大数据基础与应用
BIG DATA FOUNDATION AND APPLICATION

JSON和XML数据交换

北京理工大学计算机学院　高玉金

2019年3月

JSON格式

- JavaScript Object Notation (JSON)是源于JavaScript的当今很流行的数据交换格式，它是JavaScript语言的一个子集，也是Python合法可支持的语法
- 简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率
- Json采用完全独立于编程语言的文本格式来存储和表示数据
  - 数据在名称/值对中
  - 数据由逗号分隔
  - 花括号保存对象
  - 方括号保存数组

将Python数据结构转换为JSON

- 可以将字典转化成JSON来发送，一般使用JSON库的dumps()方法

```
data = {'name':'nick','age':12}
data_json = json.dumps(data)
```

- data_json的类型为str（注意引号）：'{"name": "Nick", "age": 20}'
- data_json为转换后的JavaScript对象，其key值默认为字符串格式
- data_json中的key的顺序与data字典中key的顺序不保证相同
- data还可以是str，list，tuple，int或对象等其他类型

```
In [34]: json.dumps(tuple("hello"))
Out[34]: '["h", "e", "l", "l", "o"]'
```

将JSON转换为Python内置数据结构

- 将JSON转化为Python的字典时，可以使用JSON库的loads()方法
- data = json.loads(data_json)
- data中的字符串默认均为unicode，JSON规定的编码为UTF-8
- 对于文件的JSON处理，应使用dump()和load()

```
# Writing JSON data
with open('data.json', 'w') as f:
    json.dump({"name":"D","age":20}, f)

# Reading data back
with open('data.json', 'r') as f:
    data = json.load(f)
```

```
data
{'name': 'D', 'age': 20}
```

Python 与 JSON 类型转换

Python	JSON
dict	object
list, tuple	array
str	string
int, float, int- & float-derived Enums	number
True	true
False	false
None	null

JSON	Python
object	dict
array	list
string	str
number (int)	int
number (real)	float
true	True
false	False
null	None

XML文件格式

- XML 指可扩展标记语言（e**X**tensible **M**arkup **L**anguage）
- XML 被设计用来传输和存储数据
- XML 是一套定义语义标记的规则，这些标记将文档分成许多部件并对这些部件加以标识
- 它也是元标记语言，即定义了用于定义其他与特定领域有关的、语义的、结构化的标记语言的句法语言

```
<?xml   version="1.0"   encoding="utf-8"?>
  <userdata   createuser="false">userdata内容
      <dataconnection>
          <server>localhost</server>
          <uid>sa</uid>
          <pwd></pwd>
      </dataconnection>
      <net>
          <name>jiayuan</name>
      </net>
  </userdata>
```

Python解析XML的方法

- SAX (simple API for XML)

Python 标准库包含 SAX 解析器，SAX 用事件驱动模型，通过在解析XML的过程中触发一个个的事件并调用用户定义的回调函数来处理XML文件

- DOM(Document Object Model)

将 XML 数据在内存中解析成一个树，通过对树的操作来操作XML

- ElementTree(元素树)

ElementTree就像一个轻量级的DOM，具有方便友好的API。代码可用性好，速度快，消耗内存少。

优先使用ET解析

import xml.etree.ElementTree

- ET提供了两个对象：ElementTree将整个XML文档转化为树，Element则代表着树上的单个节点。对整个XML文档的交互（读取，写入，查找需要的元素），一般是在ElementTree层面进行的。对单个XML元素及其子元素，则是在Element层面进行的。
  - ✓将XML文档解析为树（tree）tree = ET.ElementTree(file='doc1.xml')
  - ✓查找需要的元素for elem in tree.iter(tag='branch'):
  - ✓支持通过XPath查找元素
- 构建XML文档比较复杂，目前XML的应用场景逐渐被JSON取代，建议仅作了解