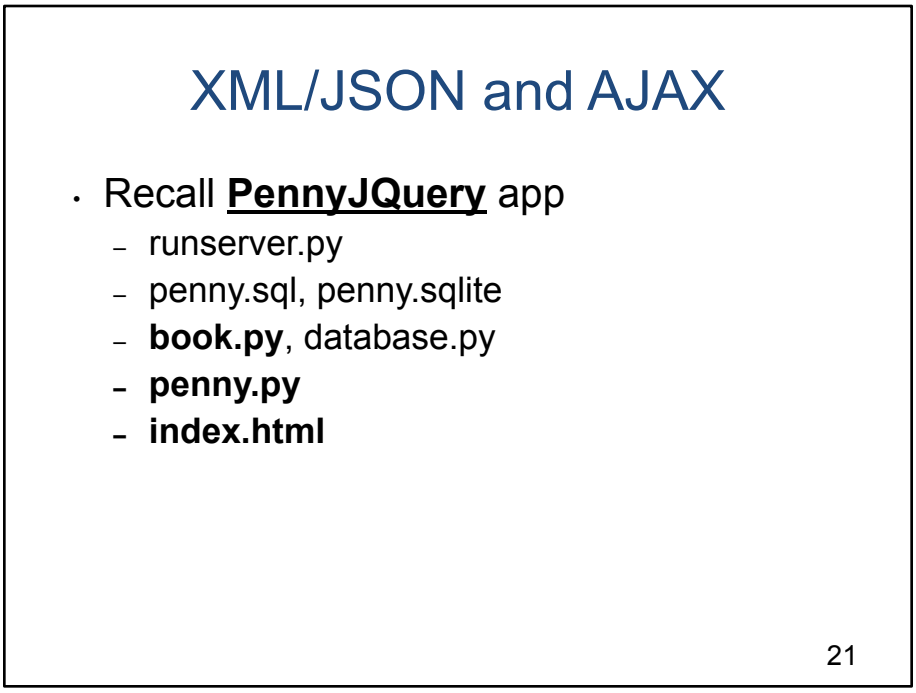
- Recall **PennyJQuery** app



XML/JSON and AJAX

- Recall **PennyJQuery** app





XML/JSON and AJAX

[see slide]

Code notes:

book.py

Defines to_tuple() method

We've been using all along

penny.py

Sends response as HTML fragment

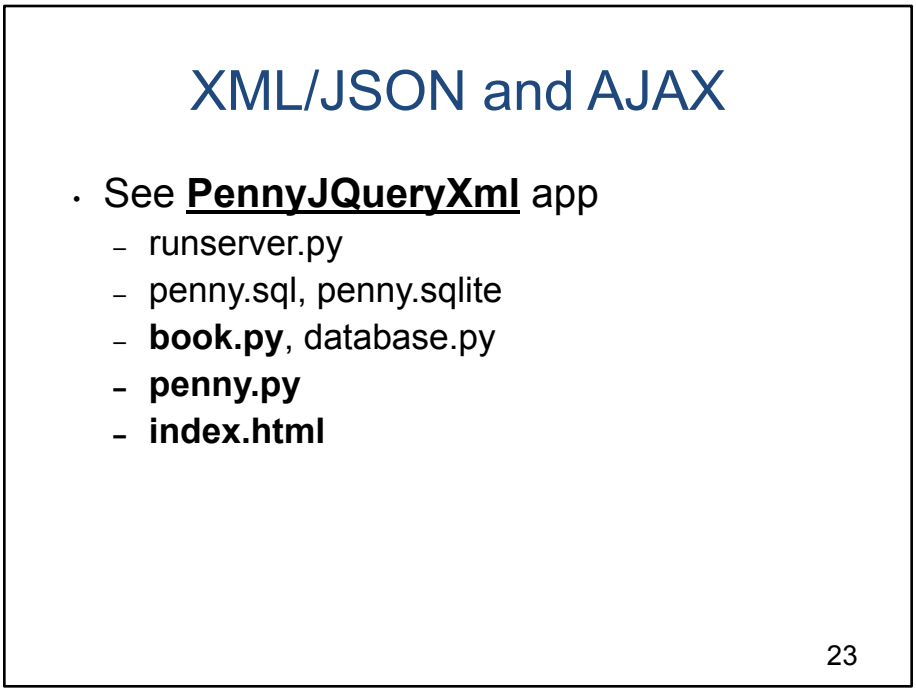
index.html

Uses jQuery `.ajax()` function

Which uses `XMLHttpRequest` object

XML/JSON and AJAX

- Observation
  - Server need not send response as HTML
  - Server could send response as XML or JSON



XML/JSON and AJAX

[see slide]

Code notes:

book.py

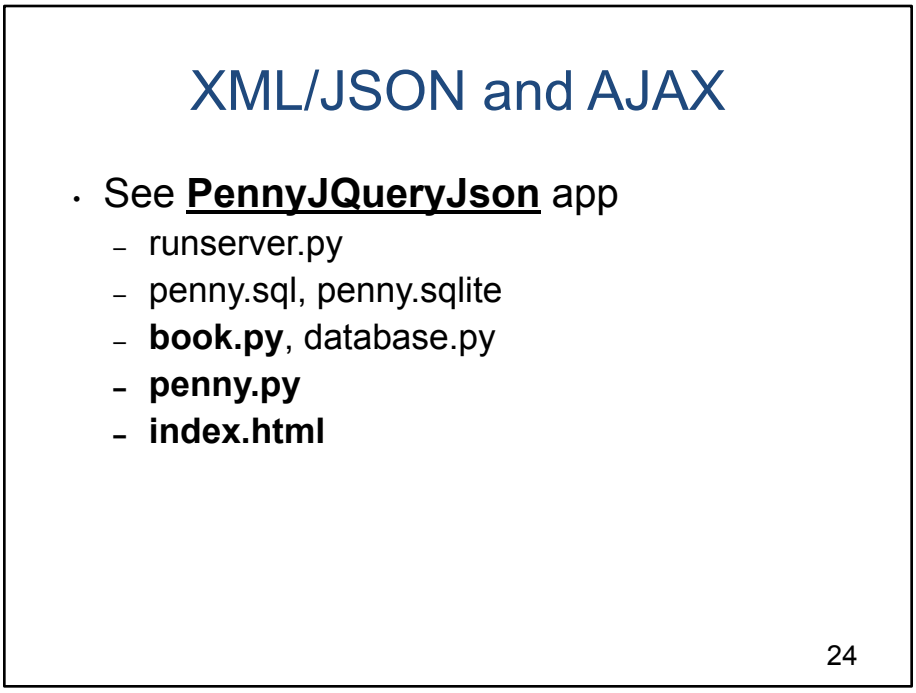
Defines to_xml() method
Returns XML representation of self

penny.py

Calls book.to_xml()
Sends response as XML doc
Sets content type to application/xml

index.html

Uses jQuery `.ajax()` function
Which uses `XMLHttpRequest` object
`XMLHttpRequest` object parses XML doc to generate JavaScript data structure
App traverses the JavaScript data structure
App creates DOM nodes containing text
App inserts the DOM nodes into DOM



XML/JSON and AJAX

[see slide]

Code notes:

book.py

Defines to_dict() method
Returns dict representation of self

penny.py

Calls book.to_dict() method
Sends response as JSON doc
Sets content-type to application/json

search.html

Uses jQuery `.ajax()` function
Which uses `XMLHttpRequest` object
jQuery (not AJAX) parses JSON response to generate JavaScript data structure
App traverses the JavaScript data structure
App creates DOM nodes containing text
App inserts the DOM nodes into DOM

XML/JSON and AJAX

- **Question**
  - Why should server send XML/JSON instead of HTML?

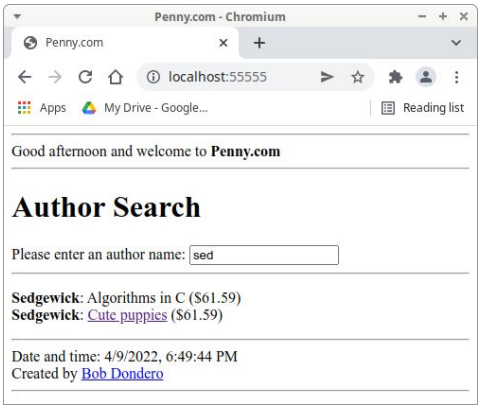
XML/JSON and AJAX

- **Answer 1:** Sending XML or JSON mitigates XSS attacks
  - Example:

```
$ sqlite3 penny.sqlite
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
sqlite> select * from books;
Kernighan|The Practice of Programming|40.74
Kernighan|The C Programming Language|24.99
Sedgewick|Algorithms in C|61.59
Sedgewick|<a
href="https://www.akc.org/expert-advice/lifestyle/cute-pu
ppies/">Cute puppies</a>|61.59
sqlite> .quit
$
```

XML/JSON and AJAX

- Recall **PennyJQuery** app



XML/JSON and AJAX

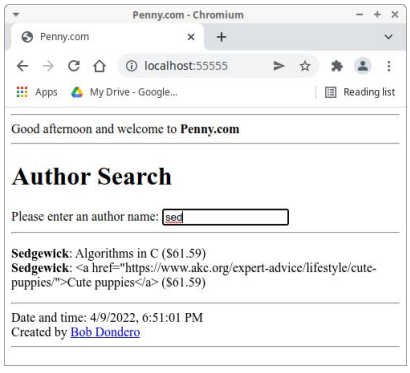
[see slide]

Server sends HTML

Browser inserts HTML into DOM

XML/JSON and AJAX

- Recall **PennyJQueryXml** and **PennyJQueryJson** apps



28

XML/JSON and AJAX

Focusing in PennyJQueryJson app…

Server sends JSON

Browser:

- Parses JSON to generate a JavaScript data structure
- Traverses the JavaScript data structure
- Creates DOM nodes containing text
- Inserts the DOM nodes into DOM

It would be harder to generate the JavaScript data structure if server sends HTML

Also could, in JavaScript, escape given HTML before inserting it into the DOM

- But then the strings and would appear in the rendered page

XML/JSON and AJAX

- **Answer 2:** Sending XML or JSON defines a convenient API
  - HTML: presentation info
    - Makes sense to send HTML if client is **browser**
  - XML/JSON: no presentation info
    - Makes sense to send XML/JSON if client is **???**
  - Example…

XML/JSON and AJAX

- See **booksbyauthorjson.py**
  - The job
    - Using PennyJQueryJson app...
    - Fetch books with given author name prefix

```
$ python booksbyauthorjson.py localhost 44445 ker
Kernighan
The Practice of Programming
40.74

Kernighan
The C Programming Language
24.99

$
```

30

XML/JSON and AJAX

Server sends JSON

booksbyauthorjson.py
Parses JSON to generate a Python data structure
Traverses the Python data structure
Prints the book data

It would be harder to generate Python data structure if the server sends HTML

Summary

- We have covered:
  - JSON
  - JSON programming
  - XML/JSON and AJAX

Summary

- We have covered:
  - XML
  - XML programming: DOM
  - XML programming: SAX
  - JSON
  - JSON programming
  - XML/JSON and AJAX
- See also:
  - **Appendix**: JSON Checking

Appendix:
JSON Checking

JSON Checking

- To run the example Python programs in this appendix:
  - `python –m pip install jsonschema`

JSON Checking: Well-Formedness

- Computer science jargon…
  - A JSON document is ***well-formed*** iff it conforms to the JSON specification

JSON Checking: Well-Formedness

- See books.json (revisited)
- See **booksmalformed.json**
- See **checkjson.py**

```
$ python checkjson.py books.json
The document is well-formed.
$ python checkjson.py booksmalformed.json
Expecting ',' delimiter: line 15 column 7 (char 353)
$
```

JSON Checking: Validity

- ***JSON Schemas***
  - Internet draft
    - Published by Internet Engineering Task Force
    - 12th draft released 2021 February 01
    - See http://json-schema.org/
    - See Wikipedia JSON page
  - A JSON schema defines whether a JSON doc is ***valid***

JSON Checking: Validity

- See **books.schema.json**
- See books.json (revisited)
- See **booksinvalid.json**
- See **checkjsonusingschema.py**

JSON Checking: Validity

```
$ python checkjsonusingschema.py books.json books.schema.json
The document is well-formed.
The document is valid.
$ python checkjsonusingschema.py booksinvalid.json books.schema.json
The document is well-formed.
'currency' is a required property

Failed validating 'required' in schema['items']:
    {'properties': {'author': {'type': 'string'},
                    'currency': {'type': 'string'},
                    'price': {'type': 'number'},
                    'title': {'type': 'string'}},
     'required': ['author', 'title', 'price', 'currency'],
     'type': 'object'}

On instance[2]:
    {'author': 'Sedgewick', 'price': 61.59, 'title': 'Algorithms in
C'}
$
```