

柳帅

搏一搏

博 首 新 联 订 管

随笔 - 178 文章 - 1 评论 - 7

客 页 随 系 阅 理

园 笔

打赏

公告



昵称: 柳帅
园龄: 10个月
粉丝: 14
关注: 30
+加关注

随笔分类

app 数据爬取
(7)
django(12)
docker(1)

jsonpath 使用教程(快速处理dict的深度查询)

一 简介

- **JSONPath - 用于JSON的XPath**
- 用来解析多层嵌套的json数据;JsonPath 是一种信息抽取类库, 是从JSON文档中抽取指定信息的工具.

二 安装

安装方法: `pip install jsonpath`

官方文档: <http://goessner.net/articles/JsonPath>

三 强大之处

```
import jsonpath
res=jsonpath.jsonpath(dic_name,'$..key_name')
```

嵌套n层也能取到所有key_name信息,其中: "\$"表示 {}, "\$.."表示模糊匹配,当传入不存在的key_name时,程序会返回 false.



js解密之路(1)

Linux(16)

luffy项目实战(3)

mongodb(2)

mysql 数据库学习(7)

py3+cv2(21)

pyspider(3)

python(40)

python 数据分析(24)

python 算法学习(5)

python 拓展(3)

Redis(2)

R语言 学习

scrapy(3)

Vue学习(3)

web学习(12)

面试知识点(4)

爬虫实战(2)

随笔档案

2019年9月(1)

2019年7月(21)

2019年6月(1)

2019年5月(12)

2019年4月(16)

2019年3月(27)

3

...

四 jsonpath表达式

- 使用点注释

```
$.store.book[0].title
```

- 括号 -注释

```
$['store']['book'][0]['title']
```

五 jsonpath与xpath的联系

JSONPath语法元素与XPath对应的完整概述和并排比较。

Xpath	JSONPath	描述
/	\$	跟节点
.	@	现行节点
/	. or []	取子节点
..	n/a	就是不管位置，选择所有符合条件的
*	*	匹配所有元素节点
[]	[]	迭代器标示(可以在里面做简单的迭代
|	[,]	支持迭代器中做多选
[]	?()	支持过滤操作
n/a	()	支持表达式计算
()	n/a	分组，JsonPath不支持

六 实例演示

```
{ "store": {  
    "book": [  
        { "category": "reference",  
          "author": "Nigel Rees",  
          "title": "Sayings of the Century",  
          "price": 8.95  
        },  
        { "category": "fiction",
```



打赏



2018年12月
(28)
2018年11月
(31)
2018年10月
(3)

阅读排行榜

- 1. jsonpath...
- 2. 学习强国...
- 3. 爬虫系列 ...
- 4. 抖音短视...
- 5. Iframe框...

推荐排行榜

- 1. jsonpath 使用教程(快速处理dict的深度查询)(3)
- 2. 抖音短视频爬取实战(3)
- 3. k邻近算法(KNN)实例(2)
- 4. 第二节 pandas 基础知识(1)
- 5. 第一节 anaconda+jupyter+numpy简单使用(1)

```
    "author": "Evelyn Waugh",
    "title": "Sword of Honour",
    "price": 12.99
  },
  {
    "category": "fiction",
    "author": "Herman Melville",
    "title": "Moby Dick",
    "isbn": "0-553-21311-3",
    "price": 8.99
  },
  {
    "category": "fiction",
    "author": "J. R. R. Tolkien",
    "title": "The Lord of the Rings",
    "isbn": "0-395-19395-8",
    "price": 22.99
  }
],
"bicycle": {
  "color": "red",
  "price": 19.95
}
}
```

XPath的	JSONPath	结果
/store/book/author	\$.store.book[*].author	商店里所有书籍的作者
//author	\$..author	所有作者
/store/*	\$.store.*	商店里的所有东西，都是一些书和一辆红色的自行车。
/store//price	\$.store..price	商店里一切的价格。
//book[3]	\$..book[2]	第三本书
//book[last()]	<div>\$.book[(@.length-1)]</div> <div>\$..book[-1:]</div>	最后一本书。
//book[position()<3]	<div>\$..book[0,1]</div> <div>\$..book[:2]</div>	前两本书
//book[isbn]	<div>\$..book[?(@.isbn)]</div>	使用isbn number过滤
//book[price<10]	<div>\$..book[?(@.price<10)]</div>	过滤所有便宜10以上的
//*	\$..*	XML文档中的所有元结构的所有成员。



python使用示例

使用格式:

```
jsonpath.jsonpath(匹配的字典, 'jsonpath表达式')
```

如: `res2= jsonpath.jsonpath(d, '$..name')` #找d字典下面所有的name对应的值, 返回一个列表

```
shop={
    "store": {
        "book": [
            {
                "category": "reference",
                "author": "Nigel Rees",
                "title": "Sayings of the Century",
                "price": 8.95
            },
            {
                "category": "fiction",
                "author": "Evelyn Waugh",
                "title": "Sword of Honour",
                "price": 12.99
            },
            {
                "category": "fiction",
                "author": "Herman Melville",
                "title": "Moby Dick",
                "isbn": "0-553-21311-3",
                "price": 8.99
            },
            {
                "category": "fiction",
                "author": "J. R. R. Tolkien",
                "title": "The Lord of the Rings",
                "isbn": "0-395-19395-8",
                "price": 22.99
            }
        ],
        "bicycle": {
            "color": "red",
            "price": 19.95
        }
    },
    "expensive": 10
}
```

```
import jsonpath
```



打赏

#商店里所有书籍的作者

```
author_list=jsonpath.jsonpath(shop,'$.store.book[*].author')
print(author_list)  #['Nigel Rees', 'Evelyn Waugh', 'Herman Melville', 'J. R. R. Tolkien']
```

#返回所有的作者

```
author_list2=jsonpath.jsonpath(shop,'$..author')
print(author_list2)  #['Nigel Rees', 'Evelyn Waugh', 'Herman Melville', 'J
```

#商店里的所有东西

```
category_dx=jsonpath.jsonpath(shop,'$.store.*')
print(category_dx)
```

#商店里一切的价格

```
store_price_list=jsonpath.jsonpath(shop,'$.store..price')
print(store_price_list)  #[8.95, 12.99, 8.99, 22.99, 19.95]
```

#第三本书

```
book_3=jsonpath.jsonpath(shop,'$..book[2]')
print(book_3)
```

#最后一本书

```
num=len(jsonpath.jsonpath(shop,'$..book'))-1
book_last=jsonpath.jsonpath(shop,f'$..book[{num}]')
print(book_last)
```

#前两本书

```
book_12=jsonpath.jsonpath(shop,f'$..book[0,1]')
print(book_12)
```

#过滤所有便宜10以上的书籍

```
book_lg10=jsonpath.jsonpath(shop,'$..book[?(@.price<10)]')
print(book_lg10)
```

#使用isbn number过滤所有书籍

```
book_lg10=jsonpath.jsonpath(shop,'$..book[?(@.isbn)]')
print(book_lg10)
```

分类: [python 数据分析](#) , [python 算法学习](#)

好文要顶

关注我

收藏该文



柳帅

关注 - 30

粉丝 - 14