JSON and the Rest Architecture

JavaScript Object Notation (JSON)

Douglas Crockford Invented JSON

JSON is Object literal notation in Javascript

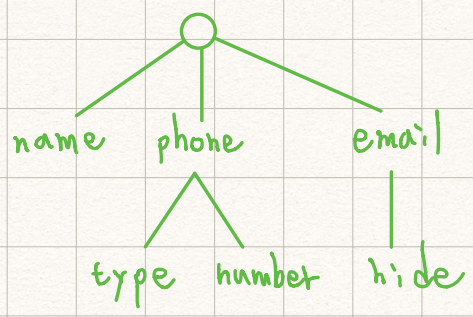
①

```
import json
info = json.loads(data)
print('Name:', info['name'])
print('Hide:', info['email']['hide'])

data = ...{
    'name': 'chuck',
    'phone': {
        'type': 'intl',
        'number': '1234'
    },
    'email': {
        'hide': 'yes'
    }
}...
```

info type is dictionary

JSON like nested list or dictionary in Python language



②

```
import json
info = json.loads(data)
print('User count:', len(info))
for item in info:
    print('Name:', info['name'])
    print('Hide:', info['id'])

data = '''[
 {"id":"001",
   "x":"2",
   "name":"chuck"},

 {"id":"009",
   "x":"7",
   "name":"chuck"}
]'''
```

info type is list

JSON like nested list or dictionary in Python language

JSON is simpler than XML

Interview: Douglas Crockford - Discovering JSON

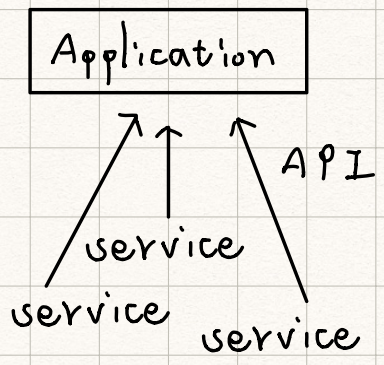
AJAX ↗ JSON (not Standard until json.org)

↕

XML JSON Faster and Easier

Service Oriented Approach

Most non-trival web applications use services



They use service from other applications

- Credit Card Charge
- Hotel Reservation systems

Services publish the "rules" applications must follow to make use of the service (API)

Multiple Systems

Initially - two systems cooperate and split the problem

As the data/service becomes useful - multiple applications want to use the information / application

Video: Service Oriented Architectures

a service layer between systems
even, between organizations nationally
and globally

service layer to this data enables users
to share, re-use, and re-purpose

Using Application Programming Interfaces

The Google Geocoding API (Google Maps API)

URL=maps.googleapis.com/maps/api/geocode/json?
API=address = Ann+Arbor%2C+MI

geojson.py

```
address = input('Enter location: ')

url = URL + urllib.parse.urlencode
({'address': address})
urllib.request.urlopen(url)
```

Securing API Requests

API Security and Rate Limiting

The compute resources to run these APIs are not "free"

The data provided by these APIs is usually valuable

The data provides might limit the number of requests per day, demand an API "key", or even charge for usage

They might change the rules as things progress ...

Twitter

twitter have different types of APIs

```
url = twurl.argument(TWITTER_URL,
{`screen_name`: acct, `count`: `5`})
```

```
connection = urllib.request.urlopen(url)
headers = dict(connection.getheaders())
print('Remaining', headers['x-rate-limit-remaining'])
```

Twitter oauth

```
import oauth

oauth.OAuthConsumer(  ......  )
oauth.OAuthToken(  ......  )
```