

测试面试题

作者：上海-悠悠

关注微信公众号：yoyoketang



个人博客地址：<http://www.cnblogs.com/yoyoketang/>

QQ 交流群：779429633

联系本人 qq：283340479

交流 qq 群：779429633

目录

一、SQL 篇	6
SQL 学生表-1	6
前言	6
万年不变学生表	6
排序 order by	7
统计总成绩 sum	8
统计总成绩	9
统计单科最好成绩	9
总结 group by 相关用法	10
各门课程成绩最好的 2 位学生	11
SQL 学生表-2	11
前言	11
万年不变学生表	12
计算学生平均分数	12
统计各科目成绩	13
每门课程平均成绩	15
成绩排名	15
同结果名次相同	17
排名相同的占个名次	18
SQL 查询-3	19
前言	19
万年不变学生表	19
查询前 3 名	20
查询第 2-3 名记录	21
查询第 3 到后面所有的	22
英语课程少于 80 分的人	23
统计每门课程不及格、一般、优秀	23
查找每科成绩前 2 名	24
SQL 经典面试题-4	26
前言	26
having 和 not in	26
解决办法一： having	26
解决办法二： not in	27
删除	27
模糊查询%	29
SQL 通配符	29
二、python 篇	30
python 笔试题-1	30
前言	30
1.统计	31
2.字符串切片	31
3.字符串切割	32

4.格式化输出.....	32
5.队列.....	32
6.交换.....	32
7.水仙花.....	33
8.完全数.....	33
9.排序.....	34
10.sort 排序	34
python 笔试题(递归)-2	35
计算 n 的阶乘.....	35
斐波那契数列.....	37
幂的递归.....	37
汉诺塔问题.....	38
python 编程题-3.....	39
python 编程 4-遍历文件	41
三、linux 篇	42
linux 常用指令-1.....	42
前言.....	42
常用指令.....	42
ps 查看进程	43
kill 杀掉进程	43
启动服务.....	43
查看日志.....	44
查看端口.....	44
find 查找文件	45
测试环境搭建-tomcat	46
测试环境如何搭建?	46
以 jenkins 搭建为例 (web 网站搭建都是一个套路)	47
前言.....	47
wget.....	47
下载 jenkins	47
启动 jenkins	48
四、接口篇.....	51
http 协议相关面试题.....	52
前言.....	52
浏览器输入 url 按回车背后经历了哪些?	52
GET 和 POST 的区别	52
cookies 机制和 session 机制的区别	53
HTTP 状态码	54
http 协议请求方式.....	54
http 和 https 区别.....	54
报文.....	55
post 请求 body	55
DNS.....	56
无状态.....	56

接口测试面试题.....	56
前言.....	56
为什么要做接口测试?	56
接口测试能发现哪些问题.....	57
接口测试怎么测.....	58
用什么工具测.....	60
WebService 接口	61
没有接口文档如何做接口测试.....	62
数据依赖.....	62
依赖第三方.....	62
抓包.....	63
弱网.....	63
分析 bug 是前端还是后端的.....	63
接口自动化面试题.....	63
前言.....	63
1.json 和字典 dict 的区别?	64
2.测试的数据你放在哪?.....	65
3.什么是数据驱动, 如何参数化?	66
4.下个接口请求参数依赖上个接口的返回数据	68
5.依赖于登录的接口如何处理.....	69
6.依赖第三方的接口如何处理.....	69
7.不可逆的操作, 如何处理, 比如删除一个订单这种接口如何测试	69
8.接口产生的垃圾数据如何清理.....	69
9.一个订单的几种状态如何全部测到?	70
10.python 如何连接数据库操作?	70
Mock 相关-单元测试	71
什么是 mock.....	71
mock 能做什么	71
mock 环境准备.....	72
依赖关系.....	72
mock-server 环境搭建—加分项!	74
前言.....	74
环境准备.....	74
Moco 环境搭建	74
遇到的问题.....	76
官方文档.....	77
五、selenium 篇.....	78
selenium 中隐藏元素如何定位?	78
前言.....	78
隐藏元素.....	78
定位隐藏元素.....	80
操作隐藏元素.....	80
JS 操作隐藏元素.....	81
selenium 面试题.....	82

前言.....	82
1.如何判断一个页面上元素是否存在?	82
2.如何提高脚本的稳定性.....	83
3.如何定位动态元素.....	84
4.如何通过子元素定位父元素.....	84
5.如果截取某一个元素的图片，不要截取全部图片.....	86
6.平常遇到过哪些问题?如何解决的.....	86
7.一个元素明明定位到了，点击无效（也没报错），如果解决?	86
六、app 篇.....	87
app 测试面试题.....	87
前言.....	87
1.什么是 activity.....	87
2.Activity 生命周期?.....	88
3.Android 四大组件.....	90
4.app 测试和 web 测试有什么区别?	90
5.android 和 ios 测试区别?	91
6.app 出现 ANR，是什么原因导致的?	92
7.App 出现 crash 原因有哪些?	93
8.app 对于不稳定偶然出现 anr 和 crash 时候你是怎么处理的?	93
9.app 的日志如何抓取?.....	94
10.你平常会看日志吗，一般会出现哪些异常（Exception）?	94
七、jenkins 持续集成-加分项.....	95
如何运行代码?	96
如何新增 slave 节点.....	97
如何设置定时任务?	104
如何结合 git 拉取代码?	107
2019 课程推荐	108
好书推荐.....	109
Selenium+python.....	109
Python 接口自动化.....	110
python 自动化框架 pytest.....	111
Appium+python	111

一、SQL 篇

Sql 是必考的基础题，sql 至少需要掌握简单的增删改查，2 表联合查询，排序等常规操作

SQL 学生表-1

常见 SQL 面试题：

1. 查询所有学生的数学成绩，显示学生姓名 name，分数，由高到低
2. 统计每个学生的总成绩, 显示字段：姓名，总成绩
3. 统计每个学生的总成绩（由于学生可能有重复名字），显示字段：学生 id，姓名，总成绩
4. 列出各门课程成绩最好的学生，要求显示字段：学号，姓名, 科目，成绩

第一步先 group by 找出单科最好成绩，作为第一张表

5. 列出各门课程成绩最好的 2 位学生，要求显示字段：学号，姓名，科目，成绩

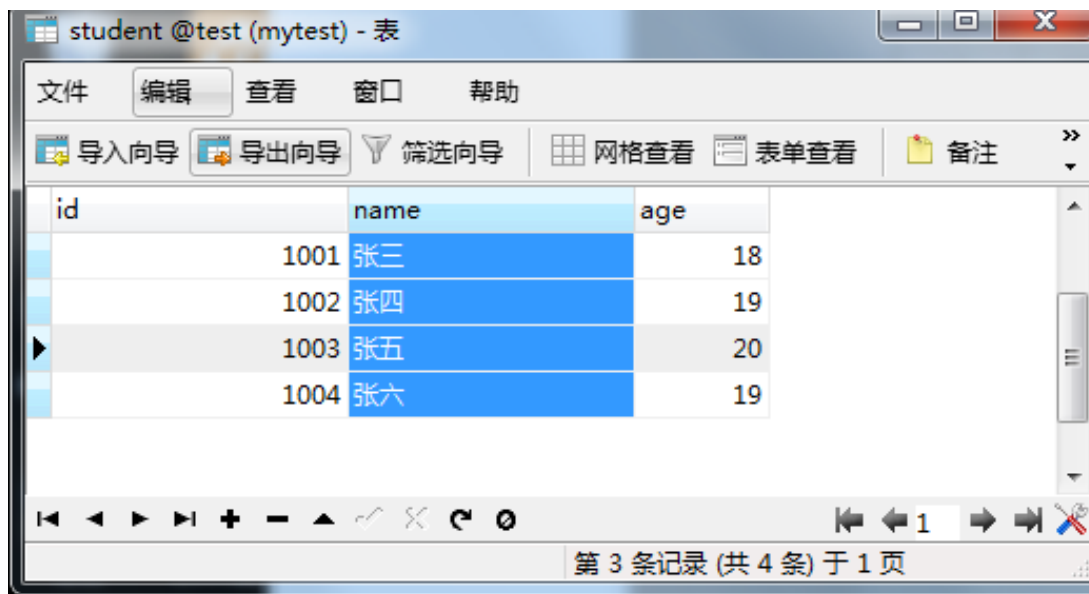
前言

每次面试必考 SQL, 小编这几年一直吃 SQ 的亏，考题无非就是万年不变学生表，看起来虽然简单，真正写出来，还是有一定难度。于是决定重新整理下关于 SQL 的面试题，也可以帮助更多的人过 SQL 这一关。

作为一个工作 3 年以上测试人员，不会 sql 基本上能拿到 offer 的希望渺茫，虽然平常也会用到数据库，都是用的简单的查询语句。困难一点的就直接找开发了，面试想留个好印象，还是得熟练掌握，能在纸上快速写出来。

万年不变学生表

有 2 张表，学生表（student）基本信息如下



id	name	age
1001	张三	18
1002	张四	19
1003	张五	20
1004	张六	19

第 3 条记录 (共 4 条) 于 1 页

科目和分数表 (grade)



no	id	kemu	score
1	1001	语文	85
2	1001	数学	86
3	1002	英语	98
4	1002	语文	94
5	1002	数学	98
6	1003	数学	56
7	1003	语文	69
8	1003	英语	68
9	1004	英语	99
10	1004	数学	97

排序 order by

1. 查询所有学生的数学成绩，显示学生姓名 name，分数， 由高到低

```
SELECT a.name, b.score
FROM student a, grade b
WHERE a.id = b.id
AND kemu = '数学'
```

ORDER BY score
DESC

查询创建工具 查询编辑器	
<pre>1 SELECT a.name, b.score 2 FROM student a, grade b 3 WHERE a.id = b.id 4 AND kemu = '数学' 5 ORDER BY score 6 DESC</pre>	
信息 结果1 概况 状态	
name	score
张四	98
张六	97
▶ 张三	86
张五	56

统计总成绩 sum

2. 统计每个学生的总成绩, 显示字段: 姓名, 总成绩

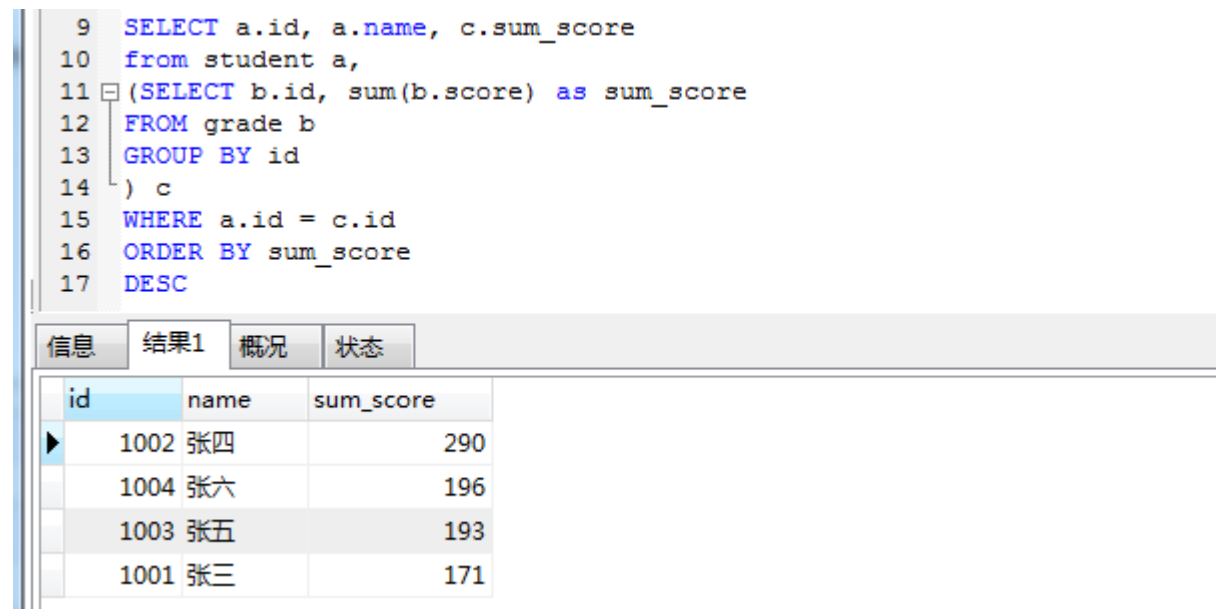
```
SELECT a.name, sum(b.score) as sum_score
FROM student a, grade b
WHERE a.id = b.id
GROUP BY name
DESC
```

查询创建工具 查询编辑器	
<pre>1 SELECT a.name, sum(b.score) as sum_score 2 FROM student a, grade b 3 WHERE a.id = b.id 4 GROUP BY name</pre>	
信息 结果1 概况 状态	
name	sum_score
▶ 张三	171
张四	290
张五	193
张六	196

统计总成绩

3. 统计每个学生的总成绩（由于学生可能有重复名字），显示字段：学生 id，姓名，总成绩

```
SELECT a.id, a.name, c.sum_score
from student a,
(SELECT b.id, sum(b.score) as sum_score
FROM grade b
GROUP BY id
) c
WHERE a.id = c.id
ORDER BY sum_score
DESC
```



```
9 SELECT a.id, a.name, c.sum_score
10 from student a,
11 (SELECT b.id, sum(b.score) as sum_score
12 FROM grade b
13 GROUP BY id
14 ) c
15 WHERE a.id = c.id
16 ORDER BY sum_score
17 DESC
```

信息	结果1	概况	状态
id	name	sum_score	
1002	张四	290	
1004	张六	196	
1003	张五	193	
1001	张三	171	

统计单科最好成绩

4. 列出各门课程成绩最好的学生， 要求显示字段：学号，姓名，科目，成绩

第一步先 group by 找出单科最好成绩，作为第一张表

```
SELECT b.kemu, MAX(b.score)
FROM grade b
GROUP BY kemu
```

```
18
19 -- 单科最好的成绩
20
21 SELECT b.kemu, MAX(b.score)
22 FROM grade b
23 GROUP BY kemu
24
25
```

信息	结果1	概况	状态
kemu	MAX(b.score)		
▶ 语文	94		
数学	98		
英语	99		

再结合学生表和分数表，得到单科最好成绩

-- 单科最好的成绩

```
SELECT c.id , a.name, c.kemu, c.score
```

```
FROM grade c, student a,
(SELECT b.kemu, MAX(b.score) as max_score
FROM grade b
GROUP BY kemu) t
WHERE c.kemu = t.kemu
AND c.score = t.max_score
AND a.id = c.id
```

```
19
20 -- 单科最好的成绩
21 SELECT c.id , a.name, c.kemu, c.score
22
23 FROM grade c, student a,
24 (SELECT b.kemu, MAX(b.score) as max_score
25 FROM grade b
26 GROUP BY kemu) t
27 WHERE c.kemu = t.kemu
28 AND c.score = t.max_score
29 AND a.id = c.id
30
31
```

信息	结果1	概况	状态
id	name	kemu	score
▶ 1002	张四	语文	94
1002	张四	数学	98
1004	张六	英语	99

总结 group by 相关用法

函数	作用	支持性
sum(列名)	求和	
max(列名)	最大值	
min(列名)	最小值	
avg(列名)	平均值	
first(列名)	第一条记录	仅 Access 支持
last(列名)	最后一条记录	仅 Access 支持
count(列名)	统计记录数	注意和 count(*) 的区别

各门课程成绩最好的 2 位学生

5. 列出各门课程成绩最好的 2 位学生， 要求显示字段：学号，姓名，科目，成绩

```
SELECT t1.id, a.name, t1.kemu, t1.score
FROM grade t1, student a
WHERE
    (SELECT count(*) FROM grade t2
     WHERE t1.kemu=t2.kemu AND t2.score>t1.score
    )<2
and a.id = t1.id
ORDER BY t1.kemu, t1.score
DESC
```

SQL 学生表-2

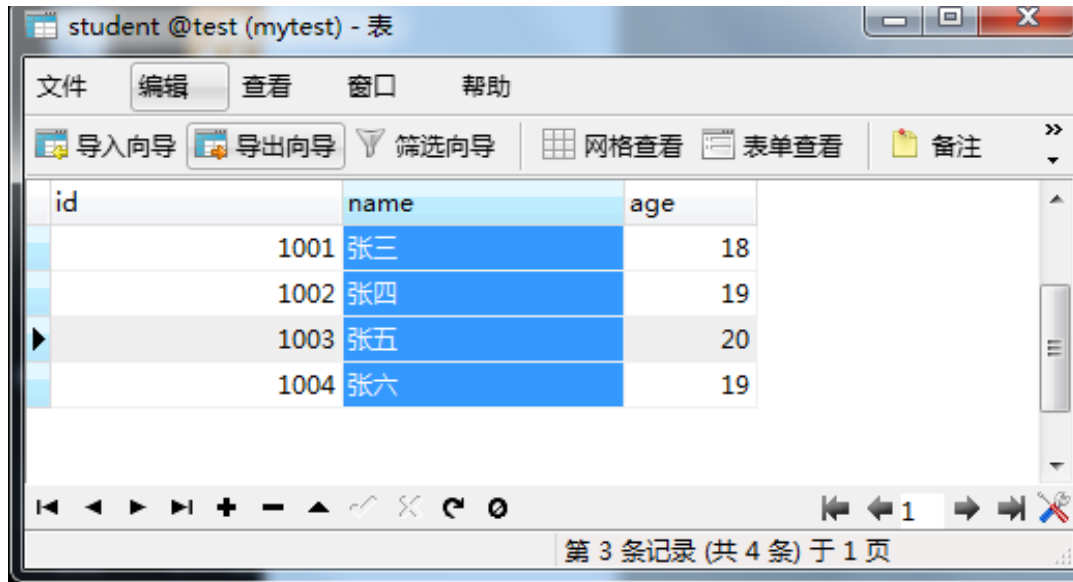
前言

接着上一篇，继续学生表 SQL

1. 计算每个人的平均成绩， 要求显示字段：学号，姓名，平均成绩
2. 计算每个人的成绩，总分数，平均分，要求显示：学号，姓名，语文，数学，英语，总分，平均分
3. 列出各门课程的平均成绩，要求显示字段：课程，平均成绩
4. 列出数学成绩的排名， 要求显示字段：学号，姓名，成绩，排名

万年不变学生表

有 2 张表，学生表（student）基本信息如下



id	name	age
1001	张三	18
1002	张四	19
1003	张五	20
1004	张六	19

第 3 条记录 (共 4 条) 于 1 页

科目和分数表（grade）



no	id	kemu	score
1	1001	语文	85
2	1001	数学	86
3	1002	英语	98
4	1002	语文	94
5	1002	数学	98
6	1003	数学	56
7	1003	语文	69
8	1003	英语	68
9	1004	英语	99
10	1004	数学	97

第 3 条记录 (共 4 条) 于 1 页

计算学生平均分数

1. 计算每个人的平均成绩， 要求显示字段：学号，姓名，平均成绩

```
select a.id, a.name, c.avg_score
from student a,
(select b.id, avg(b.score) as avg_score
from grade b
group by b.id
)c
where a.id = c.id
```

查询创建工具

查询编辑器

```

1  select  a.id, a.name, c.avg_score
2  from student a,
3  (select b.id, avg(b.score) as avg_score
4  from grade b
5  group by b.id
6  )c
7  where a.id = c.id

```

信息

结果1

概况

状态

id	name	avg_score
1001	张三	78.66666666666667
1002	张四	96.66666666666667
1003	张五	64.33333333333333
1004	张六	88

统计各科目成绩

2. 计算每个人的成绩，总分数，平均分，要求显示：学号，姓名，语文，数学，英语，总分，平均分

使用 case when 语法把科目字段分解成具体的科目：语文，数学， 英语

```
select a.id as 学号, a.name as 姓名,
(case when b.kemu='语文' then score else 0 end) as 语文,
(case when b.kemu='数学' then score else 0 end) as 数学,
(case when b.kemu='英语' then score else 0 end) as 英语
from student a, grade b
where a.id = b.id
```

```

20 select a.id as 学号, a.name as 姓名,
21 (case when b.kemu='语文' then score else 0 end) as 语文,
22 (case when b.kemu='数学' then score else 0 end) as 数学,
23 (case when b.kemu='英语' then score else 0 end) as 英语
24 from student a, grade b
25 where a.id = b.id
26

```

学号	姓名	语文	数学	英语
1001	张三	85	0	0
1001	张三	0	86	0
1002	张四	0	0	98
1002	张四	94	0	0
1002	张四	0	98	0
1003	张五	0	56	0
1003	张五	69	0	0
1003	张五	0	0	68
1004	张六	0	0	99
1004	张六	0	97	0
1001	张三	0	0	65
1004	张六	68	0	0

```

SELECT a.id as 学号, a.name as 姓名,
sum(case when b.kemu='语文' then score else 0 end) as 语文,
sum(case when b.kemu='数学' then score else 0 end) as 数学,
sum(case when b.kemu='英语' then score else 0 end) as 英语,
sum(b.score) as 总分 ,
sum(b.score)/count(b.score) as 平均分
FROM student a, grade b
where a.id = b.id
GROUP BY b.id, b.id

```

学号	姓名	语文	数学	英语
1001	张三	85	0	0
1001	张三	0	86	0
1002	张四	0	0	98
1002	张四	94	0	0
1002	张四	0	98	0
1003	张五	0	56	0
1003	张五	69	0	0
1003	张五	0	0	68
1004	张六	0	0	99
1004	张六	0	97	0
1001	张三	0	0	65
1004	张六	68	0	0


```

1 select a.id, a.name, c.avg_score
2 from student a,
3 (select b.id, avg(b.score) as avg_score
4 from grade b
5 group by b.id
6 ) c
7 where a.id = c.id

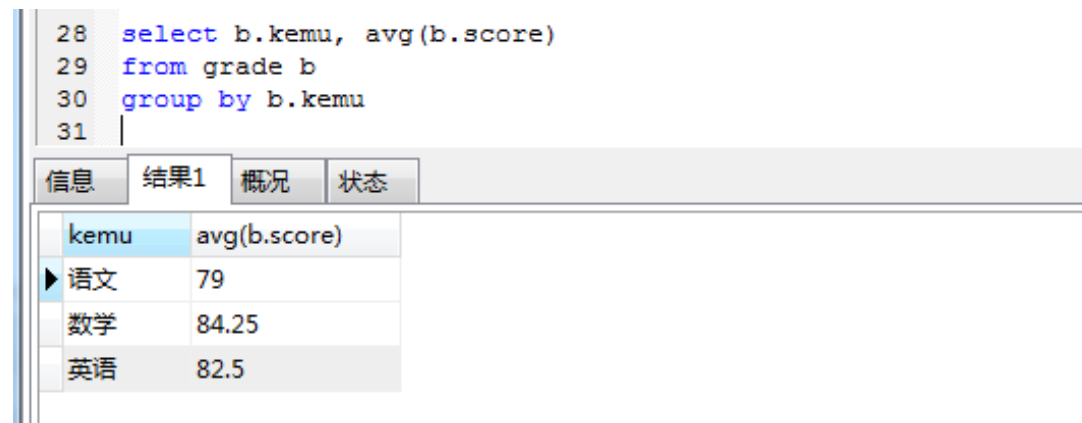
```

id	name	avg_score
1001	张三	78.66666666666667
1002	张四	96.66666666666667
1003	张五	64.33333333333333
1004	张六	88

每门课程平均成绩

3. 列出各门课程的平均成绩，要求显示字段：课程，平均成绩

```
select b.kemu, avg(b.score)
from grade b
group by b.kemu
```



The screenshot shows a SQL query execution interface. The SQL statement is as follows:

```
28 select b.kemu, avg(b.score)
29 from grade b
30 group by b.kemu
31
```

The results are displayed in a table with the following columns: kemu, avg(b.score).

kemu	avg(b.score)
语文	79
数学	84.25
英语	82.5

成绩排名

4. 列出数学成绩的排名，要求显示字段：学号，姓名，成绩，排名

在查询结果表里面添加一个变量@paiming, 让它自动加 1

```
SELECT
t.id, t.score as 数学分数, @paiming := @paiming+1 as 排名
FROM
    (SELECT b.id, b.score
    FROM grade b
    WHERE b.kemu = '数学'
    ORDER BY score
    DESC) AS t,
(SELECT @paiming := 0) r
```

```
42 SELECT
43 t.id, t.score as 数学分数, @paiming := @paiming+1 as 排名
44 FROM
45 (SELECT b.id, b.score
46 FROM grade b
47 WHERE b.kemu = '数学'
48 ORDER BY score
49 DESC) AS t,
50 (SELECT @paiming := 0) r
51
```

信息	结果1	概况	状态
id	数学分数	排名	
1002	98	1	
1004	97	2	
1001	86	3	
1003	56	4	

结合 student 表获取学生名称

```
SELECT
t.id, a.name, t.score as 数学分数, @paiming := @paiming+1 as 排名
FROM
    (SELECT b.id, b.score
    FROM grade b
    WHERE b.kemu = '数学'
    ORDER BY score
    DESC) AS t,
    (SELECT @paiming := 0) r,
    student a
WHERE a.id = t.id
```

```
52 SELECT
53 t.id, a.name, t.score as 数学分数, @paiming := @paiming+1 as 排名
54 FROM
55 (SELECT b.id, b.score
56 FROM grade b
57 WHERE b.kemu = '数学'
58 ORDER BY score
59 DESC) AS t,
60 (SELECT @paiming := 0) r,
61 student a
62 WHERE a.id = t.id
63
```

信息	结果1	概况	状态
id	name	数学分数	排名
1002	张四	98	1
1003	张五	97	2
1004	张六	97	3
1001	张三	86	4

同结果名次相同

上图由于同一个分数的小伙伴，排名不一样，本着公平、公正、公开的原则，同一分数名次一样

```
SELECT
t.id, a.name, t.score as 数学分数,
(CASE
WHEN @temp = t.score THEN
    @paiming
WHEN @temp := t.score THEN
    @paiming :=@paiming + 1
WHEN @temp = 0 THEN
    @paiming :=@paiming + 1
END) AS num

FROM
    (SELECT b.id, b.score
    FROM grade b
    WHERE b.kemu = '数学'
    ORDER BY score
    DESC) AS t,
    (SELECT @paiming := 0, @temp := 0) r,
    student a
WHERE a.id = t.id
```

```

52 SELECT
53 t.id, a.name, t.score as 数学分数,
54 (CASE
55 WHEN @temp = t.score THEN
56     @paiming
57 WHEN @temp := t.score THEN
58     @paiming := @paiming + 1
59 WHEN @temp = 0 THEN
60     @paiming := @paiming + 1
61 END) AS num
62
63 FROM
64 (SELECT b.id, b.score
65  FROM grade b
66  WHERE b.kemu = '数学'
67  ORDER BY score
68  DESC) AS t,
69 (SELECT @paiming := 0, @temp := 0) r,
70 student a
71 WHERE a.id = t.id
72

```

信息	结果1	概况	状态
id	name	数学分数	num
1002	张四	98	1
1003	张五	97	2
1004	张六	97	2
1001	张三	86	3

排名相同的占个名次

```

SELECT obj.id, obj.score as 数学,
       @rownum := @rownum + 1 AS num_tmp,
       @incnum := (CASE
WHEN @rowtotal = obj.score THEN
       @incnum
WHEN @rowtotal := obj.score THEN
       @rownum
END) AS 排名

FROM
(SELECT id, score
FROM grade
WHERE kemu = "数学"
ORDER BY
score DESC
) AS obj,
(SELECT @rownum := 0 , @rowtotal := NULL , @incnum := 0) r

```

```
53
54 SELECT obj.id, obj.score as 数学,
55     @rownum := @rownum + 1 AS num_tmp,
56     @incnum := (CASE
57 WHEN @rowtotal = obj.score THEN
58     @incnum
59 WHEN @rowtotal := obj.score THEN
60     @rownum
61 END) AS 排名
62
63 FROM
64 (SELECT id, score
65 FROM grade
66 WHERE kemu = "数学"
67 ORDER BY
68 score DESC
69 ) AS obj,
70 (SELECT @rownum := 0 ,@rowtotal := NULL ,@incnum := 0) r
71
```

信息	结果1	概况	状态
id	数学	num_tmp	排名
1002	98	1	1
1003	97	2	2
1004	97	3	2
1001	86	4	4

SQL 查询-3

前言

select top n 形式的语句可以获取查询的前几个记录,但是 mysql 没有此语法,mysql 用 limit 来实现相关功能。

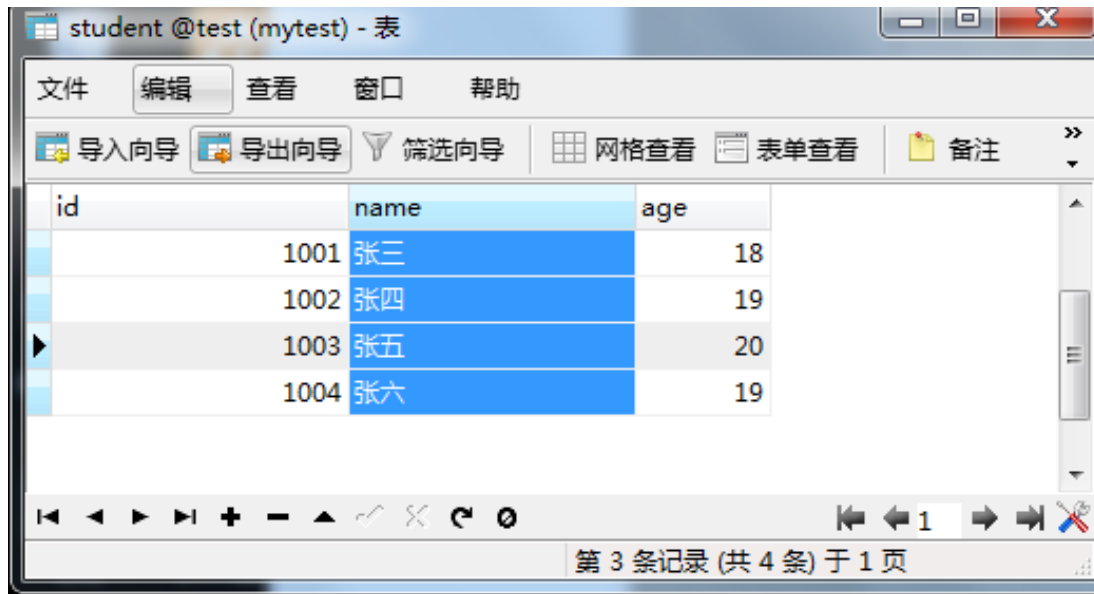
LIMIT 子句可以被用于强制 SELECT 语句返回指定的记录数。LIMIT 接受一个或两个数字参数。参数必须是一个整数常量。

如果给定两个参数,第一个参数指定第一个返回记录行的偏移量,第二个参数指定返回记录行的最大数目。

1. 列出数学成绩前 3 名的学生 (要求显示字段: 学号, 姓名, 科目, 成绩)
2. 查询数学成绩第 2 和第 3 名的学生
3. 查询第 3 名到后面所有的学生数学成绩
4. 统计英语课程少于 80 分的, 显示 学号 id, 姓名, 科目, 分数
5. 查找每科成绩前 2 名, 显示 id, 姓名, 科目, 分数

万年不变学生表

有 2 张表，学生表（student）基本信息如下



id	name	age
1001	张三	18
1002	张四	19
1003	张五	20
1004	张六	19

科目和分数表（grade）



no	id	kemu	score
1	1001	语文	85
2	1001	数学	86
3	1002	英语	98
4	1002	语文	94
5	1002	数学	98
6	1003	数学	56
7	1003	语文	69
8	1003	英语	68
9	1004	英语	99
10	1004	数学	97

查询前 3 名

1. 列出数学成绩前 3 名的学生（要求显示字段：学号，姓名，科目，成绩）

```
select *  
from grade
```

```
where kemu = '数学'  
order by score  
desc  
limit 3
```

```
5 select *  
6 from grade  
7 where kemu = '数学'  
8 order by score  
9 desc  
10 limit 3  
11
```

信息	结果1	概况	状态
no	id	kemu	score
▶	5	1002 数学	98
	6	1003 数学	97
	10	1004 数学	97

先通过 limit 取出前三条记录，再结合 student 表查询

```
select a.id, a.name, b.kemu, b.score  
from student a, grade b  
where a.id = b.id  
and kemu = '数学'  
order by score  
desc  
limit 3
```

```
5 select a.id, a.name, b.kemu, b.score  
6 from student a, grade b  
7 where a.id = b.id  
8 and kemu = '数学'  
9 order by score  
10 desc  
11 limit 3
```

信息	结果1	概况	状态
id	name	kemu	score
▶	1002 张四	数学	98
	1003 张五	数学	97
	1004 张六	数学	97

查询第 2-3 名记录

2. 查询数学成绩第 2 和第 3 名的学生

limit 后面如果只写一个整数 n，那就是查询的前 n 条记录；如果后面带 2 个整数 n 和 m，那么第一个数 n 就是查询出来队列的起点（从 0 开始），第二个是 m 是统计的总数目

第 2-3 条记录，那么起点就是 1，第 2-3 名有 2 条记录，那么第二个参数就是 2

```
select a.id, a.name, b.kemu, b.score
from student a, grade b
where a.id = b.id
and kemu = '数学'
order by score
desc
limit 1, 2
```

```
5 select a.id, a.name, b.kemu, b.score
6 from student a, grade b
7 where a.id = b.id
8 and kemu = '数学'
9 order by score
10 desc
11 limit 1, 2
12
```

信息	结果1	概况	状态
	id	name	kemu score
▶	1003	张五	数学 97
	1004	张六	数学 97

备注：limit 是按条数取的，名次一样的，也算一个记录。如果取第 5-14 的记录，那就是 limit 4 10

查询第 3 到后面所有的

3. 查询第 3 名到后面所有的学生数学成绩

```
select a.id, a.name, b.kemu, b.score
from student a, grade b
where a.id = b.id
and kemu = '数学'
order by score
desc
limit 3, 10000
```

注意：有些资料上写的 `limit 3, -1` 用 `-1` 代码最大值，这个是不对的，会报错，解决办法：随便写个非常大的整数

```
4
5 select a.id, a.name, b.kemu, b.score
6 from student a, grade b
7 where a.id = b.id
8 and kemu = '数学'
9 order by score
10 desc
11 limit 3, 10000
12
```

信息	结果1	概况	状态
id	name	kemu	score
1001	张三	数学	86

英语课程少于 80 分的人

4. 统计英语课程少于 80 分的，显示 学号 id，姓名，科目，分数

```
SELECT a.id, a.name, b.kemu, b.score
FROM student a, grade b
WHERE a.id = b.id
AND b.kemu = '英语'
AND b.score < 80
```

```
1
2
3 SELECT a.id, a.name, b.kemu, b.score
4 FROM student a, grade b
5 WHERE a.id = b.id
6 AND b.kemu = '英语'
7 AND b.score < 80
8
```

信息	结果1	概况	状态
id	name	kemu	score
1003	张五	英语	68
1001	张三	英语	65
1005	李四	英语	48

统计每门课程不及格、一般、优秀

课程 不及格 (<60) 一般 (60<= x <=80) 优秀 (>80)

```
SELECT b.kemu,
```

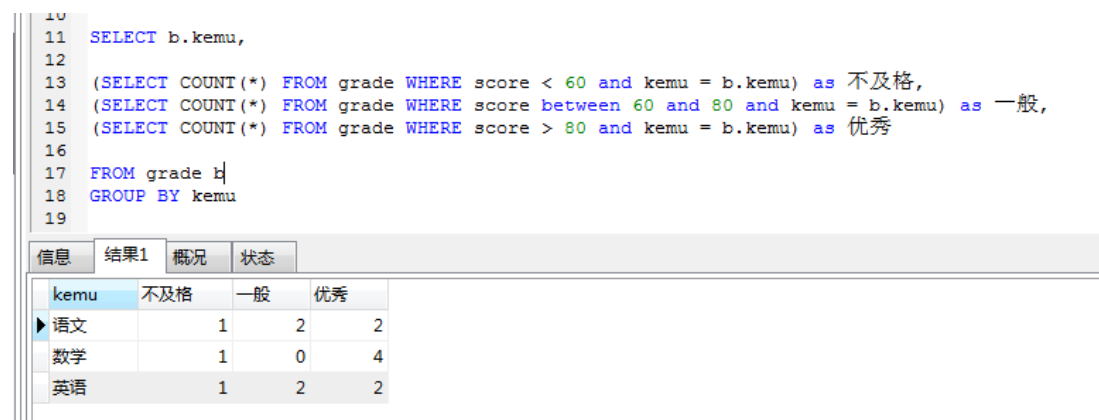
```
(SELECT COUNT(*) FROM grade WHERE score < 60 and kemu = b.kemu) as 不及格,
```

```
(SELECT COUNT(*) FROM grade WHERE score between 60 and 80 and kemu = b.kemu) as 一般,
```

```
(SELECT COUNT(*) FROM grade WHERE score > 80 and kemu = b.kemu) as 优秀
```

```
FROM grade b
```

```
GROUP BY kemu
```



```
10
11 SELECT b.kemu,
12
13 (SELECT COUNT(*) FROM grade WHERE score < 60 and kemu = b.kemu) as 不及格,
14 (SELECT COUNT(*) FROM grade WHERE score between 60 and 80 and kemu = b.kemu) as 一般,
15 (SELECT COUNT(*) FROM grade WHERE score > 80 and kemu = b.kemu) as 优秀
16
17 FROM grade b
18 GROUP BY kemu
19
```

信息	结果1	概况	状态
kemu	不及格	一般	优秀
语文	1	2	2
数学	1	0	4
英语	1	2	2

查找每科成绩前 2 名

5. 查找每科成绩前 2 名，显示 id，姓名，科目，分数

先按科目和分数查询

```
SELECT t1.id, t1.kemu, t1.score
```

```
FROM grade t1
```

```
ORDER BY t1.kemu, t1.score DESC
```



```

30
31 SELECT t1.id, t1.kemu,t1.score
32 FROM grade t1
33 ORDER BY t1.kemu,t1.score DESC
34

```

信息	结果1	概况	状态
id	kemu	score	
1005	数学	99	
1002	数学	98	
1003	数学	96	
1004	数学	96	
1001	数学	86	
1004	英语	99	
1002	英语	98	
1003	英语	68	
1001	英语	65	
1005	英语	48	
1002	语文	94	

再查找每个每科前面 2 名

```

SELECT t1.id, a.name, t1.kemu,t1.score
FROM grade t1, student a
WHERE
    (SELECT count(*) FROM grade t2
     WHERE t1.kemu=t2.kemu AND t2.score>=t1.score
    )<=2
and a.id = t1.id
ORDER BY t1.kemu,t1.score
DESC

```

```

23 SELECT t1.id, a.name, t1.kemu,t1.score
24 FROM grade t1, student a
25 WHERE
26   (SELECT count(*) FROM grade t2
27    WHERE t1.kemu=t2.kemu AND t2.score>=t1.score
28    )<=2
29 and a.id = t1.id
30 ORDER BY t1.kemu,t1.score
31 DESC
32

```

信息	结果1	概况	状态
id	name	kemu	score
1005	李四	数学	99
1002	张四	数学	98
1004	张六	英语	99
1002	张四	英语	98
1002	张四	语文	94
1001	张三	语文	85

如果第二名有重复的，也能一起查找出来

```
SELECT t1.id, a.name, t1.kemu, t1.score
FROM grade t1, student a
WHERE
    (SELECT count(*) FROM grade t2
     WHERE t1.kemu=t2.kemu AND t2.score>t1.score
    )<2
and a.id = t1.id
ORDER BY t1.kemu, t1.score
DESC
```

SQL 经典面试题-4

前言

用一条 SQL 语句查询 xuesheng 表每门课都大于 80 分的学生姓名，这个是面试考 sql 的一个非常经典的面试题

having 和 not in

查询 xuesheng 表每门课都大于 80 分的学生姓名

name kecheng score

张三	语文	81
张三	数学	73
李四	语文	86
李四	数学	90
王五	数学	89
王五	语文	88
王五	英语	96

解决办法一： having

如果不考虑学生的课程少录入情况（比如张三只有 2 个课程，王五有 3 个课程）

```
SELECT name
FROM xuesheng
GROUP BY name
```

```
HAVING MIN(score)> 80
```

如果考虑学生的课程数大于等于 3 的情况

```
SELECT name
FROM xuesheng
GROUP BY name
HAVING MIN(score)> 80
AND COUNT(kcheng)>=3
```

解决办法二: not in

可以用反向思维, 先查询出表里面有小于 80 分的 name, 然后用 not in 去除掉

```
SELECT DISTINCT name
FROM xuesheng
WHERE name NOT IN
    (SELECT DISTINCT name
     FROM xuesheng
     WHERE score <=80);
```

The screenshot shows a SQL query execution interface. The query is as follows:

```
6
7  SELECT DISTINCT name
8  FROM xuesheng
9  WHERE name NOT IN
10 (SELECT DISTINCT name
11  FROM xuesheng
12  WHERE score <=80);
```

Below the query, there are tabs for '信息' (Info), '结果1' (Result 1), '概况' (Overview), and '状态' (Status). The '结果1' tab is selected, showing the results of the query. The results are displayed in a table with the following data:

name
李四
王五

删除

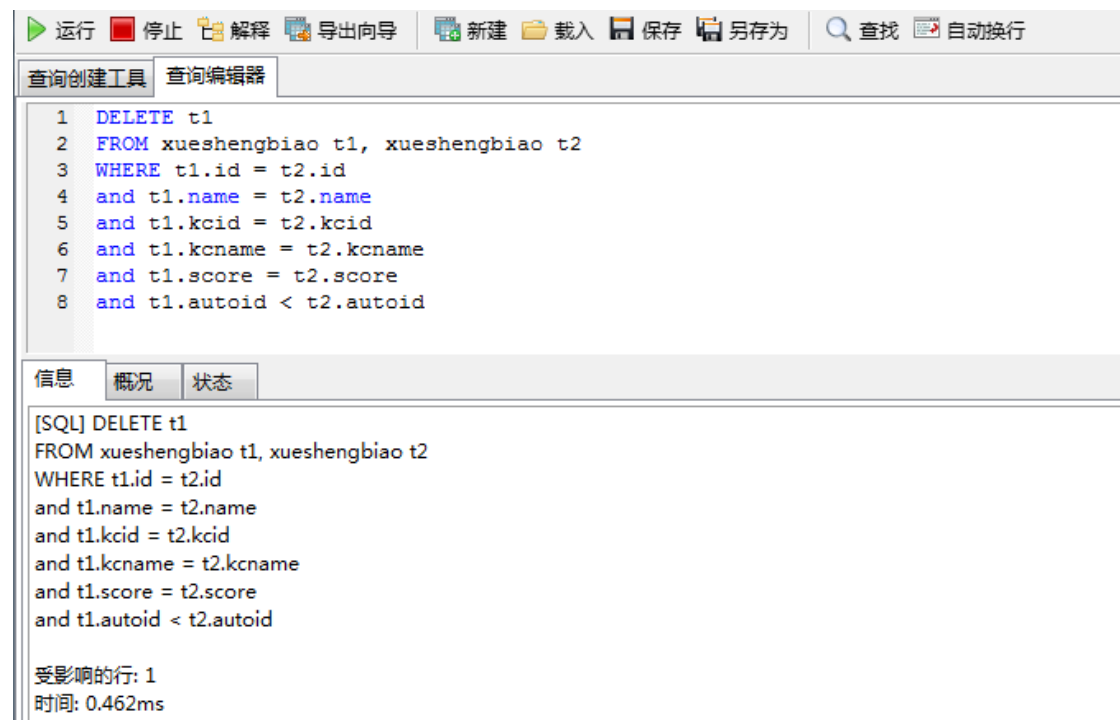
学生表 xueshengbiao 如下:自动编号 学号 姓名 课程编号 课程名称 分数

autoid	id	name	kcid	kname	score
1	2005001	张三	0001	数学	69
2	2005002	李四	0001	数学	89

autoid	id	name	kcid	kcname	score
3	2005001	张三	0001	数学	69

删除除了自动编号不同，其他都相同的学生冗余信息

```
DELETE t1
FROM xueshengbiao t1, xueshengbiao t2
WHERE t1.id = t2.id
and t1.name = t2.name
and t1.kcid = t2.kcid
and t1.kcname = t2.kcname
and t1.score = t2.score
and t1.autoid < t2.autoid
```



如果只是查询出自动编号不同，其他都相同的学生冗余信息，可以用 group by

```
SELECT * from xueshengbiao t1
WHERE t1.autoid
NOT IN
    (SELECT MIN(autoid) as autoid FROM xueshengbiao
    GROUP BY id, name, kcid, kcname, score)
```

```
10 SELECT * from xueshengbiao t1
11 WHERE t1.autoid
12 NOT IN
13 (SELECT MIN(autoid) as autoid FROM xueshengbiao
14 GROUP BY id, name, kcid, kcname, score)
```

信息		结果1		概况		状态	
	autoid	id	name	kcid	kcname	score	
▶	3	2005001	张三	0001	数学	69	

模糊查询%

表名: student ,用 sql 查询出“张”姓学生中平均成绩大于 75 分的学生信息;

name kecheng score

张青 语文 72

张华 英语 81

王华 数学 72

张青 物理 67

李立 化学 98

张青 化学 76

```
select * from student
```

```
where name in
```

```
(select name from student
```

```
where name like '张%' group by name having avg(score) > 75);
```

SQL 通配符

在 SQL 中,通配符与 SQL LIKE 操作符一起使用。SQL 通配符用于搜索表中的数据。在 SQL 中,可使用以下通配符:

通配符	描述
%	替代 0 个或多个字符
_	替代一个字符
[charlist]	字符列中的任何单一字符
[^charlist]或[!charlist]	不在字符列中的任何单一字符

MySQL 中使用 REGEXP 或 NOT REGEXP 运算符（或 RLIKE 和 NOT RLIKE）来操作正则表达式

找出姓张和姓李的同学，用 rlike 实现匹配多个

-- 找出姓张和姓李的

```
select * from xuesheng
where name in
(select name from xuesheng
where name rlike '[张李]' group by name having avg(score) > 75);
```

也可以用 REGEXP，结合正则匹配

```
select * from xuesheng
where name in
(select name from xuesheng
where name REGEXP '^[张李]' group by name having avg(score) > 75);
```

二、python 篇

个人博客地址: <http://www.cnblogs.com/yoyoketang/>

QQ 交流群: 779429633

联系本人 qq: 283340479

编程语言最近考的也很多了，一般以基础为主，有的自动化岗位，偏测试开发的岗位，对编程的能力要求会高一点，一般会让写一小段程序。

python 笔试题-1

前言

现在面试测试岗位，一般会要求熟悉一门语言（python/java），为了考验求职者的基本功，一般会出 2 个笔试题，这些题目一般不难，主要考察基本功。要是给你一台电脑，在编辑器里面边写边调试，没多大难度。主要是给你一张纸

交流 qq 群: **779429633**

和笔，让你现场写出来，那就没那么容易了。

（本篇代码都是基于 python3.6）

1. 统计

统计在一个队列中的数字，有多少个正数，多少个负数，如[1, 3, 5, 7, 0, -1, -9, -4, -5, 8]

方法一

```
# coding:utf-8
a = [1, 3, 5, 7, 0, -1, -9, -4, -5, 8]
```

```
# 用列表生成式，生成新的列表
b = [i for i in a if i > 0]
print("大于 0 的个数: %s" % len(b))
```

```
c = [i for i in a if i < 0]
print("小于 0 的个数: %s" % len(c))
```

方法二

```
# coding:utf-8
a = [1, 3, 5, 7, 0, -1, -9, -4, -5, 8]
```

```
# 用传统的判断思维，累加
m = 0
n = 0
for i in a:
    if i > 0:
        m += 1
    elif i < 0:
        n += 1
    else:
        pass
print("大于 0 的个数: %s" % m)
print("小于 0 的个数: %s" % n)
```

2. 字符串切片

字符串 "axbycdj"，如果得到结果 "abcd"

方法一

```
# 字符串切片
```

```
a = "axbyczdj"
print(a[::2])
```

方法二

传统思维

```
a = "axbyczdj"

c = []
for i in range(len(a)):
    if i % 2 == 0:
        c.append(a[i])
print("".join(c))
```

3. 字符串切割

已知一个字符串为“hello_world_yoyo”，如何得到一个队列
["hello", "world", "yoyo"]

```
a = "hello_world_yoyo"
b = a.split("_")
print(b)
```

4. 格式化输出

已知一个数字为1，如何输出“0001”

```
a = 1

print("%04d" % a)
```

5. 队列

已知一个队列, 如: [1, 3, 5, 7], 如何把第一个数字, 放到第三个位置, 得到:
[3, 5, 1, 7]

```
a = [1, 3, 5, 7]

# insert 插入数据
a.insert(3, a[0])
print(a[1:])
```

6. 交换

已知 $a = 9$, $b = 8$, 如何交换 a 和 b 的值, 得到 a 的值为 8, b 的值为 9

方法 1

```
a = 8
b = 9

a, b = b, a
print(a)
print(b)
```

方法 2

```
a = 8
b = 9

# 用中间变量 c
c = a
a = b
b = c
print(a)
print(b)
```

7. 水仙花

打印出 100-999 所有的“水仙花数”, 所谓“水仙花数”是指一个三位数, 其各位数字立方和等于该数本身。例如: 153 是一个“水仙花数”, 因为 $153=1$ 的三次方+ 5 的三次方+ 3 的三次方。

```
sxh = []
for i in range(100, 1000):
    s = 0
    m = list(str(i))
    for j in m:
        s += int(j)**len(m)
    if i == s:
        print(i)
        sxh.append(i)

print("100-999 的水仙花数: %s" % sxh)
```

8. 完全数

如果一个数恰好等于它的因子之和，则称该数为“完全数”，又称完美数或完备数。例如：第一个完全数是 6，它有约数 1、2、3、6，除去它本身 6 外，其余 3 个数相加，

$1+2+3=6$ 。第二个完全数是 28，它有约数 1、2、4、7、14、28，除去它本身 28 外，其余 5 个数相加， $1+2+4+7+14=28$ 。

那么问题来了，求 1000 以内的完全数有哪些？

```
a = []

for i in range(1, 1000):
    s = 0
    for j in range(1, i):
        if i % j == 0 and j < i:
            s += j
    if s == i:
        print(i)
        a.append(i)
print("1000 以内完全数: %s" % a)
```

9. 排序

用 python 写个冒泡排序

```
a = [1, 3, 10, 9, 21, 35, 4, 6]

s = range(1, len(a))[:-1]
print(list(s)) # 交换次数

for i in s:
    for j in range(i):
        if a[j] > a[j + 1]:
            a[j], a[j + 1] = a[j + 1], a[j]
    print("第 %s 轮交换后数据: %s" % (len(s)-i+1, a))
print(a)
```

10. sort 排序

已知一个队列[1, 3, 6, 9, 7, 3, 4, 6]

- 按从小到大排序
- 按从大大小排序
- 去除重复数字

```
a = [1, 3, 6, 9, 7, 3, 4, 6]
```

```
# 1. sort 排序, 正序
```

```
a.sort()
```

```
print(a)
```

```
# 2. sort 倒叙
```

```
a.sort(reverse=True)
```

```
print(a)
```

```
# 3. 去重
```

```
b = list(set(a))
```

```
print(b)
```

python 笔试题(递归)-2

前言

本篇继续收集一些常见的 python 笔试题，以基础知识为主，递归是面试最喜欢考的一个问题，不管是做开发还是测试，都无法避免考递归。本篇结合实际案例，讲下几种关于递归的场景。

计算 n 的阶乘

计算 $n!$ ，例如 $n=3$ (计算 $3! = 6$)，求 $10!$

方法 1: 可以用 python 里面的 `reduce` 函数，`reduce()` 函数会对参数序列中元素进行累积。

函数将一个数据集合(链表, 元组等)中的所有数据进行下列操作: 用传给 `reduce` 中的函数 `function` (有两个参数) 先对集合中的第 1、2 个元素进行操作，得到的结果再与第三个数据用 `function` 函数运算，最后得到一个结果。

```
from functools import reduce
```

```
# 方法 1: 推荐!
```

```
a = 10
```

```
b = reduce(lambda x, y: x*y, range(1, a+1))  
print(b)
```

如果不想用 lambda 函数，可以定义一个函数

```
from functools import reduce  
  
def chengfa(x, y):  
    return x*y  
  
a = 10  
b = reduce(chengfa, range(1, a+1))  
print(b)
```

方法 2：自己写个递归函数

```
def digui(n):  
    if n == 1:  
        return 1  
    else:  
        return n*digui(n-1)  
  
a = 10  
print(digui(a))
```

方法 3：用 for 循环（不推荐!）

```
a = 10
s = 1
for i in range(1, a+1):
    s = s*i
print(s)
```

斐波那契数列

已知一个数列：1、1、2、3、5、8、13、。。。。的规律为从 3 开始的每一项都等于其前两项的和，这是斐波那契数列。求满足规律的 100 以内的所有数据

```
a = 0
b = 1
while b < 100:
    print(b, end=",")
    a, b = b, a+b
```

幂的递归

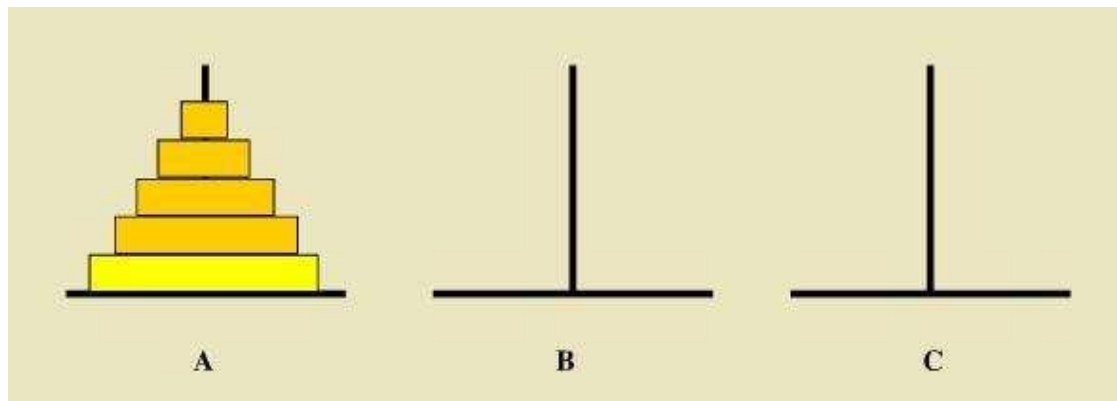
计算 x 的 n 次方，如：3 的 4 次方 为 $3*3*3*3=81$

```
def mi(x, n):
    '''计算 x 的 n 次方'''
    if n == 0:
        return 1
    else:
        return x*mi(x, n-1)
```

```
x = 3  
num = 4  
print(mi(x, num))
```

汉诺塔问题

汉诺塔：汉诺塔（又称河内塔）问题是源于印度一个古老传说的益智玩具。大梵天创造世界的时候做了三根金刚石柱，在一根柱子上从下往上按照大小顺序摞着 64 片黄金圆盘。大梵天命令婆罗门把圆盘从下面开始按大小顺序重新摆放在另一根柱子上。并且规定，在小圆盘上不能放大圆盘，在三根柱子之间一次只能移动一个圆盘



当只有一个盘子的时候，只需要从将 A 塔上的一个盘子移到 C 塔上。

当 A 塔上有两个盘子是，先将 A 塔上的 1 号盘子(编号从上到下)移动到 B 塔上，再将 A 塔上的 2 号盘子移动的 C 塔上，最后将 B 塔上的小盘子移动到 C 塔上。

当 A 塔上有 3 个盘子时，先将 A 塔上编号 1 至 2 的盘子（共 2 个）移动到 B 塔上（需借助 C 塔），然后将 A 塔上的 3 号最大的盘子移动到 C 塔，最后将 B 塔上的两个盘子借助 A 塔移动到 C 塔上。

当 A 塔上有 n 个盘子是，先将 A 塔上编号 1 至 n-1 的盘子（共 n-1 个）移动到 B 塔上（借助 C 塔），然后将 A 塔上最大的 n 号盘子移动到 C 塔上，最后将 B 塔上的 n-1 个盘子借助 A 塔移动到 C 塔上。

综上所述，除了只有一个盘子时不需要借助其他塔外，其余情况均一样（只是事件的复杂程度不一样）。

```
def hanoi(n, a, b, c):  
    '''汉诺塔问题'''  
    if n == 1:
```

```
        print(a, '-->', c)

    else:

        hanoi(n - 1, a, c, b)

        print(a, '-->', c)

        hanoi(n - 1, b, a, c)

hanoi(5, 'A', 'B', 'C')
```

一般汉诺塔问题不会经常考，前面几个考的比较频繁

python 编程题-3

有写公司会要求能写一小段程序出来，如题：

写一个小程序：控制台输入邮箱地址（格式为 username@companyname.com），程序识别用户名和公司名后，将用户名和公司名输出到控制台。

要求：

1. 校验输入内容是否符合规范（xx@yy.com），如是进入下一步，如否则抛出提示“incorrect email format”。注意必须以.com 结尾
2. 可以循环“输入—输出判断结果”这整个过程
3. 按字母Q（不区分大小写）退出循环，结束程序

```
import re

# 判断邮箱.com 结尾

def is_mail_style(x):

    a =
```

```
re.match(r'^[0-9a-zA-Z\_\-]*@[0-9a-zA-Z]+(\.com)$', x)

    if a:

        yhm = re.findall("^(.+?)@", x)

        print("用户名:%s " %yhm[0])

        gc = re.findall("@(.+?)\.com", x)

        print("公司名:%s " %gc[0])

        return True

    else:

        print("incorrect email format")

        return False

a = input("请输入: ")

while 1:

    if a == "q" or a == "Q":

        exit()

    else:

        if is_mail_style(a):

            break

        a = input("请输入: ")

print("下一步! ")
```


python 编程 4-遍历文件

如何遍历查找出某个文件夹内所有的子文件呢？并且找出某个后缀的所有文件

```
# coding:utf-8

import os

def get_files(path='D:\\xx', rule=".py"):

    all = []

    for fpathe,dirs,fs in os.walk(path):    # os.walk 是获取所有的目录

        for f in fs:

            filename = os.path.join(fpathe,f)

            if filename.endswith(rule):    # 判断是否是"xxx"结尾

                all.append(filename)

    return all

if __name__ == "__main__":

    b = get_files(r"D:\test\python2")

    for i in b:

        print(i)
```

三、linux 篇

个人博客地址: <http://www.cnblogs.com/yoyoketang/>

QQ 交流群: 779429633

联系本人 qq: 283340479

linux 常用指令-1

前言

现在做测试的出去面试, 都会被问到 linux, 不会几个 linux 指令都不好意思说自己是做测试的了, 本篇收集了几个被问的频率较高的 linux 面试题

常用指令

1. 说出 10 个 linux 常用的指令

- ls 查看目录中的文件
- cd /home 进入 '/home' 目录; cd .. 返回上一级目录; cd ../../ 返回上两级目录
- mkdir dir1 创建一个叫做 'dir1' 的目录
- rmdir dir1 删除一个叫做 'dir1' 的目录 (只能删除空目录)
- rm -f file1 删除一个叫做 'file1' 的文件, -f 参数, 忽略不存在的文件, 从不给出提示。
- rm -rf /mulu 目录下面文件以及子目录下文件
- cp /test1/file1 /test3/file2 如将/test1 目录下的 file1 复制到 /test3 目录, 并将文件名改为 file2
- mv /test1/file1 /test3/file2 如将/test1 目录下的 file1 移动到 /test3 目录, 并将文件名改为 file2
- mv * ../ Linux 当前目录所有文件移动到上一级目录
- ps -ef|grep xxx 显示进程 pid

交流 qq 群: 779429633

- kill 使用 kill 命令来终结进程。先使用 ps 命令找到进程 id, 使用 kill -9 命令, 终止进程。
- tar -xvf file.tar 解压 tar 包
- unzip file.zip 解压 zip
- unrar e file.rar 解压 rar
- free -m 查看服务器内存使用情况

ps 查看进程

2. 如何查看所有 java 进程

- grep 是搜索关键字

ps -ef | grep java

- -aux 显示所有状态

ps -aux | grep java

kill 杀掉进程

3. 如何杀掉某个服务的进程

- kill 命令用于终止进程
- -9 强迫进程立即停止

kill -9 [PID]

这里 pid 需要用 ps -ef | grep 查询 pid

```
[root@localhost bin]# ps -ef|grep tomcat
root      5933      1  92 10:17 pts/0    00:00:19 /opt/jdk1.7.0_71/bin.
til.logging.manager=org.apache.juli.ClassLoaderLogManager -Xms512m -Xm
=2048 -Djava.endorsed.dirs=/usr/local/wedora/tomcat7/endorsed -classpath
atalina.base=/usr/local/wedora/tomcat7 -Dcatalina.home=/usr/local/wedor
start
root      6091    2898  0 10:17 pts/0    00:00:00 grep tomcat
[root@localhost bin]#
```

启动服务

4. 如何启动服务

以启动 Tomcat 为例, 先 cd 到启动的. sh 文件目录

```
> cd /java/tomcat/bin  
> ./startup.sh
```

停止 Tomcat 服务命令

```
./shutdown.sh
```

查看日志

5. 如何查看测试项目的日志

一般测试的项目里面，有个 logs 的目录文件，会存放日志文件，有个 xxx.out 的文件，可以用 tail -f 动态实时查看后端日志

先 cd 到 logs 目录(里面有 xx.out 文件)

```
tail -f xx.out
```

这时屏幕上会动态实时显示当前的日志，ctr+c 停止

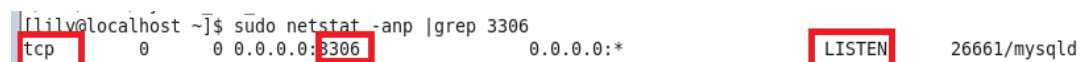
6. 如何查看最近 1000 行日志

```
tail -1000 xx.out
```

查看端口

7. LINUX 中如何查看某个端口是否被占用

```
netstat -anp | grep 端口号
```



```
lilv@localhost ~]$ sudo netstat -anp |grep 3306  
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN 26661/mysqld
```

图中主要看监控状态为 LISTEN 表示已经被占用，最后一列显示被服务 mysqld 占用，查看具体端口号，只要有如图这一行就表示被占用了

查看 82 端口的使用情况，如图

```
netstat -anp |grep 82
```

```
[lily@localhost ~]$ sudo netstat -anp |grep 82
unix 2      [ ACC ]     STREAM     LISTENING   108828 27228/firefox      /tmp/orbit-root/linc-6a5c-0-78d317ffe80bb
unix 2      [ ACC ]     STREAM     LISTENING   17886  2617/gnome-keyring- /tmp/orbit-root/linc-a39-0-3984082358bcf
unix 2      [ ACC ]     STREAM     LISTENING   18271  2682/gnome-panel    /tmp/orbit-root/linc-a7a-0-63105032174ec
unix 2      [ ACC ]     STREAM     LISTENING   18348  2690/bonobo-activat /tmp/orbit-root/linc-a82-0-3b38a6fe3c5ff
unix 2      [  ]       DGRAM      13682  2020/hald            @/org/freedesktop/hal/udev_event
unix 2      [  ]       DGRAM      113829 27694/pickup
unix 3      [  ]       STREAM     CONNECTED   108827 2650/gconfd-2       /tmp/orbit-root/linc-a5a-0-29d43c21f2624
unix 3      [  ]       STREAM     CONNECTED   108826 27228/firefox
unix 3      [  ]       STREAM     CONNECTED   108825 2636/dbus-daemon    @/tmp/dbus-Px0fajTqB5
unix 3      [  ]       STREAM     CONNECTED   108824 27228/firefox
unix 3      [  ]       STREAM     CONNECTED   20313  2826/gvfsd-metadata
unix 3      [  ]       STREAM     CONNECTED   19986  2682/gnome-panel    /tmp/orbit-root/linc-a7a-0-63105032174ec
unix 3      [  ]       STREAM     CONNECTED   19984  2682/gnome-panel    /tmp/orbit-root/linc-a7a-0-63105032174ec
unix 3      [  ]       STREAM     CONNECTED   19982  2770/gnote          /tmp/orbit-root/linc-a82-0-5e5bd575285b
unix 3      [  ]       STREAM     CONNECTED   19981  2682/gnome-panel
unix 3      [  ]       STREAM     CONNECTED   19978  2690/bonobo-activat /tmp/orbit-root/linc-a82-0-3b38a6fe3c5ff
unix 3      [  ]       STREAM     CONNECTED   19975  2682/gnome-panel
unix 3      [  ]       STREAM     CONNECTED   19882  2705/gvfsd-trash    @/dbus-vfs-daemon/socket-wUfJjKIq
unix 3      [  ]       STREAM     CONNECTED   19863  2690/bonobo-activat /tmp/orbit-root/linc-a82-0-3b38a6fe3c5ff
unix 3      [  ]       STREAM     CONNECTED   19782  2636/dbus-daemon    @/tmp/dbus-Px0fajTqB5
unix 3      [  ]       STREAM     CONNECTED   19748  2682/gnome-panel    /tmp/orbit-root/linc-a7a-0-63105032174ec
unix 3      [  ]       STREAM     CONNECTED   19745  2682/gnome-panel
unix 3      [  ]       STREAM     CONNECTED   19734  2690/bonobo-activat /tmp/orbit-root/linc-a82-0-3b38a6fe3c5ff
unix 3      [  ]       STREAM     CONNECTED   19722  2682/gnome-panel    /tmp/orbit-root/linc-a7a-0-63105032174ec
unix 3      [  ]       STREAM     CONNECTED   19719  2682/gnome-panel
unix 3      [  ]       STREAM     CONNECTED   19691  2690/bonobo-activat /tmp/orbit-root/linc-a82-0-3b38a6fe3c5ff
unix 3      [  ]       STREAM     CONNECTED   19393  2682/gnome-panel    /tmp/orbit-root/linc-a7a-0-63105032174ec
unix 3      [  ]       STREAM     CONNECTED   19282  2650/gconfd-2
unix 3      [  ]       STREAM     CONNECTED   19281  2682/gnome-panel    /tmp/orbit-root/linc-a7a-0-63105032174ec
unix 3      [  ]       STREAM     CONNECTED   19260  2682/gnome-panel
unix 3      [  ]       STREAM     CONNECTED   19259  2682/gnome-panel
unix 3      [  ]       STREAM     CONNECTED   18452  2690/bonobo-activat /tmp/orbit-root/linc-a82-0-3b38a6fe3c5ff
unix 3      [  ]       STREAM     CONNECTED   18447  2690/bonobo-activat /tmp/orbit-root/linc-a82-0-3b38a6fe3c5ff
```

可以看出并没有 LISTEN 那一行，所以就表示没有被占用。此处注意，图中显示的 LISTENING 并不表示端口被占用，不要和 LISTEN 混淆哦，查看具体端口时候，必须要看到 tcp，端口号，LISTEN 那一行，才表示端口被占用了

查看当前所有已经使用的端口情况，如图：

netstat -nulp (此处不用加端口号)

```
[lily@localhost ~]$ sudo netstat -nulp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:9000          0.0.0.0:*               LISTEN      26481/php-fpm
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      26661/mysqld
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2136/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1970/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      2295/master
tcp        0      0 :::8080                  :::*                    LISTEN      27977/httpd
tcp        0      0 :::22                    :::*                    LISTEN      2136/sshd
tcp        0      0 :::1:631                 :::*                    LISTEN      1970/cupsd
tcp        0      0 :::1:25                   :::*                    LISTEN      2295/master
udp        0      0 0.0.0.0:631             0.0.0.0:*               1970/cupsd
udp        0      0 0.0.0.0:68              0.0.0.0:*               1973/dhclient
```

find 查找文件

8. 如何查找一个文件大小超过 5M 的文件

```
find . -type f -size +100M
```

9. 如果知道一个文件名称，怎么查这个文件在 linux 下的哪个目录，如：要查找 tnsnames.ora 文件

```
find / -name tnsnames.ora
```

查到:

```
/opt/app/oracle/product/10.2/network/admin/tnsnames.ora
/opt/app/oracle/product/10.2/network/admin/samples/tnsnames.ora
```

还可以用 locate 来查找

```
locate tnsnames.ora
```

结果是:

```
/opt/app/oracle/product/10.2/hs/admin/tnsnames.ora.sample
/opt/app/oracle/product/10.2/network/admin/tnsnames.ora
/opt/app/oracle/product/10.2/network/admin/samples/tnsnames.ora
```

10. find 查找文件

```
find / -name httpd.conf    #在根目录下查找文件 httpd.conf，表示在整个
                             硬盘查找
find /etc -name httpd.conf  #在/etc 目录下文件 httpd.conf
find /etc -name 'srm'       #使用通配符 (0 或者任意多个)。表示在/etc 目录下
                             查找文件名中含有字符串 'srm' 的文件
find . -name 'srm'         #表示当前目录下查找文件名开头是字符串 'srm' 的
                             文件
```

按照文件特征查找

```
find / -amin -10          # 查找在系统中最后 10 分钟访问的文件(access time)
find / -atime -2          # 查找在系统中最后 48 小时访问的文件
find / -empty             # 查找在系统中为空的文件或者文件夹
find / -group cat         # 查找在系统中属于 group 为 cat 的文件
find / -mmin -5           # 查找在系统中最后 5 分钟里修改过的文件(modify time)
find / -mtime -1          #查找在系统中最后 24 小时里修改过的文件
find / -user fred         #查找在系统中属于 fred 这个用户的文件
find / -size +10000c      #查找出大于 10000000 字节的文件(c:字节, w:双字,
k:KB, M:MB, G:GB)
find / -size -1000k       #查找出小于 1000KB 的文件
```

测试环境搭建-tomcat

测试环境如何搭建?

通常面试会问到会不会搭建测试环境? 到底啥是测试环境搭建呢, 其实测试环境没有想像的那么高大上, 弄个 tomcat, 把测试的 war 包丢进去, 重启下服务就可以了。

环境准备

Java 和 tomcat 的环境准备参考这篇 <https://www.cnblogs.com/yoyoketang/p/10186513.html>

以 jenkins 搭建为例（web 网站搭建都是一个套路）

前言

通常做自动化测试，需要用到 jenkins 来做持续集成，那么 linux 系统里面如何使用 tomcat 安装 Jenkins 环境呢？

前面一篇已经搭建好 java 和 tomcat 环境，接着直接下载 jenkins 的 war 包放到 tomcat 下就能部署了。

环境准备：

1. 一台 Linux 服务器，操作系统：CentOS 7.4 64 位
2. Xshell 5.0
3. java 1.8.0
4. tomcat 7.0.76
5. wget

wget

wget 是一个从网络上自动下载文件的自由工具，支持通过 HTTP、HTTPS、FTP 三个最常见的 TCP/IP 协议 下载，并可以使用 HTTP 代理。“wget” 这个名称来源于 “World Wide Web” 与 “get” 的结合。

所谓自动下载，是指 wget 可以在用户退出系统的之后在继续后台执行，直到下载任务完成。

打开 Xshell 连上服务器，先看 CentOS 系统有没自带 wget 这个工具：`rpm -qa | grep wget`

```
[root@yoyo ~]# rpm -qa | grep wget
wget-1.14-15.el7.x86_64
[root@yoyo ~]#
```

能查看到版本号，说明已经安装过了，没有的话，就用 yum 安装下

```
yum install wget
```

下载 jenkins

jenkins 安装包下载地址：

<http://mirrors.jenkins-ci.org/war/latest/jenkins.war>

可以在本地下载之后，使用 Xftp 工具传到 tomcat 的 “/usr/share/tomcat/webapps” 目录下。

这里我们介绍直接使用 wget 工具下载到 linux 服务器上, 先在 webapps 里面新建一个 jenkins 目录

```
[root@yoyo ~]# mkdir /usr/share/tomcat/webapps/jenkins
[root@yoyo ~]# cd /usr/share/tomcat/webapps/jenkins
[root@yoyo jenkins]# wget
http://mirrors.jenkins-ci.org/war/latest/jenkins.war
```

wget 下载速度太慢(10k 左右)了, 跟蜗牛一样, 可以更换下阿里源

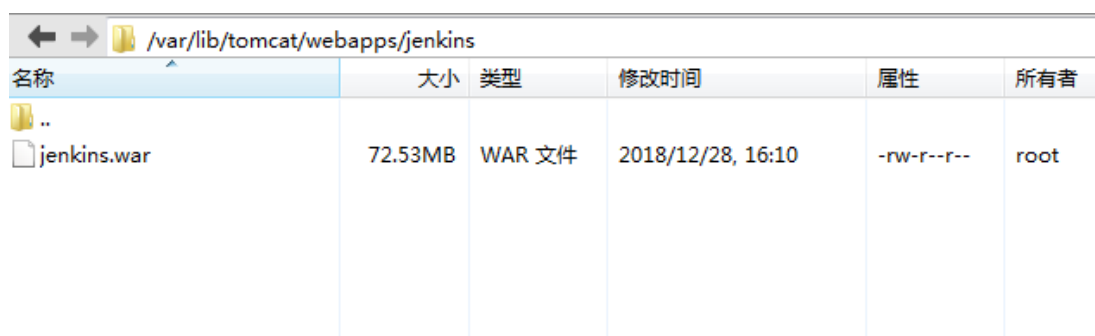
```
cd /etc/yum.repos.d/
mv CentOS-Base.repo CentOS-Base.repo.back #建议备份或者改名
wget -O CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
yum makecache #生成缓存
```

替换完之后, 接着 cd 到 /usr/share/tomcat/webapps/jenkins 再用 wget 下载 war 包, 速度在 100k 左右, 喝杯咖啡就下载完成了

```
[root@yoyo ~]# cd /usr/share/tomcat/webapps/jenkins
[root@yoyo jenkins]# wget
http://mirrors.jenkins-ci.org/war/latest/jenkins.war
```

注意: 最好 cd 到 /usr/share/tomcat/webapps/jenkins 目录再执行下载, 要不然下载完之后, 还得移过去(如下 cp 命令)

```
cp jenkins.war /usr/share/tomcat/webapps/jenkins
```



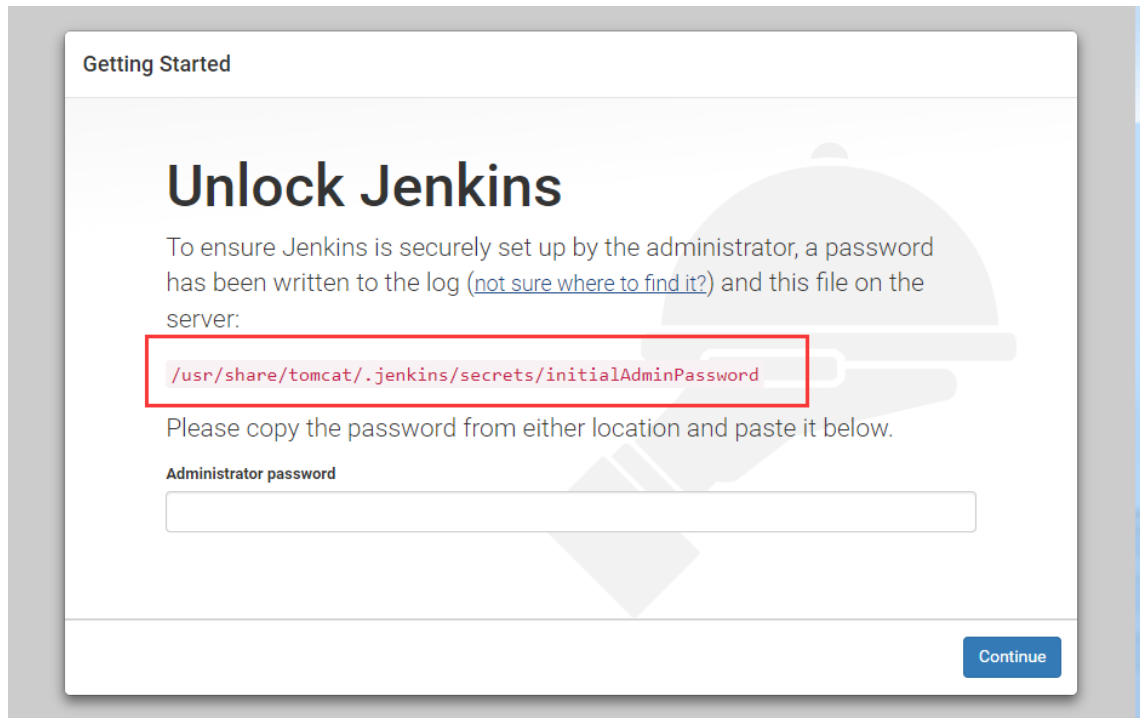
/var/lib/tomcat/webapps/jenkins					
名称	大小	类型	修改时间	属性	所有者
..					
jenkins.war	72.53MB	WAR 文件	2018/12/28, 16:10	-rw-r--r--	root

下载完之后, 解压 war 包: jar -xvf jenkins.war

```
[root@yoyo /]# cd /usr/share/tomcat/webapps/jenkins
[root@yoyo jenkins]# jar -xvf jenkins.war
```

启动 jenkins

解压完之后，只要 tomcat 是启动的，浏览器输入：`http://公网 ip:8080/jenkins` 就能访问了（打不开的话重启 tomcat）

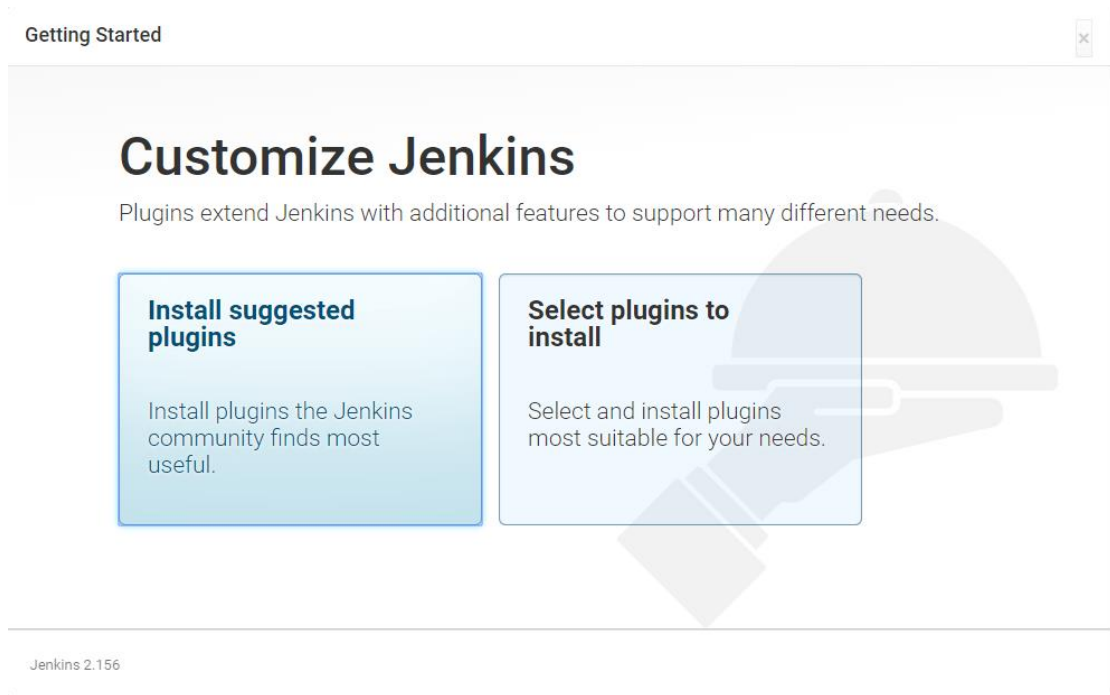


首次打开，在 `/usr/share/tomcat/.jenkins/secrets/initialAdminPassword` 目录找到密码，输入进去就能访问了

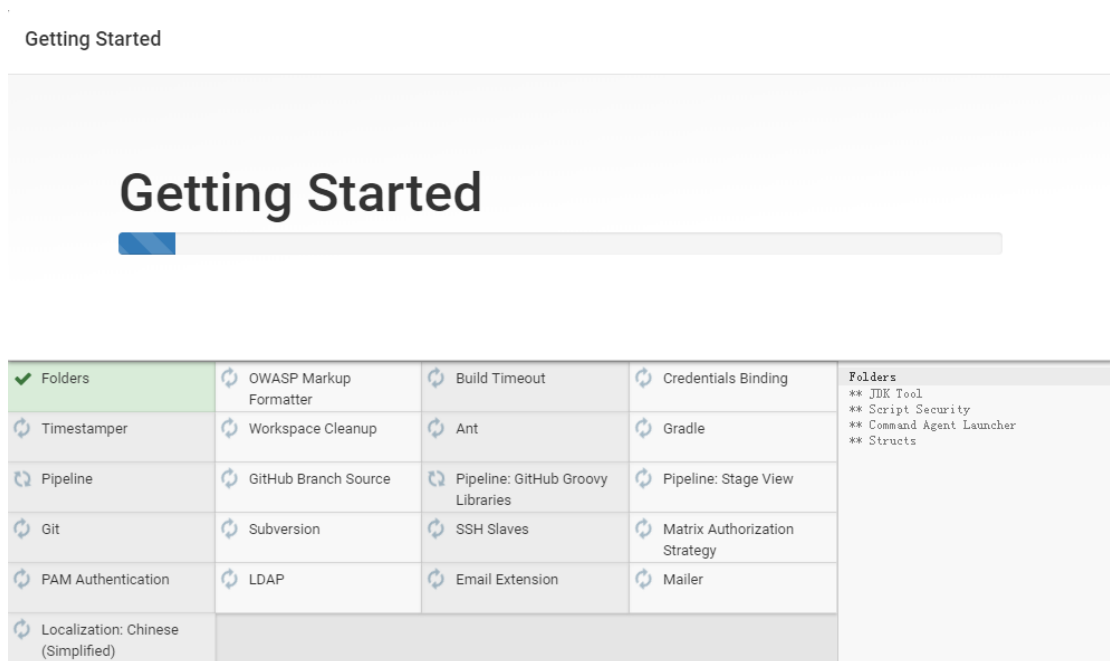
```
[root@yoyo tomcat]# cd /
[root@yoyo /]# cd
/usr/share/tomcat/.jenkins/secrets/initialAdminPassword
-bash: cd: /usr/share/tomcat/.jenkins/secrets/initialAdminPassword:
Not a directory
[root@yoyo /]# cd /usr/share/tomcat/.jenkins/secrets
[root@yoyo secrets]# vim initialAdminPassword
```

9475379d346744cea5056130e1ab85xx

把密码复制了，贴到浏览器输入框就能访问了，弹出的框选左边这个安装全部插件



接下来喝杯咖啡，让它全部下载完成就可以了



如果 Getting Started 页面遇到插件下载失败，打开 jenkins->系统管理->管理插件->高级

把: <https://updates.jenkins-ci.org/update-center.json>

换成: <http://mirror.esuni.jp/jenkins/updates/update-center.json>

或者 https 换成 http: <http://updates.jenkins.io/update-center.json>

Advanced...

Submit

Upload Plugin

You can upload a .hpi file to install a plugin from outside the central plugin repository.

File: 未选择任何文件


Upload

Update Site


URL


Submit


插件下载完成后，就可以开始正常使用 jenkins 了


 **Jenkins** [admin](#) | [log out](#)


Jenkins > [ENABLE AUTO REFRESH](#) [add description](#)


 New 任务


 People


 Build History

 系统管理



 我的视图

 Lockable Resources

 凭据

 新建视图

所有 +

S	W	名称 ↓	Last Success	Last Failure	Last Duration
		22	N/A	N/A	N/A

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

四、接口篇

个人博客地址: <http://www.cnblogs.com/yoyoketang/>

QQ 交流群: 779429633

联系本人 qq: 283340479

交流 qq 群: 779429633

http 协议相关面试题

前言

在 PC 浏览器的地址栏输入一串 URL，然后按 Enter 键这个页面渲染出来，这个过程中都发生了什么事?这个是很多面试官喜欢问的一个问题

如果测试只是停留在表面上点点点，不知道背后的逻辑，是无法发现隐藏的 bug，只能找一些页面上看得到的 bug。

测试人员如果想在技术上有所提升，必然要都懂接口（API）测试，这也是近年来越来越多的公司意识到接口测试的重要性，招聘的时候要招一个中高级的测试人员，接口测试是必备技能了。

浏览器输入 url 按回车背后经历了哪些？

1. 在 PC 浏览器的地址栏输入一串 URL，然后按 Enter 键这个页面渲染出来，这个过程中都发生了什么事？

- 1、首先，在浏览器地址栏中输入 url，先解析 url，检测 url 地址是否合法
- 2、浏览器先查看浏览器缓存-系统缓存-路由器缓存，如果缓存中有，会直接在屏幕中显示页面内容。若没有，则跳到第三步操作。
浏览器缓存: 浏览器会记录 DNS 一段时间, 因此, 只是第一个地方解析 DNS 请求;
操作系统缓存: 如果在浏览器缓存中不包含这个记录, 则会使系统调用操作系统, 获取操作系统的记录 (保存最近的 DNS 查询缓存);
路由器缓存: 如果上述两个步骤均不能成功获取 DNS 记录, 继续搜索路由器缓存;
ISP 缓存: 若上述均失败, 继续向 ISP 搜索。
- 3、在发送 http 请求前，需要域名解析 (DNS 解析)，解析获取相应的 IP 地址。
- 4、浏览器向服务器发起 tcp 连接，与浏览器建立 tcp 三次握手。
- 5、握手成功后，浏览器向服务器发送 http 请求，请求数据包。
- 6、服务器处理收到的请求，将数据返回至浏览器
- 7、浏览器收到 HTTP 响应
- 8、浏览器解码响应，如果响应可以缓存，则存入缓存。
- 9、浏览器发送请求获取嵌入在 HTML 中的资源 (html, css, javascript, 图片, 音乐)，对于未知类型，会弹出对话框。
- 10、浏览器发送异步请求。
- 11、页面全部渲染结束。

GET 和 POST 的区别

2. get 和 post 请求区别，这个是被问烂的题了

首先这个题看似简单，实际上是个送命题！如果你百度搜到的标准答案可能是这样的（本标准答案参考自 w3schools）：

- GET 在浏览器回退时是无害的，而 POST 会再次提交请求。
- GET 产生的 URL 地址可以被 Bookmark，而 POST 不可以。
- GET 请求会被浏览器主动 cache，而 POST 不会，除非手动设置。
- GET 请求只能进行 url 编码，而 POST 支持多种编码方式。
- GET 请求参数会被完整保留在浏览器历史记录里，而 POST 中的参数不会被保留。
- GET 请求在 URL 中传送的参数是有长度限制的，而 POST 么有。
- 对参数的数据类型，GET 只接受 ASCII 字符，而 POST 没有限制。
- GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息。
- GET 参数通过 URL 传递，POST 放在 Request body 中。

如果我告诉你，你死记硬背的这些所谓“标准答案”不是面试官想要的，你肯定不服，首先从安全性讲，get 和 post 都一样，没啥所谓的哪个更安全
get 请求参数在 url 地址上，直接暴露，post 请求的参数放 body 部分，按 F12 也直接暴露了，所以没啥安全性可言

“GET 参数通过 URL 传递，POST 放在 Request body 中”这个其实也不准，post 请求也可以没 body，也可以在 url 传递呢？

如果我告诉你 get 请求和 post 请求本质上没区别，你肯定不信！

GET 和 POST 有一个重大区别，简单的说：

GET 产生一个 TCP 数据包；POST 产生两个 TCP 数据包。

长的说：

对于 GET 方式的请求，浏览器会把 http header 和 data 一并发送出去，服务器响应 200（返回数据）；

而对于 POST，浏览器先发送 header，服务器响应 100 continue，浏览器再发送 data，服务器响应 200 ok（返回数据）。

详情可以参考这篇，写的挺好的 [《GET 和 POST 两种基本请求方法的区别》](#)

cookies 机制和 session 机制的区别

3. cookies 机制和 session 机制的区别，这个也是经常会问的

- cookies 数据保存在客户端，session 数据保存在服务器端；
- cookies 可以减轻服务器压力，但是不安全，容易进行 cookies 欺骗；
- session 较安全，但占用服务器资源

HTTP 状态码

4. HTTP 状态码 2xx, 3xx, 4xx, 5xx 分别是什么意思？这个是最基本的了，这个得熟练掌握，如果这个状态码都分不清，基本功就很弱了，印象分会大打折扣！

- 200 请求已成功，请求所希望的响应头或数据体将随此响应返回。
- 201 请求已经被实现，而且有一个新的资源已经依据请求的需要而建立，且其 URI 已经随 Location 头信息返回
- 202 服务器已接受请求，但尚未处理
- 301 （永久移动） 请求的网页已永久移动到新位置。 服务器返回此响应（对 GET 或 HEAD 请求的响应）时，会自动将请求者转到新位置。
- 302 （临时移动） 服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。
- 303 （查看其他位置） 请求者应当对不同的位置使用单独的 GET 请求来检索响应时，服务器返回此代码。
- 304 （未修改） 自从上次请求后，请求的网页未修改过。 服务器返回此响应时，不会返回网页内容。
- 305 （使用代理） 请求者只能使用代理访问请求的网页。 如果服务器返回此响应，还表示请求者应使用代理。
- 307 （临时重定向） 服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。
- 401 当前请求需要用户验证。如果当前请求已经包含了 Authorization 证书，那么 401 响应代表着服务器验证已经拒绝了那些证书
- 403 服务器已经理解请求，但是拒绝执行它。与 401 响应不同的是，身份验证并不能提供任何帮助，而且这个请求也不应该被重复提交
- 404 请求失败，请求所希望得到的资源未被在服务器上发现
- 500 服务器遇到了一个未曾预料的状态，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器的程序码出错时出现。
- 501 服务器不支持当前请求所需要的某个功能。当服务器无法识别请求的方法，并且无法支持其对任何资源的请求。
- 502 作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应。
- 503 由于临时的服务器维护或者过载，服务器当前无法处理请求。这个状况是临时的，并且将在一段时间以后恢复。

http 协议请求方式

5. http 协议有哪几种请求方式？

GET, POST 和 HEAD 方、OPTIONS, PUT, DELETE, TRACE 和 CONNECT 方法。

http 和 https 区别

6. http 和 https 区别?

HTTP 协议传输的数据都是未加密的，也就是明文的，因此使用 HTTP 协议传输隐私信息非常不安全，为了保证这些隐私数据能加密传输，于是网景公司 设计了 SSL (Secure Sockets Layer) 协议用于对 HTTP 协议传输的数据进行加密，从而就诞生了 HTTPS。简单来说，HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，要比 http 协议安全。

HTTPS 和 HTTP 的区别主要如下：

总的来说： HTTPS=SSL+HTTP

- 1、https 协议需要到 ca 申请证书，一般免费证书较少，因而需要一定费用。
- 2、http 是超文本传输协议，信息是明文传输，https 则是具有安全性的 ssl 加密传输协议。
- 3、http 和 https 使用的是完全不同的连接方式，用的端口也不一样，前者是 80，后者是 443。
(这个只是默认端口不一样，实际上端口是可以改的)
- 4、http 的连接很简单，是无状态的；HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，比 http 协议安全。

报文

7. HTTP 请求报文与响应报文格式

请求报文包含三部分：

- a、请求行：包含请求方法、URI、HTTP 版本信息
- b、请求头部 (headers) 字段
- c、请求内容实体 (body)

响应报文包含三部分：

- a、状态行：包含 HTTP 版本、状态码、状态码的原因短语
- b、响应头部 (headers) 字段
- c、响应内容 (body) 实体

post 请求 body

8. 常见的 POST 提交数据方式

application/x-www-form-urlencoded
multipart/form-data

application/json
text/xml

DNS

9. 什么是 DNS?

域名解析服务。将主机名转换为 IP 地址。如将 <http://www.cnblogs.com/> 主机名转换为 IP 地址：211.137.51.78

无状态

10. 什么是 Http 协议无状态协议?怎么解决 Http 协议无状态协议?

(1)、无状态协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息

(2)、无状态协议解决办法： 通过 1、Cookie 2、通过 Session 会话保存。

接口测试面试题

前言

接口测试最近几年被炒的火热了,越来越多的测试同行意识到接口测试的重要性。接口测试为什么会如此重要呢?

主要是平常的功能点点点,大家水平都一样,是个人都能点,面试时候如果问你平常在公司怎么测试的,你除了说点点点,还能说什么呢,无非就是这个项目点完了点那个项目,

这就是为什么各行各业的只要手指能点得动的人都来转行软件测试了。面试的时候面试官希望你除了点点点,还能更深入一点的思考页面上看不到的功能,也就是接口测试了。

为什么要做接口测试?

到底什么是接口测试,我们为什么要做接口测试?这是很多初入行的小伙伴的一个疑问,讲理论的你可能看不进去,接下来讲个实际案例,如下图一个提现功能

可提现金额：100 元

提现金额 元

提交

比如这个输入框，平常拿到这个 web 页面，会对输入框做用例设计：

- 输入一个负数（如：-100），点提交
- 输入金额为 0（如：0），点提交
- 输入金额为 0-100 的数（如：20），点提交
- 输入金额为 100（如：100），点提交
- 输入金额大于 100（如：108），点提交
- 输入 1 位小数（如：10.1），点提交
- 输入 2 位小数（如：10.12），点提交
- 输入 3 位小数（如：10.123），点提交

按照这个等价类，边界值用例测完，页面上不能输入负数和大于 3 位数小数点，然后就可以上线了。

然而。。。突然有一天数据库里面插入了一个提现金额为负数（-100），于是整个部门炸锅了，首先找到测试（背锅）去复现问题，测试在页面上反复输入负数，无法提交，认为没问题啊！

首先前端开发对输入框是做了限制的，前端 web 开发肯定没问题，这个锅前端开发 MM 不背。那么如果别人用户不通过你的 web 页面，直接发请求提交了呢？纳尼！！！不通过页面也能提交。。。这就是我们接下来要提到的接口测试了。

接口测试能发现哪些问题

面试题 1：你平常做接口测试的过程中发现过哪些 bug？

这个问题其实回到起来很简单，只要做过接口测试的，总能发现几个 BUG 吧，把你平常发现的 bug 说 2-3 个就可以了。

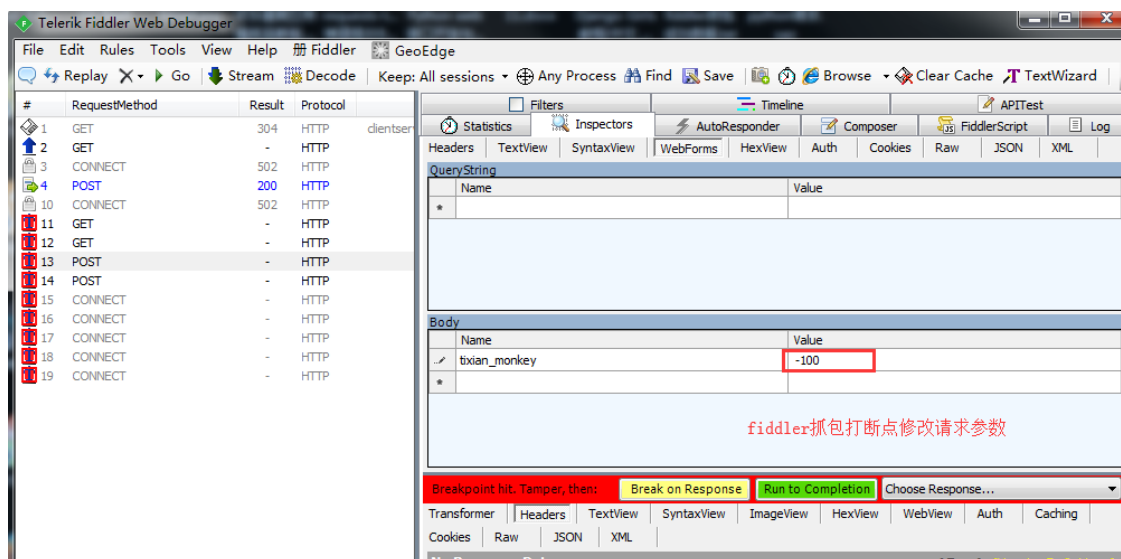
面试官出这个题，主要是想知道你是不是真的做过接口测试，毕竟现在很多小伙伴简历都是写的假的（你要不写估计面试机会都没有，没办法，为了生存，能理解）

比如上面说的，提现输入框，在页面上输入负数，肯定是无法提交过去（前端页面会判断金额），如果我不走前端，直接用接口工具发请求，输入一个负数过去。

（假设服务端没做提现金额数据判断）

余额=当前余额（100）-提现金额（-100），那么提现-100，余额就变成 200 了，也就是越提现，余额越大了

可以用接口工具去直接请求接口，也可以 fiddler 抓包，抓到接口后修改金额为负数



所以，接口测试的必要性就体现出来了：

1. 可以发现很多在页面上操作发现不了的 bug
2. 检查系统的异常处理能力
3. 检查系统的安全性、稳定性
4. 前端随便变，接口测好了，后端不用变
5. 可以测试并发情况，一个账号，同时（大于 2 个请求）对最后一个商品下单，或不同账号，对最后一个商品下单
6. 可以修改请求参数，突破前端页面输入限制（如金额）

接口测试怎么测

面试题 2：平常你是怎么测试接口的？

- 通过性验证：首先肯定要保证这个接口功能是好使的，也就是正常的通过性测试，按照接口文档上的参数，正常传入，是否可以返回正确的结果。
- 参数组合：现在有一个操作商品的接口，有个字段 type，传 1 的时候代表修改商品，商品 id、商品名称、价格有一个是必传的，type 传 2 的时候是删除商品，商品 id 是必传的，这样的，就要测参数组合了，type 传 1 的时候，只传商品名称能不能修改成功，id、名称、价格都传的时候能不能修改成功。
- 接口安全：
 - 1、绕过验证，比如说购买了一个商品，它的价格是 300 元，那我在提交

订单时候，我把这个商品的价格改成 3 元，后端有没有做验证，更狠点，我把钱改成-3，是不是我的余额还要增加？

2、绕过身份授权，比如说修改商品信息接口，那必须得是卖家才能修改，那我传一个普通用户，能不能修改成功，我传一个其他的卖家能不能修改成功

3、参数是否加密，比如说我登陆的接口，用户名和密码是不是加密，如果不加密的话，别人拦截到你的请求，就能获取到你的信息了，加密规则是否容易破解。

4、密码安全规则，密码的复杂程度校验

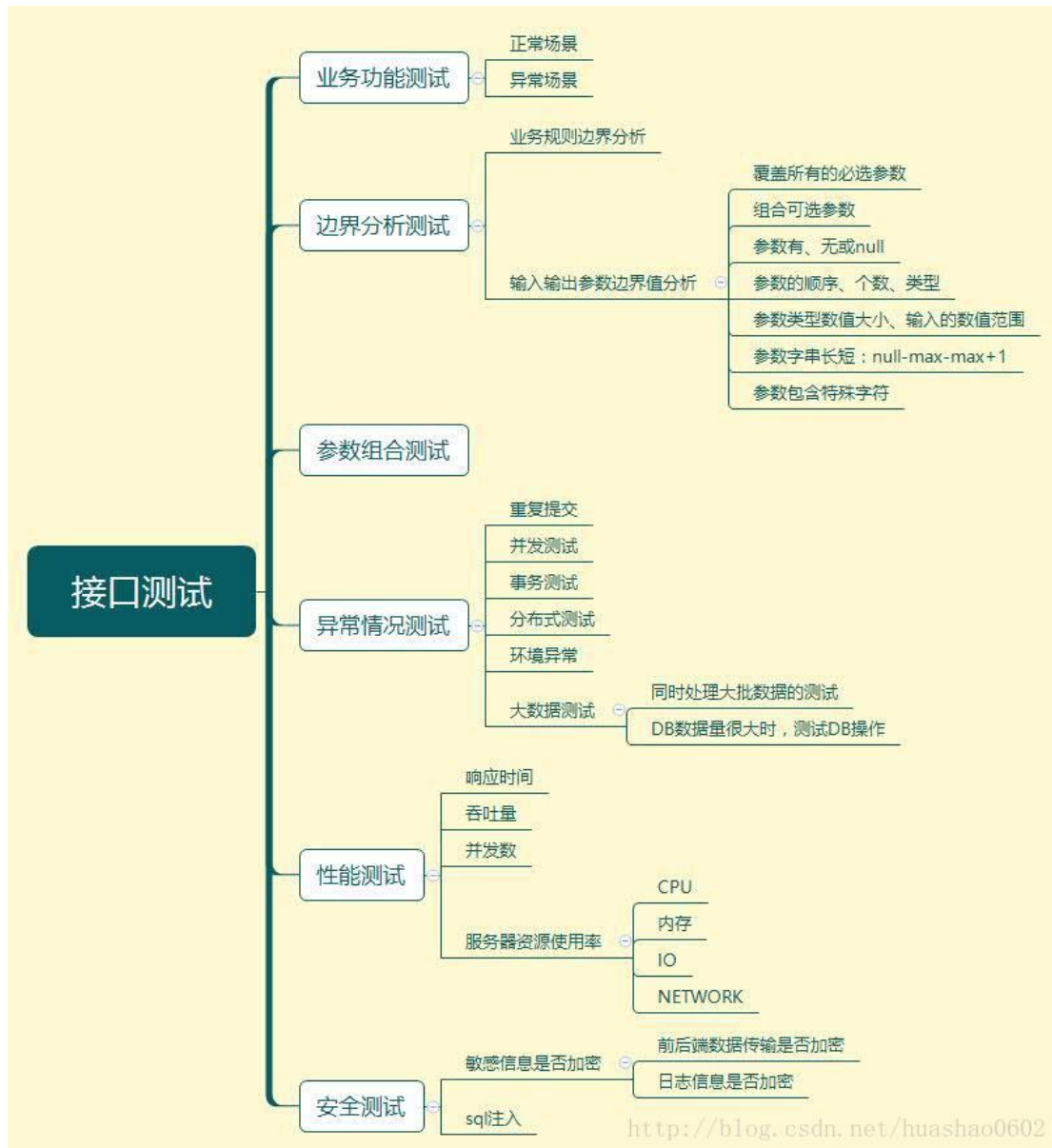
- 异常验证：

所谓异常验证，也就是我不按照你接口文档上的要求输入参数，来验证接口对异常情况的校验。比如说必填的参数不填，输入整数类型的，传入字符串类型，长度是 10 的，传 11，总之就是你说怎么来，我就不怎么来，其实也就这三种，必传非必传、参数类型、入参长度。

- 性能测试

接口并发情况，如上面提到的：一个账号，同时（大于 2 个请求）对最后一个商品下单，或不同账号，对最后一个商品下单

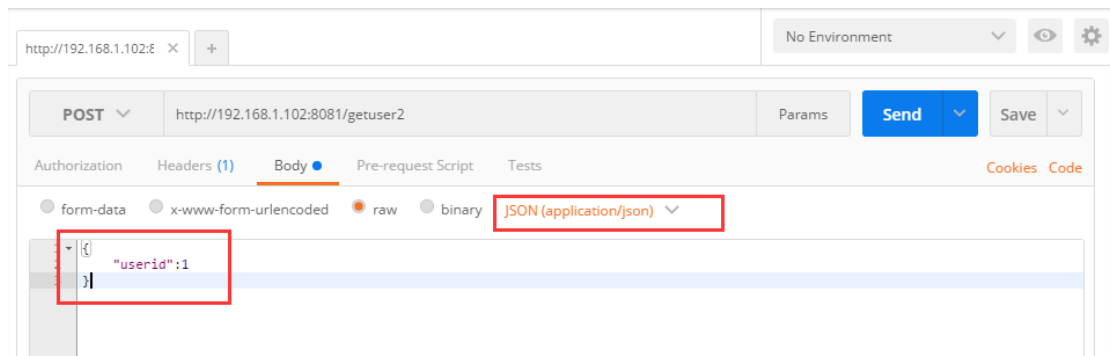
接口响应时间，响应时间太长了，肯定需要优化，一般都是毫秒级别



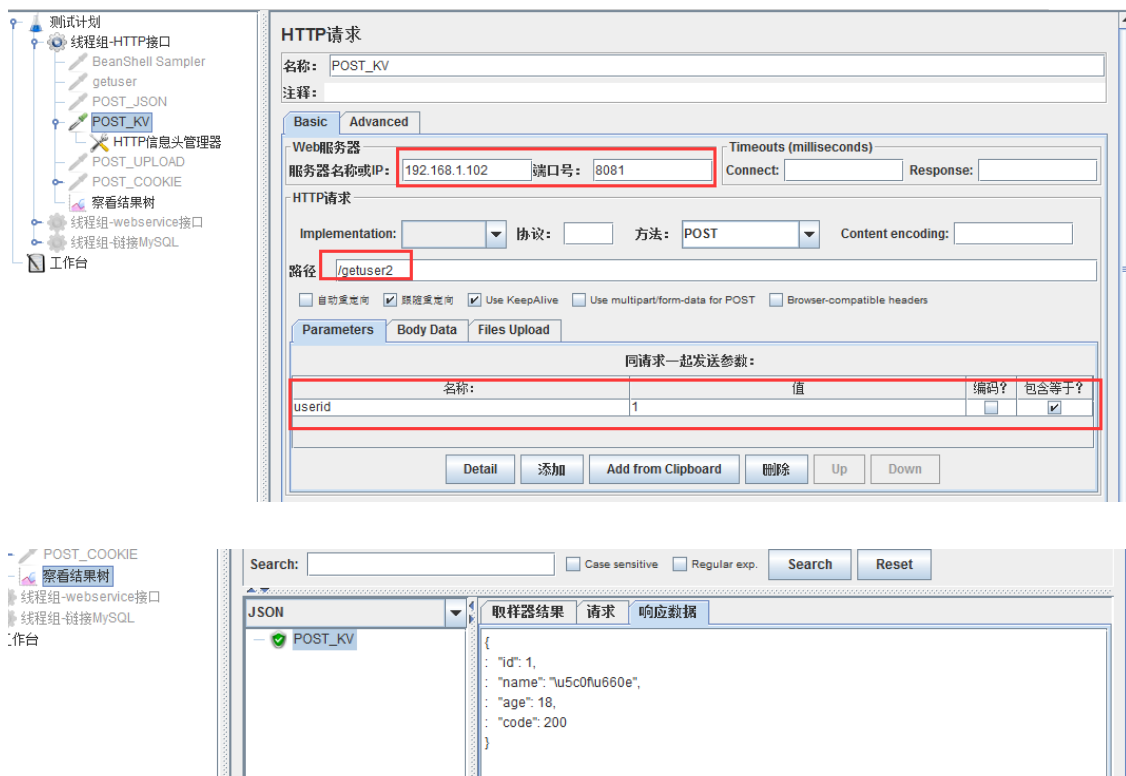
用什么工具测

面试题 3：平常用什么工具测接口的

接口测试工具很多，首先 postman



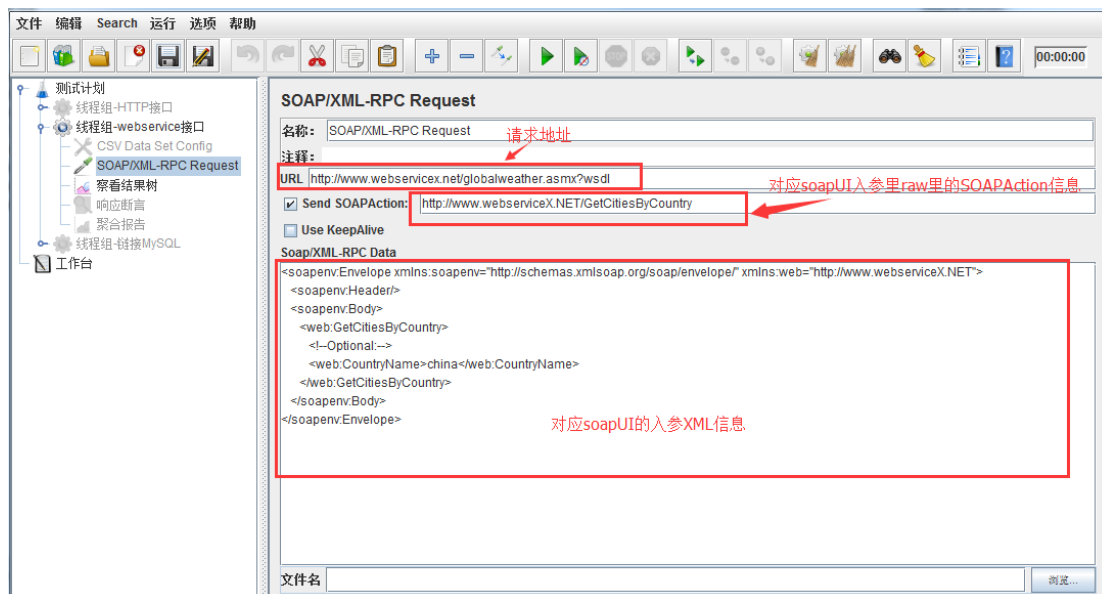
其次用 jmeter



webservice 接口

面试题 4: webservice 接口是如何测试的

webservice 接口用 SoapUI



没有接口文档如何做接口测试

面试题 5：没有接口文档，如何做接口测试？（这是个送命题）

没有接口文档，那还能咋办，瞎测呗！一个公司的开发流程里面，如果接口文档都没有，是无法展开接口测试的，你都不知道这个接口干什么的，也不知道具体每个字段代表什么意思，那还测啥呢？

——当然，你肯定不能回答面试官不测（心理 mmp，脸上笑嘻嘻），接下来就是扯犊子时间

1. 没有接口文档，那就需要先跟开发沟通，然后整理接口文档（本来是开发写的，没办法，为了唬住面试官，先说自己整理了）
2. 没有接口文档，可以抓包看接口请求参数，然后不懂的跟开发沟通

本题主要靠情商，通俗来说就是忽悠能力，先唬住面试官了再说，进去了也是瞎测测，随时做好背锅的准备

数据依赖

面试题 6：在手工接口测试或者自动化接口测试的过程中，上下游接口有数据依赖如何处理？

用一个全局变量来处理依赖的数据，比如登录后返回 token，其它接口都需要这个 token，那就用全局变量来传 token 参数

依赖第三方

面试题 7：依赖于第三方数据的接口如何进行测试？

这个标准答案是：mock

接着面试官会问你，如果 mock 的，然后你就顺着坑继续挖，搭建 mock 服务，参考这篇【<https://www.cnblogs.com/yoyoketang/p/9348552.html>】

抓包

面试题 8：当一个接口出现异常时候，你是如何分析异常的？

1. 抓包，用 fiddler 工具抓包，或者浏览器上 f12, app 上的话，那就用 fiddler 设置代理，去看请求报文和返回报文了
2. 查看后端日志，xshell 连上服务器，查看日志

弱网

面试题 9：如何模拟弱网测试

fiddler 和 charles 都可以模拟弱网测试，平常说的模拟丢包，也是模拟弱网测试

分析 bug 是前端还是后端的

面试题 10：如何分析一个 bug 是前端还是后端的？

平常提 bug 的时候，前端开发和后端开发总是扯皮，不承认是对方的 bug
这种情况很容易判断，先抓包看请求报文，对着接口文档，看请求报文有没有问题，有问题就是前端发的数据不对
请求报文没问题，那就看返回报文，返回的数据不对，那就是后端开发的问题咯

接口自动化面试题

前言

前面总结了一篇关于接口测试的常规面试题，现在接口自动化测试用的比较多，也是被很多公司看好。那么想做接口自动化测试需要具备哪些能力呢？
也就是面试的过程中，面试官会考哪些问题，知道你是不是真的做过接口自动化测试？总的来说问的比较多的就是以下几个问题：

1. json 和字典的区别？ - 对基础数据类型的考察
2. 测试的数据你放在哪？ - 数据与脚本分离
3. 参数化 - 数据驱动模式
4. 下个接口请求参数依赖上个接口的返回数据 - 参数关联
5. 依赖于登录的接口如何处理 - token 和 session 的管理
6. 依赖第三方的接口如何处理 - mock 模拟数据返回
7. 不可逆的操作，如何处理，比如删除一个订单这种接口如何测试 - 造数据
8. 接口产生的垃圾数据如何清理 - 数据清理
9. 一个订单的几种状态如何全部测到，如：未处理，处理中，处理失败，处理成功 - 造数据，改数据库订单状态
10. python 如何连接数据库操作？
11. 其它的就是运行出报告、代码管理（git）、运行策略和持续集成 jenkins 相关了

1. json 和字典 dict 的区别？

现在自动化培训烂大街，是个人都能说的上几个框架，面试如果问框架相关问题，求职者只需一瓶 82 年的雪碧，会吹的让你怀疑人生！

所以面试官为了更清楚的知道你是停留在表面上的花拳绣腿还是有扎实的基础，就不会问框架这种东西了。基本上问几个数据类型的基础就知道有没货了。

那么 json 和字典到底有什么区别呢？初学者连 python 的基础数据类型都没搞清楚，直接撸框架，有的人学了几个月可能都迷迷糊糊的，以为 json 就是字典。这个是肯定不对的。

首先 python 里面的基础数据类型有：int、str、float、list、bool、tuple、dict、set 这几种类型，里面没 json 这种数据类型。

JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMA Script (欧洲计算机协会制定的 js 规范) 的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。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。

由于你的代码是 python 写的（也有可能是 php, java, c, ruby 等语言），但是后端接口是 java 写的（也有可能是其它语言），不同的语言数据类型是不一样的（就好比中国的语言和美国的语言数据类型也不一样，中国的一般说一只羊，一头牛，美国都是 a /an 这种单位），所以就导致你提交的数据，别的开发语言无法识别，这就需要规范传输的数据（传输的数据都是一个字符串），大家都遵循一个规范，按一个标准的格式去传输，于是就有 json 这种国际化规范的数据类型。

json 本质上还是字符串，只是按 key: value 这种键值对的格式来的字符串


```
import json

# a 是字典 dict
a = {"a": 1, "b": 2, "c": True}

# b 是 json
b = '{"a": 1, "b": 2, "c": true}'

print(type(a))
print(json.dumps(a))    # a 转 json
```

运行结果

```
<class 'dict'>
{"a": 1, "b": 2, "c": true}
<class 'str'>
{'a': 1, 'b': 2, 'c': True}
```

2. 测试的数据你放在哪？

测试数据到底该怎么放，这个是面试官最喜欢问的一个题了，似乎仁者见仁智者见智，没有标准的答案，有的人说放 excel，也有的说放.py 脚本，也有的说放 ini 配置文件，还有放到 json，yaml 文件，txt 文件，甚至有的放数据库，五花八门，一百个做自动化的小伙伴有 100 个放的地方。

这里总结下测试的数据到底该怎么放？

首先测试的数据是分很多种的，有登录的账户数据，也有注册的账户数据，还有接口的参数，还有邮箱配置的数据等等等等，所以这个题不能一概而论给答死了。要不然就是给自己挖坑。

以下两个大忌不能回答：

- 测试的数据是不能写死到代码里面的，这个是原则问题，也是写代码的大忌（你要是回答写在代码里面，估计就是回去等通知了）
- 测试数据放到.py 的开头，这种其实很方便，对于少量的，固定不变的数据其实是可以放的，但是面试时候，千万不能这样说，面试官喜欢装逼的方法

测试数据存放总结：

1. 对于账号密码，这种管全局的参数，可以用命令行参数，单独抽出来，写的配置文件里（如 ini）

2. 对于一些一次性消耗的数据，比如注册，每次注册不一样的数，可以用随机函数生成
3. 对于一个接口有多组测试的参数，可以参数化，数据放 yaml, text, json, excel 都可以
4. 对于可以反复使用的数据，比如订单的各种状态需要造数据的情况，可以放到数据库，每次数据初始化，用完后再清理
5. 对于邮箱配置的一些参数，可以用 ini 配置文件
6. 对于全部是独立的接口项目，可以用数据驱动方式，用 excel/csv 管理测试的接口数据
7. 对于少量的静态数据，比如一个接口的测试数据，也就 2-3 组，可以写到 py 脚本的开头，十年八年都不会变更的

总之不同的测试数据，可以用不同的文件管理

3. 什么是数据驱动，如何参数化？

参数化和数据驱动的概念这个肯定要知道的，参数化的思想是代码用例写好了后，不需要改代码，只需维护测试数据就可以了，并且根据不同的测试数据生成多个用例

python 里面用 unittest 框架

```
import unittest
import ddt

# 测试数据
datas = [ {"user": "admin", "psw": "123", "result": "true"},
          {"user": "admin1", "psw": "1234", "result": "true"},
          {"user": "admin2", "psw": "1234", "result": "true"},
          {"user": "admin3", "psw": "1234", "result": "true"},
          {"user": "admin4", "psw": "1234", "result": "true"},
          {"user": "admin5", "psw": "1234", "result": "true"},
          {"user": "admin6", "psw": "1234", "result": "true"},
          {"user": "admin7", "psw": "1234", "result": "true"},
          {"user": "admin8", "psw": "1234", "result": "true"},
          {"user": "admin9", "psw": "1234", "result": "true"},
          {"user": "admin10", "psw": "1234", "result": "true"},
          {"user": "admin11", "psw": "1234", "result": "true"}]

@ddt.ddt
class Test(unittest.TestCase):

    @ddt.data(*datas)
    def test_(self, d):
        """上海-悠悠: {0}"""
```

```
print("测试数据: %s" % d)
```

```
if __name__ == "__main__":  
    unittest.main()
```

unittest 框架还有一个 paramunittest 也可以实现

```
import unittest  
import paramunittest  
import time  
# python3.6  
# 作者: 上海-悠悠
```

```
@paramunittest.parametrize(  
    {"user": "admin", "psw": "123", "result": "true"},  
    {"user": "admin1", "psw": "1234", "result": "true"},  
    {"user": "admin2", "psw": "1234", "result": "true"},  
    {"user": "admin3", "psw": "1234", "result": "true"},  
    {"user": "admin4", "psw": "1234", "result": "true"},  
    {"user": "admin5", "psw": "1234", "result": "true"},  
    {"user": "admin6", "psw": "1234", "result": "true"},  
    {"user": "admin7", "psw": "1234", "result": "true"},  
    {"user": "admin8", "psw": "1234", "result": "true"},  
    {"user": "admin9", "psw": "1234", "result": "true"},  
    {"user": "admin10", "psw": "1234", "result": "true"},  
    {"user": "admin11", "psw": "1234", "result": "true"},  
)
```

```
class TestDemo(unittest.TestCase):  
    def setParameters(self, user, psw, result):  
        '''这里注意了, user, psw, result 三个参数和前面定义的字典一一  
对应'''  
        self.user = user  
        self.psw = psw  
        self.result = result  
  
    def testcase(self):  
        print("开始执行用例: -----")  
        time.sleep(0.5)  
        print("输入用户名: %s" % self.user)  
        print("输入密码: %s" % self.psw)  
        print("期望结果: %s" % self.result)  
        time.sleep(0.5)  
        self.assertTrue(self.result == "true")
```

```
if __name__ == "__main__":
    unittest.main(verbosity=2)
```

如果用的是 pytest 框架，也能实现参数化

```
# content of test_canshu1.py

# coding:utf-8

import pytest
@pytest.mark.parametrize("test_input, expected",
                          [ ("3+5", 8),
                            ("2+4", 6),
                            ("6 * 9", 42),
                          ])
def test_eval(test_input, expected):
    assert eval(test_input) == expected

if __name__ == "__main__":
    pytest.main(["-s", "test_canshu1.py"])
```

pytest 里面还有一个更加强大的功能，获得多个参数化参数的所有组合，可以堆叠参数化装饰器

```
import pytest
@pytest.mark.parametrize("x", [0, 1])
@pytest.mark.parametrize("y", [2, 3])
def test_foo(x, y):
    print("测试数据组合: x->%s, y->%s" % (x, y))

if __name__ == "__main__":
    pytest.main(["-s", "test_canshu1.py"])
```

4. 下个接口请求参数依赖上个接口的返回数据

这个很容易，不同的接口封装成不同的函数或方法，需要的数据 return 出来，用一个中间变量 a 去接受，后面的接口传 a 就可以了

参考这篇[【python 接口自动化 26-参数关联和 JSESSIONID\(上个接口返回数据作为下个接口请求参数\)】](#)

5. 依赖于登录的接口如何处理

登录接口依赖 token 的，可以先登录后，token 存到一个 yaml 或者 json，或者 ini 的配置文件里面，后面所有的请求去拿这个数据就可以全局使用了

参考之前分享的一篇 [python 接口自动化 24-有 token 的接口项目使用 unittest 框架设计](#)

如果是 cookies 的参数，可以用 session 自动关联

```
s=requests.session()
```

后面请求用 s.get() 和 s.post() 就可以自动关联 cookies 了

6. 依赖第三方的接口如何处理

这个需要自己去搭建一个 mock 服务，模拟接口返回数据，参考【python 笔记 25-mock-server 之 moco】

(<https://www.cnblogs.com/yoyoketang/p/9348552.html>)

moco 是一个开源的框架，在 github 上可以下载到

<https://github.com/dreamhead/moco>

moco 服务搭建需要自己能够熟练掌握，面试会问你具体如何搭建，如何模拟返回的数据，是用的什么格式，如何请求的

7. 不可逆的操作，如何处理，比如删除一个订单这种接口如何测试

此题考的是造数据的能力，接口的请求数据，很多都是需要依赖前面一个状态的比如 workflow 这种，流向不同的人状态不一样，操作权限不一样，测试的时候，每种状态都要测到，就需要自己会造数据了。

平常手工测试造数据，直接在数据库改字段状态。那么自动化也是一样，造数据可以用 python 连数据库了，做增删改查的操作

测试用例前置操作，setUp 做数据准备

后置操作，tearDown 做数据清理

8. 接口产生的垃圾数据如何清理

跟上面一样，造数据和数据清理，需用 python 连数据库了，做增删改查的操作测试用例前置操作，setUp 做数据准备

后置操作，tearDown 做数据清理

9. 一个订单的几种状态如何全部测到？

如：未处理，处理中，处理失败，处理成功

跟上面一样，也是考察造数据，修改数据的状态

10. python 如何连接数据库操作？

这个就是详细的考察你是如何用 python 连数据库的，并且最好能现场写代码那种（有的笔试题就是 python 连数据库）

具体问你用到哪个模块，查询的数据是什么类型？如何删除数据？如何新增数据？如何修改数据？

PyMySQL 是在 Python3.x 版本中用于连接 MySQL 服务器的一个库，Python2 中则使用 mysqlldb。

详情参考教程 <http://www.runoob.com/python3/python3-mysql.html>

```
#!/usr/bin/python3
# 查询 EMPLOYEE 表中 salary（工资）字段大于 1000 的所有数据：
import pymysql

# 打开数据库连接
db = pymysql.connect("localhost","testuser","test123","TESTDB" )

# 使用 cursor() 方法获取操作游标
cursor = db.cursor()

# SQL 查询语句
sql = "SELECT * FROM EMPLOYEE \
       WHERE INCOME > %s" % (1000)
try:
    # 执行 SQL 语句
    cursor.execute(sql)
    # 获取所有记录列表
    results = cursor.fetchall()
    for row in results:
        fname = row[0]
        lname = row[1]
        age = row[2]
        sex = row[3]
        income = row[4]
        # 打印结果
        print ("fname=%s,lname=%s, age=%s, sex=%s, income=%s" % \
              (fname, lname, age, sex, income ))
```

```
except:
    print ("Error: unable to fetch data")

# 关闭数据库连接
db.close()
```

其它的就是运行出报告、代码管理（git）、运行策略和持续集成 jenkins 相关了，这个所以的自动化但是一样的，后面会单独讲一篇 jenkins 持续集成相关

Mock 相关-单元测试

说到接口测试，必问 mock, mock 通俗一点来说就是模拟接口返回。解决接口的依赖关系，主要是为了解耦，单元测试用的多

什么是 mock

unittest.mock 是一个用于在 Python 中进行单元测试的库，Mock 翻译过来就是模拟的意思，顾名思义这个库的主要功能是模拟一些东西。

它的主要功能是使用 mock 对象替代掉指定的 Python 对象，以达到模拟对象的行为。

学过 python 自动化的对 unittest 并不陌生，unittest 其实是单元测试框架，但对于单元测试，估计很多小伙伴都不懂，单元测试才是自动化测试的至高境界，其中 mock 是单元测试的精髓所在

mock 能做什么

- 1. 前后端联调，如果你是一个前端页面开发，现在需要开发一个功能：下一个订单，支付页面的接口，根据支付结果，支付成功，展示支付成功页，支付失败，展示支付失败页。
要完成此功能，你需要调用后端的接口，根据返回给你的结果，来展示不同的页面。此时后端接口还没开发好，
作为一个前端开发总不能等别人开发好了，你再开发，那你只有加班的命了。
为了同步开发完成任务，此时，你可以根据接口文档的规定，把接口的地址和入参传过去，然后自己 mock 接口的不同返回界面，来完成前端的开发任务
- 2. 单元测试，单元测试的目的是测试某个小小单元的功能，但现实中开发的函数或方法都是有依赖关系的，比如 b 函数的参数，需要调用 a 函数的返回结果，但是我前面已经测试 a 函数了

这种情况下，就不需要再测一次 a 函数了，此时就可以用 mock 模块来模拟调用这部分内容，并给出返回结果

- 3. 第三方接口依赖，在做接口自动化的时候，有时候需要调用第三方的接口，但是别人公司的接口服务不受你的控制，有可能别人提供的测试环境今天服务给你开着，别人就关掉了，给自动化接口测试带来很多的麻烦，此时就可以自己写一个 mock-server 来模拟接口的返回数据

mock 环境准备

1. python2.x 的版本，mock 是一个独立的模块，需要用 pip 安装

```
pip install -U mock
```

3. 从 Python 3.3 以后的版本 mock 已经合并到 unittest 模块中了，是 unittest 单元测试的一部分，直接导入过来就行

```
from unittest import mock
```

依赖关系

1. 如下场景：支付是一个独立的接口，由其它开发提供，根据支付的接口返回状态去显示失败，还是成功，这个是你需要实现的功能

也就是说你写一个 b 功能，你的同事写一个 a 功能，你的 b 功能需要根据 a 功能的结果去判断，然后实现对应的功能。这就是存在依赖关系，你同事开发的进度你是无法控制的

你要是等他开发完了，你再开发，那你就坐等加班吧。

2. 以下是自己写的 zhifu_statues() 函数功能，大概设计如下，保存为 temple.py 文件

```
# 保存为 temple.py
```

```
# coding:utf-8
```

```
# 作者：上海-悠悠 QQ 交流群：588402570
```

```
def zhifu():  
    ''' 假设这里是一个支付的功能, 未开发完  
    支付成功返回: {"result": "success", "reason": "null"}  
    支付失败返回: {"result": "fail", "reason": "余额不足"}  
    reason 返回失败原因  
    '''
```



```
pass

def zhifu_statues():
    '''根据支付的结果 success or fail, 判断跳转到对应页面'''
    result = zhifu()
    print(result)
    try:
        if result["result"] == "success":
            return "支付成功"
        elif result["result"] == "fail":
            print("失败原因: %s" % result["reason"])
            return "支付失败"
        else:
            return "未知错误异常"
    except:
        return "Error, 服务端返回异常!"
```

3. 单元测试用例设计

```
# coding:utf-8
from unittest import mock
import unittest
import temple
# 作者: 上海-悠悠 QQ 交流群: 588402570

class Test_zhifu_statues(unittest.TestCase):
    '''单元测试用例'''
    def test_01(self):
        '''测试支付成功场景'''
        # mock 一个支付成功的数据
        temple.zhifu = mock.Mock(return_value={"result": "success",
"reason": "null"})
        # 根据支付结果测试页面跳转
        statues = temple.zhifu_statues()
        print(statues)
        self.assertEqual(statues, "支付成功")

    def test_02(self):
        '''测试支付失败场景'''
        # mock 一个支付成功的数据
        temple.zhifu = mock.Mock(return_value={"result": "fail",
"reason": "余额不足"})
        # 根据支付结果测试页面跳转
        statues = temple.zhifu_statues()
```

```
self.assertEqual(statues, "支付失败")
```

```
if __name__ == "__main__":  
    unittest.main()
```

4. 单元测试是为了保证函数的每个分支都测到，以上 zhifu_statues() 函数有四个分支，也就是说要写四个案例，这里我只写了 2 个，还有 2 个案例，你们可以自己练习下

mock-server 环境搭建—加分项！

前言

mock 除了用在单元测试过程中，还有一个用途，当前端开发在开发页面的时候，需要服务端提供 API 接口

此时服务端没开发完成，或者说没搭建测试环境，这个时候前端开发会自己 mock 一个 api 服务端，自己给自己提供调用接口的返回数据

mock-server 用途就是开发在开发的过程中，需要依赖一部分的接口，但是对方没有提供或者环境等情况

环境准备

- 1.moco-runner-0.12.0-standalone.jar
- 2.java 版本 1.8.0_20

Moco 环境搭建

1. moco 是一个开源的框架，在 github 上可以下载到，[github 源码地址](#) 目前最新版是 moco-runner-0.12.0-standalone.jar，[【点这里下载】](#)

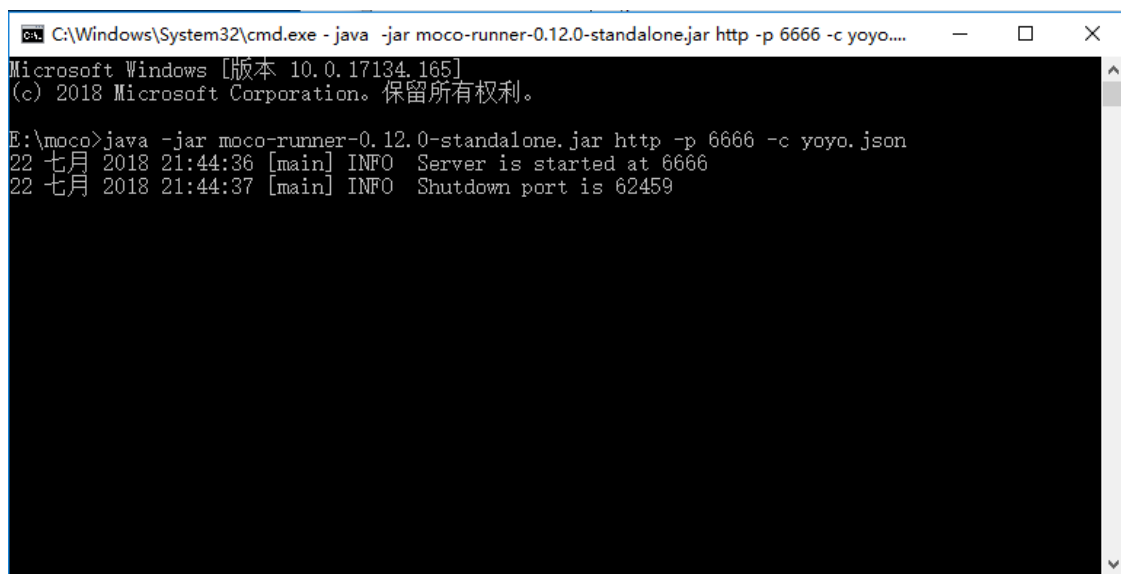
2. 下载完成后，保存到电脑本地的一个目录，然后在同一目录下新建一个 yoyo.json 文件，里面写入如下内容：



```
[
  {
    "response" :
    {
      "text" : "Hello World! yoyoketang"
    }
  }
]
```

3. 接下来启动 moco 服务，打开 moco-runner-0.12.0-standalone.jar 所在的文件夹目录 (E:\moco)，地址栏左上角输入 cmd，回车打开 cmd 窗口

```
java -jar moco-runner-0.12.0-standalone.jar http -p 6666 -c yoyo.json
```

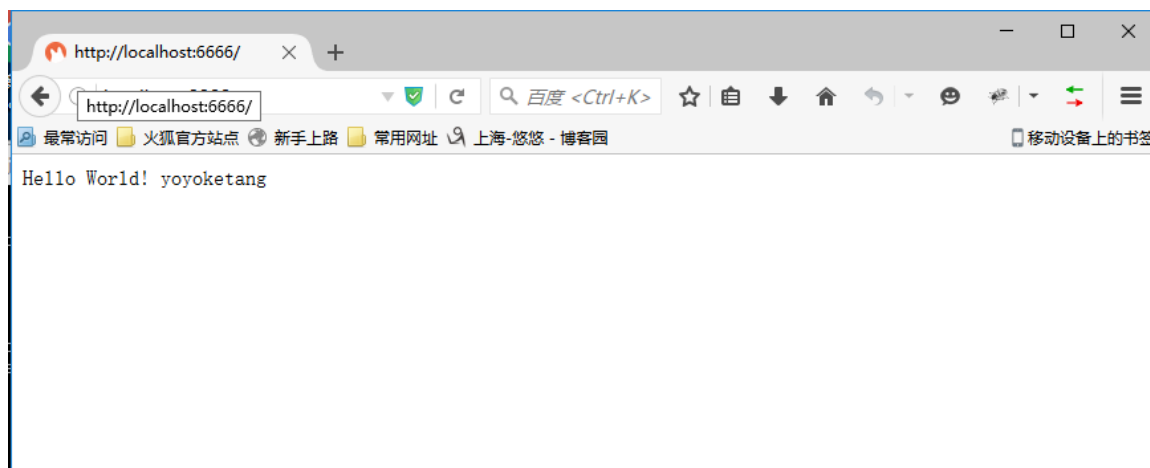


出现如下结果就是启动成功了

```
E:\moco>java -jar moco-runner-0.12.0-standalone.jar http -p 6666 -c yoyo.json
22 七月 2018 21:44:36 [main] INFO Server is started at 6666
22 七月 2018 21:44:37 [main] INFO Shutdown port is 62459
```

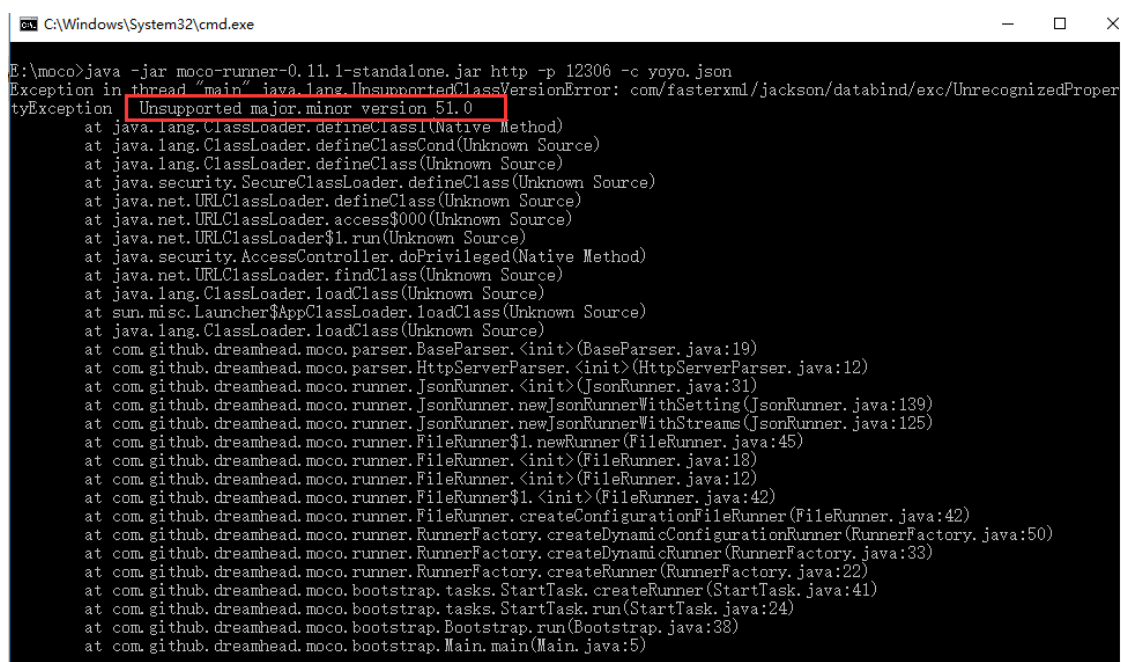
4. 打开火狐浏览器(谷歌浏览器貌似打不开)地址栏输入:

<http://localhost:6666/> 看到如下界面, 说明 mock-server 环境搭建成功



遇到的问题

1. 刚开始 jdk 版本是 1.6 的, 启动一直报错, 报错信息如下:Unsupported major.minor version 51.0



2. 百度搜了下, 找到问题原因了, jdk 版本和 jar 包不匹配导致。比如本机的 JDK 为 1.6, 但是项目编译时用的 JDK 为 1.7 那么就会出现这个异常, 因为本机 JDK 版本较低不能执行编译版本为高版本的 Class 文件, 各 JDK 版本对应的错误编号如下:

J2SE 8 = 52,

J2SE 7 = 51,

J2SE 6.0 = 50,

J2SE 5.0 = 49,

JDK 1.4 = 48,

JDK 1.3 = 47,

JDK 1.2 = 46,

JDK 1.1 = 45

官方文档

1. 更多参考资料, 查阅官方文档, [【点这里查看官方文档】](#)

GitHub, Inc. [US] | <https://github.com/dreamhead/moco>

Quick Start

- Download [Standalone Moco Runner](#)
- Write your own configuration file to describe your Moco server configuration as follow:

```
[
  {
    "response" :
    {
      "text" : "Hello, Moco"
    }
  }
]
```

(foo.json)

- Run Moco HTTP server with the configuration file.

```
java -jar moco-runner-<version>-standalone.jar http -p 12306 -c foo.json
```

- Now, open your favorite browser to visit <http://localhost:12306> and you will see "Hello, Moco".

** 作者: 上海-悠悠 QQ 交流群: 779429633**

五、selenium 篇

个人博客地址: <http://www.cnblogs.com/yoyoketang/>

QQ 交流群: 779429633

联系本人 qq: 283340479

selenium 中隐藏元素如何定位?

前言

面试题: selenium 中隐藏元素如何定位? 这个是很多面试官喜欢问的一个题, 如果单纯的定位的话, 隐藏元素和普通不隐藏元素定位没啥区别, 用正常定位方法就行了

但是吧~~~很多面试官自己都搞不清楚啥叫定位, 啥叫操作元素 (如 click, clear, send_keys)

隐藏元素

如下图有个输入框和一个登录的按钮, 本来是显示的



元素的属性隐藏和显示，主要是 `type="hidden"` 和 `style="display: none;"` 属性来控制的，接下来在元素属性里面让它隐藏

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<p>这里有个按钮，是隐藏的

<!-- type="hidden" -->
<br>
输入账号<input id="yoyo" name="hello" type="hidden">

<!-- type="display: none;" -->
<br>
<button id="yy" name="heo" style="display: none;">登录</button>
<br>

<a hidden id="baidu" href="https://www.baidu.com">访问百度</a>
</p>

</body>
</html>
```

这样元素就不会显示了，也就是面试官所说的隐藏属性了

这里有个按钮，是隐藏的
输入账号

定位隐藏元素

前面说了，定位隐藏元素和普通的元素没啥区别，接下来就来验证下，是不是能定位到呢？

```
from selenium import webdriver

driver = webdriver.Firefox()
driver.get("http://localhost:63342/test1122/a/b.html")

# 定位 type="hidden" 隐藏元素
ele1 = driver.find_element_by_id("yoyo")
print("打印元素信息: %s" % ele1)

# 获取元素属性
print(ele1.get_attribute("name"))

# 判断元素是否隐藏
print(ele1.is_displayed())
```

运行结果：

```
打印元素信息: <selenium.webdriver.remote.webelement.WebElement
(session="1debdd46-21b1-451e-b8a7-5aeffd74f9d",
element="{28628a87-7f22-4574-9e14-931f9c6f20e1}")>
hello
False
```

运行结果可以看出，隐藏元素用普通定位方法，事实上是定位到了呢！

操作隐藏元素

隐藏元素可以正常定位到，只是不能操作（定位元素和操作元素是两码事，很多初学者傻傻分不清楚），操作元素是 click, clear, send_keys 这些方法

```
# 隐藏输入框元素输入文本
ele1 = driver.find_element_by_id("yoyo")
ele1.send_keys("yoyo")
```

隐藏元素用 send_keys() 方法会抛异常 'ElementNotVisibleException':
Message: Element is not currently visible and so may not be interacted with

这个报错是说元素不可见，不可以被操作，同样的对“登录”按钮点击操作也是会报 'ElementNotVisibleException'

```
# 点击隐藏登录框
ele2 = driver.find_element_by_id("yy")
ele2.click()
```

JS 操作隐藏元素

如果面试官想问的是定位后操作隐藏元素的话，本质上说这个问题就是毫无意义的，web 自动化的目的是模拟人的正常行为去操作。

如果一个元素页面上都看不到了，你人工也是无法操作的是不是？人工都不能操作，那你自动化的意义又在哪呢？所以这个只是为了单纯的考察面试者处理问题的能力，没啥实用性！（面试造飞机，进去拧螺丝）

既然面试官这么问了，那就想办法回答上给个好印象吧！

首先 selenium 是无法操作隐藏元素的（但是能正常定位到），本身这个框架就是设计如此，如果非要去操作隐藏元素，那就用 js 的方法去操作，selenium 提供了一个入口可以执行 js 脚本。

js 和 selenium 不同，只有页面上有的元素（在 dom 里面的），都能正常的操作，接下来用 js 试试吧！

<a hidden id="baidu" href="https://www.baidu.com">访问百度 这个链接是隐藏的，但是能用 js 点到

```
from selenium import webdriver

driver = webdriver.Firefox()
driver.get("http://localhost:63342/test1122/a/b.html")

# js 点击 hidden 元素

js = 'document.getElementById("baidu").click()'
driver.execute_script(js)
```

运行完之后，会发现页面正常的点击，跳转到百度页面了

备注：百度搜到的可能方法是先用 js 去掉 hidden 属性，再用 selenium 操作，这个有点多此一举，你既然都已经会用 js 了，何必不一次性到位直接 click 呢？

selenium 面试题

前言

面试 web 自动化必然会问到 selenium，问 selenium 相关的问题定位是最基本的，也是自动化的根本，所以面试离不开元素定位问题。
之前看到招聘要求里面说“只会复制粘贴 xpath 的就不要投简历了”，说明面试官对求职者的自动化能力要求不能停留在复制粘贴上。
还是那句话，想学自动化的话，需牢记：录制穷三代，复制毁一生！

1. 如何判断一个页面上元素是否存在？

这个可以说是被问烂的题了，判断元素存在方法有三种：

方法一，用 try...except...

```
def is_element_exsist(driver, locator):  
    """  
    判断元素是否存在, 存在返回 True, 不存返回 False  
    :param locator: locator 为元组类型, 如("id", "yoyo")  
    :return: bool 值, True or False  
    """  
    try:  
        driver.find_element(*locator)  
        return True  
    except Exception as msg:  
        print("元素%s 找不到: %s" % (locator, msg))  
        return False  
  
if __name__ == '__main__':  
    loc1 = ("id", "yoyo") # 元素 1  
    print(is_element_exsist(driver, loc1))
```

方法二：用 elements 定义一组元素方法

```
def is_element_exsist1(driver, locator):  
    """  
    判断元素是否存在, 存在返回 True, 不存返回 False  
    :param locator: locator 为元组类型, 如("id", "yoyo")  
    :return: bool 值, True or False  
    """  
  
    eles = driver.find_elements(*locator)  
    if len(eles) < 1:  
        return False  
    else:  
        return True  
  
if __name__ == '__main__':  
    loc1 = ("id", "yoyo") # 元素 1  
    print(is_element_exsist1(driver, loc1))
```

(强烈推荐!)方法三: 结合 WebDriverWait 和 expected_conditions 判断

```
from selenium import webdriver  
from selenium.webdriver.support import expected_conditions as EC  
from selenium.webdriver.support.ui import WebDriverWait  
def is_element_exsist2(driver, locator):  
    """  
    结合 WebDriverWait 和 expected_conditions 判断元素是否存在,  
    每间隔 1 秒判断一次, 30s 超时, 存在返回 True, 不存返回 False  
    :param locator: locator 为元组类型, 如("id", "yoyo")  
    :return: bool 值, True or False  
    """  
  
    try:  
        WebDriverWait(driver, 30,  
1).until(EC.presence_of_element_located(locator))  
        return True  
    except:  
        return False  
if __name__ == '__main__':  
    loc1 = ("id", "yoyo") # 元素 1  
    print(is_element_exsist2(driver, loc1))
```

2. 如何提高脚本的稳定性

相关类似问题还有“用例在运行过程中经常会出现不稳定的情况, 也就是说这次可以通过, 下次就没办法通过了, 如何去提升用例的稳定性?”

“如何提高 selenium 脚本的执行速度?”

“selenium 中如何保证操作元素的成功率? 也就是说不管网络加载慢还是快”

如果一个元素今天你能定位到，过两天就定位不到了，只要这个页面没变过，说明定位方法是没啥问题的。

优化方向：1. 不要右键复制 xpath(十万八千里那种路径，肯定不稳定)，自己写相对路径，多用 id 为节点查找

2. 定位没问题，第二个影响因素那就是等待了，sleep 等待尽量少用（影响执行时间）

driver.implicitly_wait(30)这个等待也不要，不要以为是全局的就是好事，有些 js 加载失败时候会一直等，并且页面跳转时候也无法识别

3. 定位元素方法重新封装，结合 WebDriverWait 和 expected_conditions 判断元素方法，自己封装一套定位元素方法

```
from selenium import webdriver
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.ui import WebDriverWait

def find(locator, timeout=30):
    '''定位元素，参数 locator 是元祖类型，如("id", "yoyo")'''
    element = WebDriverWait(driver, timeout,
1).until(EC.presence_of_element_located(locator))
    return element
```

3. 如何定位动态元素

动态元素有 2 种情况，一个是属性动态，比如 id 是动态的，定位时候，那就不要用 id 定位就是了

```
<p id="yo" class="hello world">
  <button id="yy_auto_1929292" name="heo" >登录</button>
  <br>
</p>
```

比如上面这个 button 元素，id 是动态的，定位方法千千万，何必死在 id 上，可以用 name 定位，

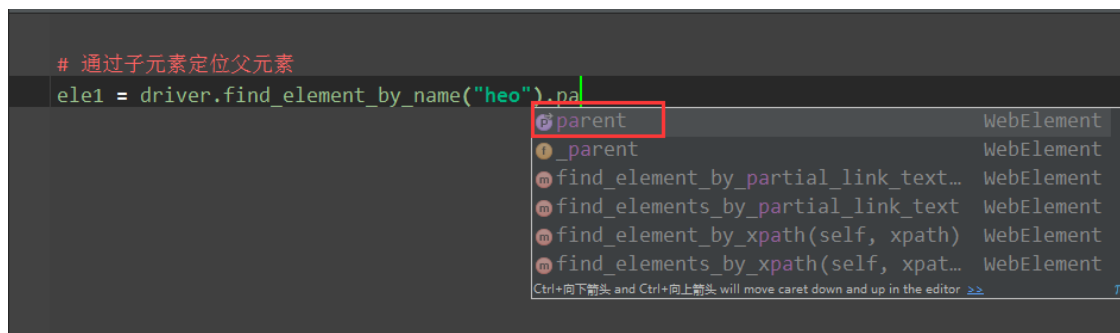
哪怕这个元素属性都是动态的，它的标签不可能动态吧，那就定位父元素 id="yo"啊：.//*[@id='yo']/button

还有一种情况动态的，那就是这个元素一会在页面上方，一会在下方，飘忽不定的动态元素，定位方法也是一样，按 f12，根据元素属性定位（元素的 tag、name 的步伐属性是不会变的，动的只是 class 属性和 styles 属性）

4. 如何通过子元素定位父元素

面试官尽喜欢搞一些冷门的定位来考求职者，当初我也被这个问题送了小命。回来后专门查了相关资料，找到了这个定位方法

selenium 里面通过父元素，定位子元素，可以通过二次定位来找到该元素:ele1 = driver.find_element_by_id("yoyo").find_element_by_id("ziyuans")
但是通过子元素找父元素这种思维之前真没注意过，实际上 selenium 里面提供了该方法



```
<p id="yo" class="hello world">
  <button id="yy_auto" name="heo" >登录</button>
  <br>
</p>
```



虽然用 parent 方法定位到了父元素，但是无法获取元素属性，也不能操作，没搞懂有啥意义

另外一个思路，子元素定位父元素，可以通过 xpath 的语法直接定位: .//*[@name="heo"]/.. 两个点.. 就是代表父级元素了

5. 如果截取某一个元素的图片，不要截取全部图片

可以参考之前写过的这篇：

<https://www.cnblogs.com/yoyoketang/p/7748693.html>

```
# coding:utf-8
from selenium import webdriver
from PIL import Image
driver = webdriver.Chrome()
driver.get('http://www.baidu.com/')

driver.save_screenshot('button.png')
element = driver.find_element_by_id("su")
print(element.location)           # 打印元素坐标
print(element.size)              # 打印元素大小

left = element.location['x']
top = element.location['y']
right = element.location['x'] + element.size['width']
bottom = element.location['y'] + element.size['height']

im = Image.open('button.png')
im = im.crop((left, top, right, bottom))
im.save('button.png')
```

6. 平常遇到过哪些问题?如何解决的

可以把平常遇到的元素定位的一些坑说下，然后说下为什么没定位到，比如动态id、有iframe、没加等待等因素

如何解决的一百度：上海-悠悠，上面都有解决办法

7. 一个元素明明定位到了，点击无效（也没报错），如果解决？

使用 js 点击，selenium 有时候点击元素是会失效

js 点击

```
js = 'document.getElementById("baidu").click()'
driver.execute_script(js)
```

六、app 篇

个人博客地址: <http://www.cnblogs.com/yoyoketang/>

QQ 交流群: 779429633

联系本人 qq: 283340479

app 测试面试题

前言

现在面试个测试岗位, 都是要求全能的, web、接口、app 啥都要会测, 那么 APP 测试一般需要哪些技能呢?

面试 app 测试岗位会被问到哪些问题, 怎样让面试官觉得你对 APP 测试很精通的样子?

本篇总结了 app 测试面试时候经常被问的 10 个相关问题

1. 什么是 activity?
2. Activity 生命周期?
3. Android 四大组件?
4. app 测试和 web 测试有什么区别?
5. android 和 ios 测试区别?
6. app 出现 ANR, 是什么原因导致的?
7. App 出现 crash 原因有哪些?
8. app 对于不稳定偶然出现 anr 和 crash 时候你是怎么处理的?
9. app 的日志如何抓取?
10. 你平常会看日志吗, 一般会出现哪些异常 (Exception) ?

1. 什么是 activity

什么是 activity, 这个前两年出去面试 APP 测试岗位, 估计问的最多了, 特别是一些大厂, 先问你是不是做过 APP 测试, 那好, 你说说什么是 activity?

如果没看过 android 的开发原理, 估计这个很难回答, 要是第一个问题就被难住了, 面试的信心也会失去一半了, 士气大减。

Activity 是 Android 的四大组件之一，也是平时我们用到最多的一个组件，可以用来显示 View。

官方的说法是 Activity 一个应用程序的组件，它提供一个屏幕来与用户交互，以便做一些诸如打电话、发邮件和看地图之类的事情，原话如下：

An Activity is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map.

Activity 是一个 Android 的应用组件，它提供屏幕进行交互。每个 Activity 都会获得一个用于绘制其用户界面的窗口，窗口可以充满屏幕也可以小于屏幕并浮动在其他窗口之上。

一个应用通常是由多个彼此松散联系的 Activity 组成，一般会指定应用中的某个 Activity 为主活动，也就是说首次启动应用时给用户呈现的 Activity。将 Activity 设为主活动的方法

当然 Activity 之间可以进行互相跳转，以便执行不同的操作。每当新 Activity 启动时，旧的 Activity 便会停止，但是系统会在堆栈也就是返回栈中保留该 Activity。

当新 Activity 启动时，系统也会将其推送到返回栈上，并取得用户的操作焦点。当用户完成当前 Activity 并按返回按钮是，系统就会从堆栈将其弹出销毁，然后回复前一 Activity

当一个 Activity 因某个新 Activity 启动而停止时，系统会通过该 Activity 的生命周期回调方法通知其这一状态的变化。

Activity 因状态变化每个变化可能有若干种，每一种回调都会提供执行与该状态相应的特定操作的机会

2. Activity 生命周期?

周期即活动从开始到结束所经历的各种状态。生命周期即活动从开始到结束所经历各个状态。从一个状态到另一个状态的转变，从无到有再到无，这样一个过程中所经历的状态就叫做生命周期。

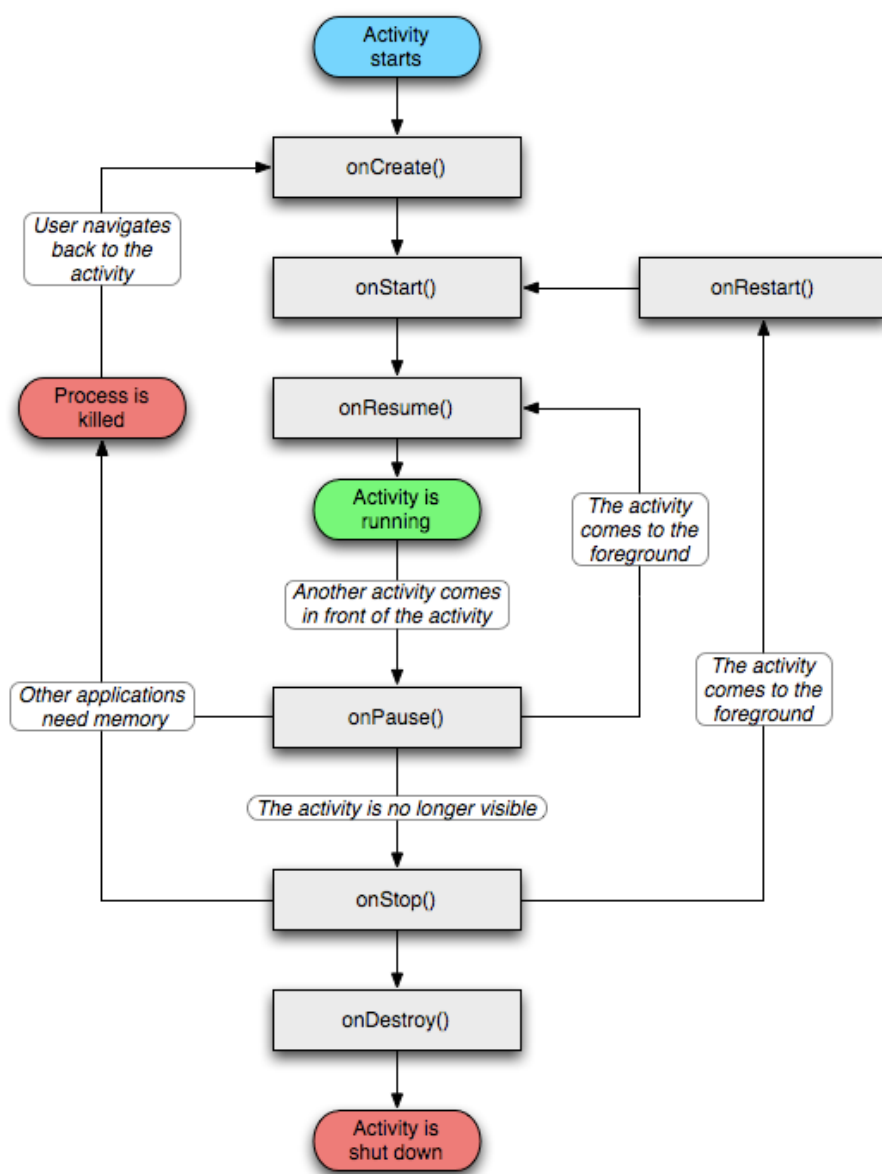
Activity 本质上有四种状态：

1. **运行 (Active/Running)** : Activity 处于活动状态，此时 Activity 处于栈顶，是可见状态，可以与用户进行交互
2. **暂停 (Paused)** : 当 Activity 失去焦点时，或被一个新的非全面屏的 Activity，或被一个透明的 Activity 放置在栈顶时，Activity 就转化为 Paused 状态。此刻并不会被销毁，只是失去了与用户交互的能力，其所有的状态信息及其成员变量都还在，只有在系统内存紧张的情况下，才有可能被系统回收掉
3. **停止 (Stopped)** : 当 Activity 被系统完全覆盖时，被覆盖的 Activity 就会进入 Stopped 状态，此时已不在可见，但是资源还是没有被收回

4. 系统回收 (Killed) :当 Activity 被系统回收掉, Activity 就处于 Killed 状态

如果一个活动在处于停止或者暂停的状态下, 系统内存缺乏时会将其结束 (finish) 或者杀死 (kill)。这种非正常情况下, 系统在杀死或者结束 之前会调用 onSaveInstanceState() 方法来保存信息, 同时, 当 Activity 被移动到前台时, 重新启动该 Activity 并调用 onRestoreInstanceState() 方法加载保留的信息, 以保持原有的状态。

在上面的四中常有的状态之间, 还有着其他的生命周期来作为不同状态之间的过度, 用于在不同的状态之间进行转换, 生命周期的具体说明见下。



原文: https://blog.csdn.net/fjnu_se/article/details/80703815

3. Android 四大组件

Android 四大基本组件：Activity、BroadcastReceiver 广播接收器、ContentProvider 内容提供者、Service 服务。

Activity:

应用程序中，一个 Activity 就相当于手机屏幕，它是一种可以包含用户界面的组件，主要用于和用户进行交互。一个应用程序可以包含许多活动，比如事件的点击，一般都会触发一个新的 Activity。

BroadcastReceiver 广播接收器:

应用可以使用它对外部事件进行过滤只对感兴趣的外部事件(如当电话呼入时，或者数据网络可用时)进行接收并做出响应。广播接收器没有用户界面。然而，它们可以启动一个 activity 或 service 来响应它们收到的信息，或者用 NotificationManager 来通知用户。通知可以用很多种方式来吸引用户的注意力——闪动背灯、震动、播放声音 等。一般来说是在状态栏上放一个持久的图标，用户可以打开它并获取消息。

ContentProvider 内容提供者:

内容提供者主要用于在不同应用程序之间实现数据共享的功能，它提供了一套完整的机制，允许一个程序访问另一个程序中的数据，同时还能保证被访问数据 的安全性。只有需要在多个应用程序间共享数据时才需要内容提供者。例如：通讯录数据被多个应用程序使用，且必须存储在一个内容提供者中。它的好处：统一数据访问方式。

Service 服务:

是 Android 中实现程序后台运行的解决方案，它非常适合去执行那些不需要和用户交互而且还要长期运行的任务（一边打电话，后台挂着 QQ）。服务 的运行不依赖于任何用户界面，即使程序被切换到后台，或者用户打开了另一个应用程序，服务仍然能够保持正常运行，不过服务并不是运行在一个独立的进程当中，而是依赖于创建服务时所在的应用程序进程。当某个应用程序进程被杀掉后，所有依赖于该进程的服务也会停止运行（正在听音乐，然后把音乐程序退出）。

原文: https://blog.csdn.net/m0_37989980/article/details/78681367

4. app 测试和 web 测试有什么区别?

WEB 测试和 App 测试从流程上来说，没有区别。

都需要经历测试计划方案，用例设计，测试执行，缺陷管理，测试报告等相关活

动。

从技术上来说，WEB 测试和 APP 测试其测试类型也基本相似，都需要进行功能测试、性能测试、安全性测试、GUI 测试等测试类型。

他们的主要区别在于具体测试的细节和方法有区别，比如：性能测试，在 WEB 测试只需要测试响应时间这个要素，在 App 测试中还需要考虑流量测试和耗电量测试。

兼容性测试：在 WEB 端是兼容浏览器，在 App 端兼容的是手机设备。而且相对应的兼容性测试工具也不相同，WEB 因为是测试兼容浏览器，所以需要使用不同的浏览器进行兼容性测试（常见的是兼容 IE6，IE8，chrome，firefox）如果是手机端，那么就需要兼容不同品牌，不同分辨率，不同 android 版本甚至不同操作系统的兼容。（常见的兼容方式是兼容市场占用率前 N 位的手机即可），有时候也可以使用到兼容性测试工具，但 WEB 兼容性工具多用 IETester 等工具，而 App 兼容性测试会使用 Testin 这样的商业工具也可以做测试。

安装测试：WEB 测试基本上没有客户端层面的安装测试，但是 App 测试是存在客户端层面的安装测试，那么就具备相关的测试点。

还有，App 测试基于手机设备，还有一些手机设备的专项测试。如交叉事件测试，操作类型测试，网络测试（弱网测试，网络切换）

交叉事件测试：就是在操作某个软件的时候，来电话、来短信，电量不足提示等外部事件。

操作类型测试：如横屏测试，手势测试

网络测试：包含弱网和网络切换测试。需要测试弱网所造成的用户体验，重点要考虑回退和刷新是否会造成二次提交。弱网络的模拟，据说可以用 360wifi 实现设置。

从系统架构的层面，WEB 测试只要更新了服务器端，客户端就会同步会更新。而且客户端是可以保证每一个用户的客户端完全一致的。但是 APP 端是不能够保证完全一致的，除非用户更新客户端。如果是 APP 下修改了服务器端，意味着客户端用户所使用的核心版本都需要进行回归测试一遍。

还有升级测试：升级测试的提醒机制，升级取消是否会影响原有功能的使用，升级后用户数据是否被清除了。

原文地址：<https://www.cnblogs.com/laoluoints/p/5673291.html>

5. android 和 ios 测试区别？

App 测试中 ios 和 Android 有哪些区别呢？

1. Android 长按 home 键呼出应用列表和切换应用，然后右滑则终止应用；
2. 多分辨率测试，Android 端 20 多种，ios 较少；
3. 手机操作系统，Android 较多，ios 较少且不能降级，只能单向升级；新的 ios 系统中的资源库不能完全兼容低版本中的 ios 系统中的应用，低版本 ios 系统中的应用调用了新的资源库，会直接导致闪退（Crash）；
4. 操作习惯：Android，Back 键是否被重写，测试点击 Back 键后的反馈是否正确；应用数据从内存移动到 SD 卡后能否正常运行等；
5. push 测试：Android：点击 home 键，程序后台运行时，此时接收到 push，点击后唤醒应用，此时是否可以正确跳转；ios，点击 home 键关闭程序和屏幕锁屏的情况（红点的显示）；
6. 安装卸载测试：Android 的下载和安装的平台和工具和渠道比较多，ios 主要有 app store，iTunes 和 testflight 下载；
7. 升级测试：可以被升级的必要条件：新旧版本具有相同的签名；新旧版本具有相同的包名；有一个标示符区分新旧版本（如版本号），对于 Android 若有内置的应用需检查升级之后内置文件是否匹配（如内置的输入法）

另外：对于测试还需要注意以下几点：

1. 并发（中断）测试：闹铃弹出框提示，另一个应用的启动、视频音频的播放，来电、用户正在输入等，语音、录音等的播放时强制其他正在播放的要暂停；
2. 数据来源的测试：输入，选择、复制、语音输入，安装不同输入法输入等；
3. push（推送）测试：在开关机、待机状态下执行推送，消息先死及其推送跳转的正确性；

应用在开发、未打开状态、应用启动且在后台运行的情况下是 push 显示和跳转否正确；

推送消息阅读前后数字的变化是否正确；

多条推送的合集的显示和跳转是否正确；

4. 分享跳转：分享后的文案是否正确；分享后跳转是否正确，显示的消息来源是否正确；

5. 触屏测试：同时触摸不同的位置或者同时进行不同操作，查看客户端的处理情况，是否会 crash 等

原文链接：<https://www.jianshu.com/p/91d7acfb036e>

6. app 出现 ANR，是什么原因导致的？

那么导致 ANR 的根本原因是什么呢？简单的总结有以下两点：

1. 主线程执行了耗时操作，比如数据库操作或网络编程
2. 其他进程（就是其他程序）占用 CPU 导致本进程得不到 CPU 时间片，比如其他进程的频繁读写操作可能会导致这个问题。

细分的话，导致 ANR 的原因有如下几点：

1. 耗时的网络访问
2. 大量的数据读写
3. 数据库操作
4. 硬件操作（比如 camera）
5. 调用 thread 的 join() 方法、sleep() 方法、wait() 方法或者等待线程锁的时候
6. service binder 的数量达到上限
7. system server 中发生 WatchDog ANR
8. service 忙导致超时无响应
9. 其他线程持有锁，导致主线程等待超时
10. 其它线程终止或崩溃导致主线程一直等待。

原文：https://blog.csdn.net/jaychou_maple/article/details/78782822

7. App 出现 crash 原因有哪些？

为什么 App 会出现崩溃呢？百度了一下，查到和 App 崩溃相关的几个因素：内存管理错误，程序逻辑错误，设备兼容，网络因素等，如下：

1. 内存管理错误：可能是可用内存过低，app 所需的内存超过设备的限制，app 跑不起来导致 App crash。
或是内存泄露，程序运行的时间越长，所占用的内存越大，最终用尽全部内存，导致整个系统崩溃。
亦或非授权的内存位置的使用也可能会导致 App crash。
2. 程序逻辑错误：数组越界、堆栈溢出、并发操作、逻辑错误。
e. g. app 新添加一个未经测试的新功能，调用了一个已释放的指针，运行的时候就会 crash。
3. 设备兼容：由于设备多样性，app 在不同的设备上可能会有不同的表现。
4. 网络因素：可能是网速欠佳，无法达到 app 所需的快速响应时间，导致 app crash。或者是不同网络的切换也可能会影响 app 的稳定性。

原文：<https://blog.csdn.net/yangtuxiaojie/article/details/47123243>

8. app 对于不稳定偶然出现 anr 和 crash 时候你是怎么处理的？

app 偶然出现 anr 和 crash 是比较头疼的问题，由于偶然出现无法复现步骤，这也是一个测试人员必备的技能，需要抓日志。查看日志主要有 3 个方法：

方法一：app 开发保存错误日志到本地

一般 app 开发在 debug 版本，出现 anr 和 crash 的时候会自动把日志保存到本地实际的 sd 卡上，去对应的 app 目录取出来就可以了

方法二：实时抓取

当出现偶然的 crash 时候，这时候可以把手机拉到你们 app 开发那，手机连上他的开发代码的环境，有 ddms 会抓日志，这时候出现 crash 就会记录下来日志。尽量重复操作让 bug 复现就可以了

也可以自己开着 logcat，保存日志到电脑本地，参考这篇：

<https://www.cnblogs.com/yoyoketang/p/9101365.html>

```
adb logcat | find "com.sankuai.meituan" >d:\hello.txt
```

方法三：第三方 sdk 统计工具

一般接入了第三方统计 sdk，比如友盟统计，在友盟的后台会抓到报错的日志

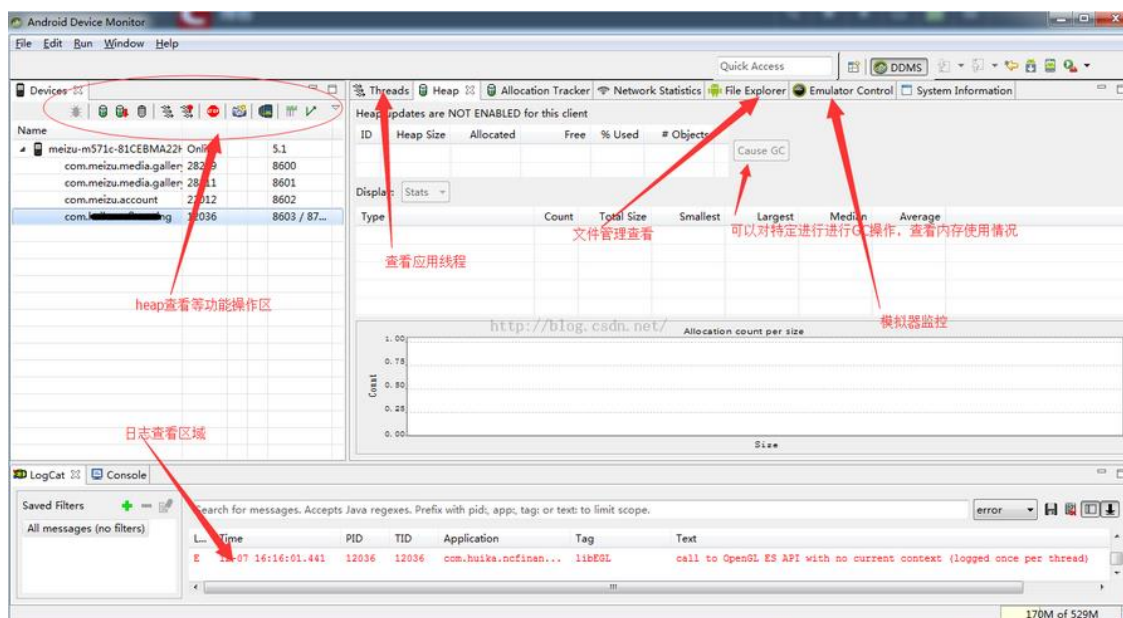
9. app 的日志如何抓取？

app 本身的日志，可以用 logcat 抓取，参考这篇：

<https://www.cnblogs.com/yoyoketang/p/9101365.html>

```
adb logcat | find "com.sankuai.meituan" >d:\hello.txt
```

也可以用 ddms 抓取，手机连上电脑，打开 ddms 工具，或者在 Android Studio 开发工具中，打开 DDMS



关于 ddms 更多的功能，参考这篇：

<https://www.cnblogs.com/gaobig/p/5029381.html>

10. 你平常会看日志吗，一般会出现哪些异常（Exception）？

这个主要是面试官考察你会不会看日志，是不是看得懂 java 里面抛出的异常，Exception

一般面试中 java Exception (runtimeException) 是必会被问到的问题
app 崩溃的常见原因应该也是这些了。常见的异常列出四五种，是基本要求。

常见的几种如下：

NullPointerException - 空指针引用异常

ClassCastException - 类型强制转换异常。

IllegalArgumentException - 传递非法参数异常。

ArithmeticException - 算术运算异常

ArrayStoreException - 向数组中存放与声明类型不兼容对象异常

IndexOutOfBoundsException - 下标越界异常

NegativeArraySizeException - 创建一个大小为负数的数组错误异常

NumberFormatException - 数字格式异常

SecurityException - 安全异常

UnsupportedOperationException - 不支持的操作异常

七、jenkins 持续集成-加分项

个人博客地址: <http://www.cnblogs.com/yoyoketang/>

QQ 交流群: 779429633

联系本人 qq: 283340479

Jenkins 持续集成也是比较火的，面试问的持续集成相关，基本上就是加分项了。要是学会了，那就是送分题

主要有 3 个考点

- 1.如何运行代码
- 2.如何新增 slave 节点
- 3.如何定时构建
- 4.如何结合 git 自动拉取代码

交流 qq 群: **779429633**

如何运行代码？

1 打开 jenkins，新增一个任务



The image shows the Jenkins 'New Item' configuration page. At the top, there is a section titled '输入一个任务名称' (Enter a task name) with a text input field containing 'yoyo' and a label '» 必填项' (» Required). Below this, there are four options for creating a new job, each with an icon and a description:

- 构建一个自由风格的软件项目** (Build a free-style software project): 这是Jenkins的主要功能. Jenkins将会结合任何SCM和任何构建系统来构建你的项目, 甚至可以构建软件以外的系统.
- 构建一个maven项目** (Build a maven project): 构建一个maven项目. Jenkins利用你的POM文件, 这样可以大大减轻构建配置.
- External Job**: This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that use Jenkins as a dashboard of your existing automation system.
- 构建一个多配置项目** (Build a multi-configuration project): 适用于多配置项目, 例如多环境测试, 平台指定构建, 等等.

Below these options, there is a section titled '如果你想根据一个已经存在的任务创建, 可以使用这个选项' (If you want to create a new job based on an existing one, you can use this option). It features a '复制' (Copy) button and a text input field containing '输入自动完成' (Input auto-complete).

At the bottom left, there is a '确定' (OK) button.

2 添加构建步骤-Execute Windows batch command



The image shows the Jenkins 'Build' configuration page. A dropdown menu titled '增加构建步骤' (Add build step) is open, displaying a list of build steps. The first option, 'Execute Windows batch command', is highlighted with a red border. Other options include 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'.

At the bottom of the page, there are two buttons: '保存' (Save) and 'Apply'. The footer of the page contains the text '生成 页面: 2017-8-27 下午11时25分07秒' (Generated page: 2017-8-27 11:25:07 AM), a link to 'REST API', and 'Jenkins ver. 2'.

3. 配置运行 python 脚本的指令



如何新增 slave 节点

我们在跑自动化项目的时候，希望有单独的测试机能跑自动化项目，并且能集成到 jenkins 上构建任务。如果公司已经有 jenkins 环境了，那无需重新搭建。

只需在现有的平台基础上，新增一个 slave 节点管理就行了。

一、管理节点

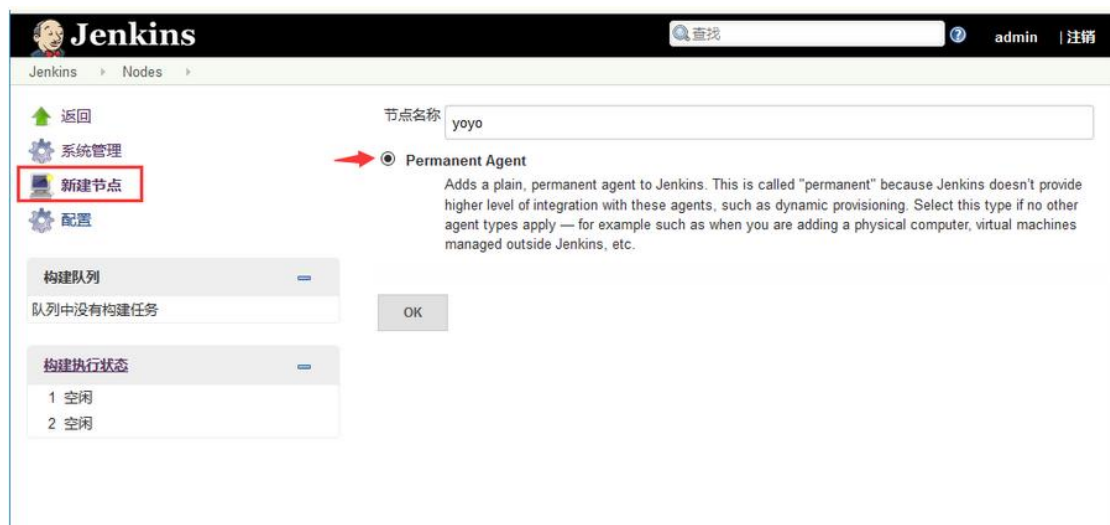
1. 打开需要配置的测试电脑（一定要在需要配置的机器上操作），浏览器输入 jenkins 地址
2. 系统管理-管理节点



3. 新建节点，给节点命名：yoyo

选中 Permanent Agent

点 OK



二、配置节点

1. 远程工作目录：D:\YOYO

—这个地址是测试机的 jenkins 的 workspace 工作目录，自己随便写个本机的路径

2. 用法

--尽可能的使用这个节点：其它的 job 也能在这台机器上运行, 如果只想运行测试自动化代码，就不选这个

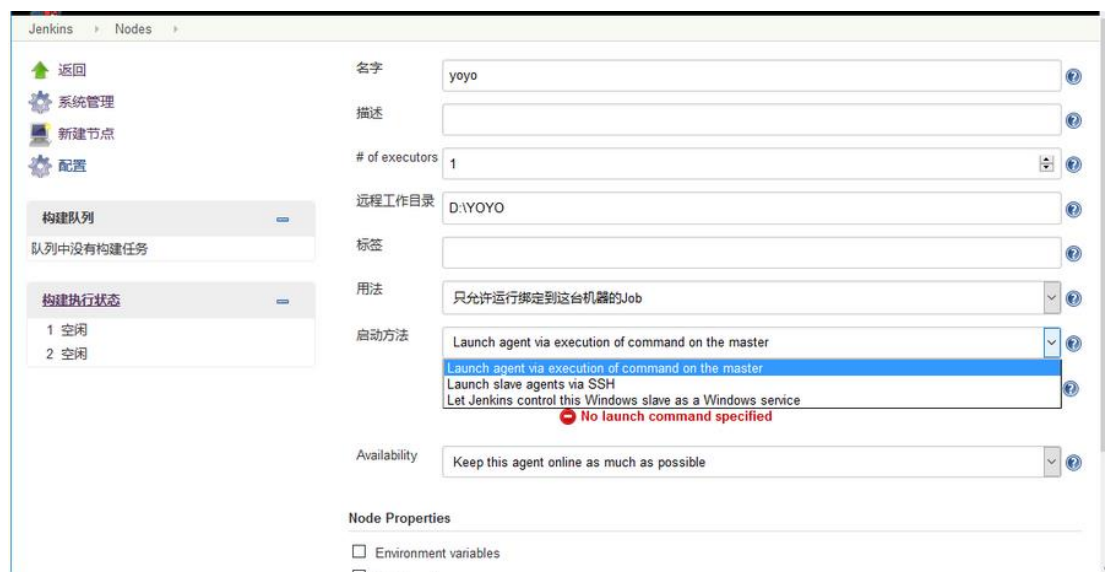
--只允许运行绑定到这台机器的 Job：测试自动化的 Job 绑定这个机器后，就只能在这个机器运行了

3. 启动方法：

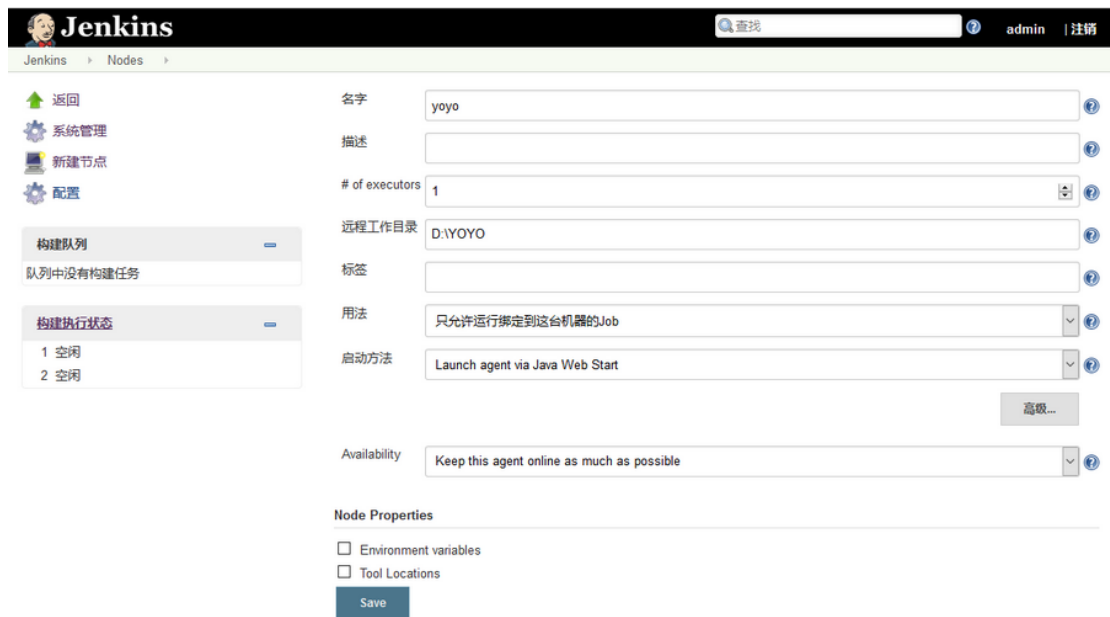
--windows 上启动方法最好通过 Java web start 来启动 slave

（如果没这个选项就看下一步）

--linux 上启动选：Launch slave agents via SSH

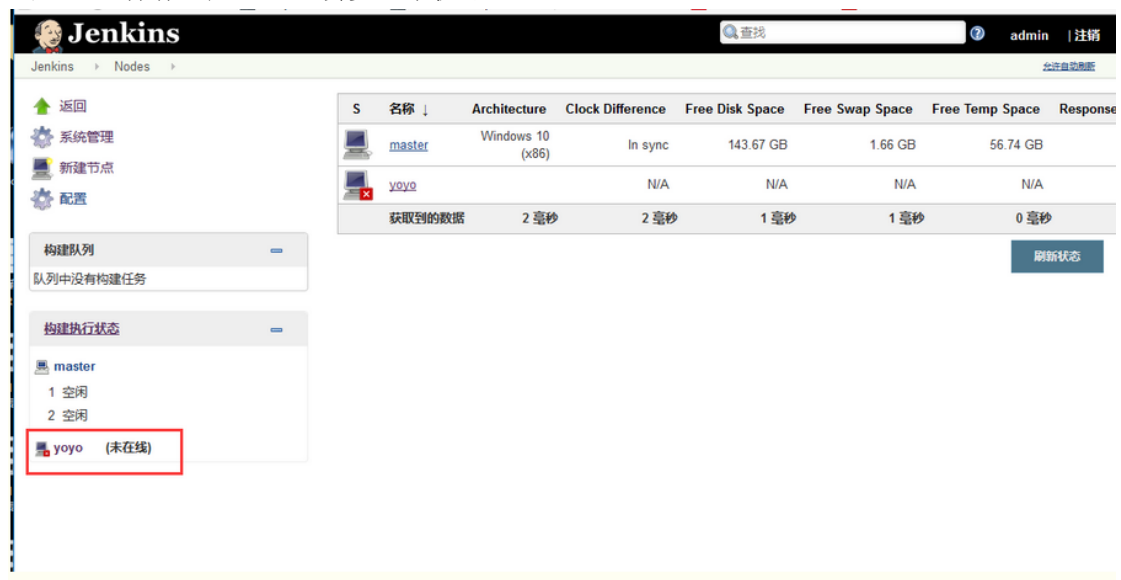


4. 如果有的话，启动方法勾选：Launch agents Java Web Start



The image shows the Jenkins 'New Node' configuration page. The left sidebar contains navigation links: '返回' (Back), '系统管理' (System Management), '新建节点' (New Node), and '配置' (Configure). Below these are two expandable sections: '构建队列' (Build Queue) showing '队列中没有构建任务' (No build tasks in queue) and '构建执行状态' (Build Execution Status) showing '1 空闲' (1 Idle) and '2 空闲' (2 Idle). The main form fields are: '名字' (Name) set to 'yoyo', '描述' (Description) empty, '# of executors' set to '1', '远程工作目录' (Remote workspace) set to 'D:\YOYO', '标签' (Labels) empty, '用法' (Usage) set to '只允许运行绑定到这台机器的Job' (Only allow jobs bound to this machine), '启动方法' (Startup method) set to 'Launch agent via Java Web Start', and 'Availability' set to 'Keep this agent online as much as possible'. At the bottom, there are checkboxes for 'Environment variables' and 'Tool Locations', and a 'Save' button.

5.点 Save 保存之后，左边会多一个机器



The image shows the Jenkins 'Nodes' page after saving the new node. The left sidebar is the same as the previous image. The main area displays a table of nodes. The 'yoyo' node is now listed as '未在线' (Offline). Below the table is a '刷新状态' (Refresh Status) button.

S	名称 ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response
	master	Windows 10 (x86)	In sync	143.67 GB	1.66 GB	56.74 GB	
	yoyo		N/A	N/A	N/A	N/A	
	获取到的数据	2 毫秒	2 毫秒	1 毫秒	1 毫秒	0 毫秒	

三、Java web start

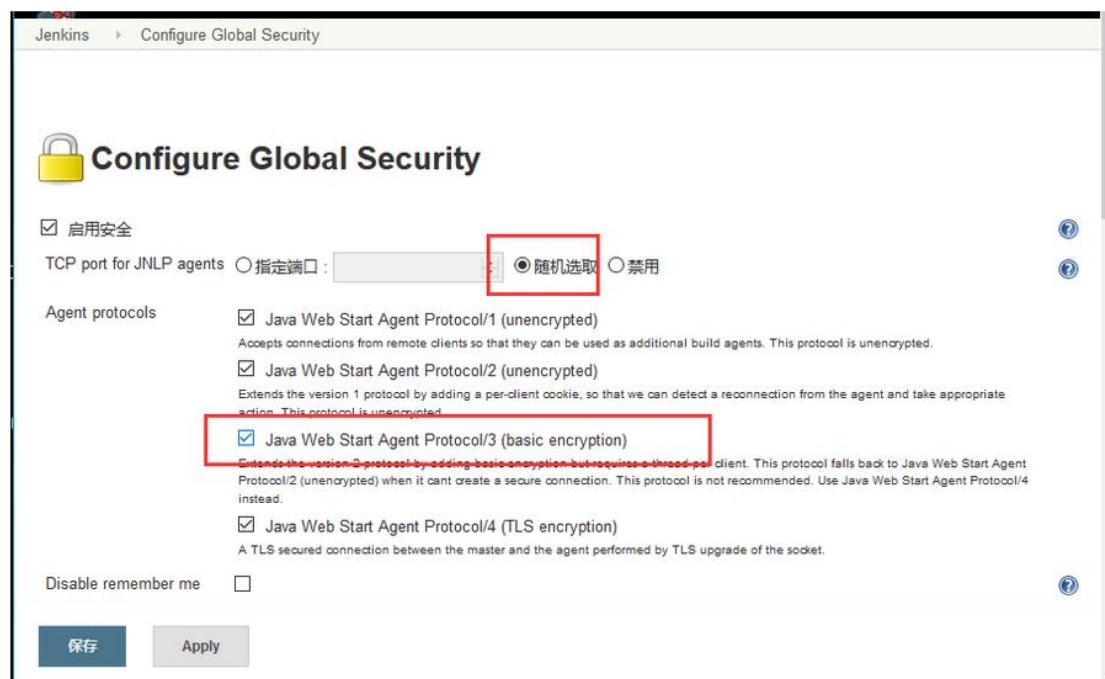
1. windows 上最稳的启动方法，选 Java web start 这个选项，如果没有的话，先去系统管理开启

2. 系统管理--Configure Global Security



3. 点开后，勾选‘随机选取’

点开 Agent Protocols 选项，勾选 Java Web Start Agent

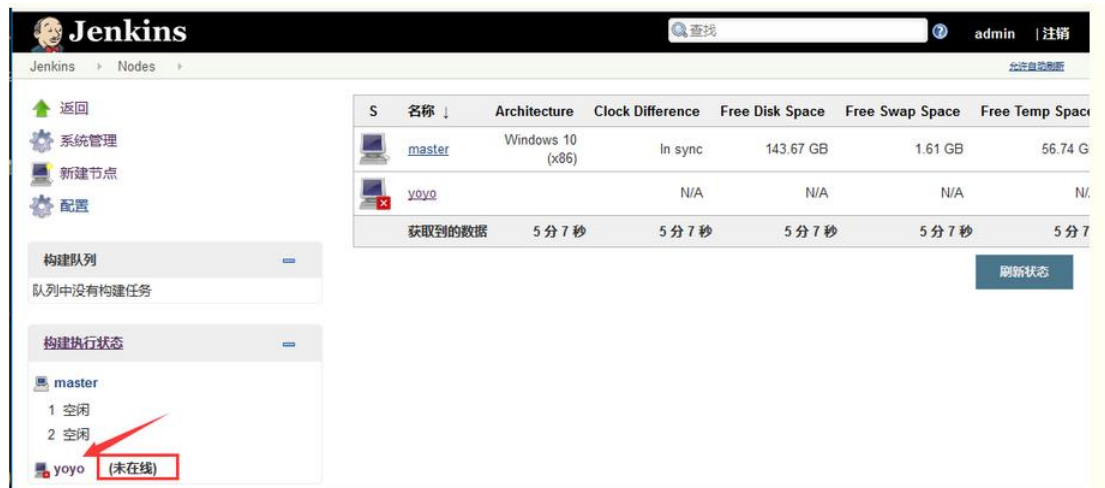


4. 点 Apply，再点保存按钮

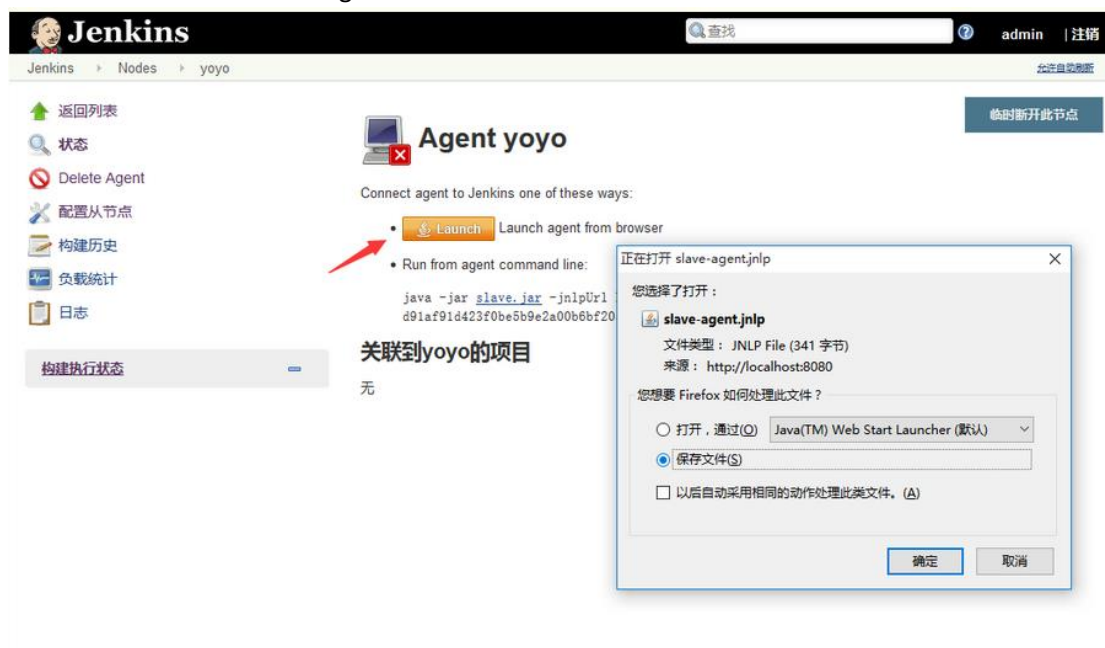
5. 再回到上面的配置节点，启动方法，勾选 Java web start

四、启动 Salve

1. 上面配置的节点是未在线状态，如：yoyo(未在线)，这个时候是不能构建任务的



2. 点开它，本地安装 slave-agent



3. 点上图的 Launch 按钮，会弹出下载提示，可以保存到本地电脑，放到上面新建的 D:\YOYO 目录，方便以后查找



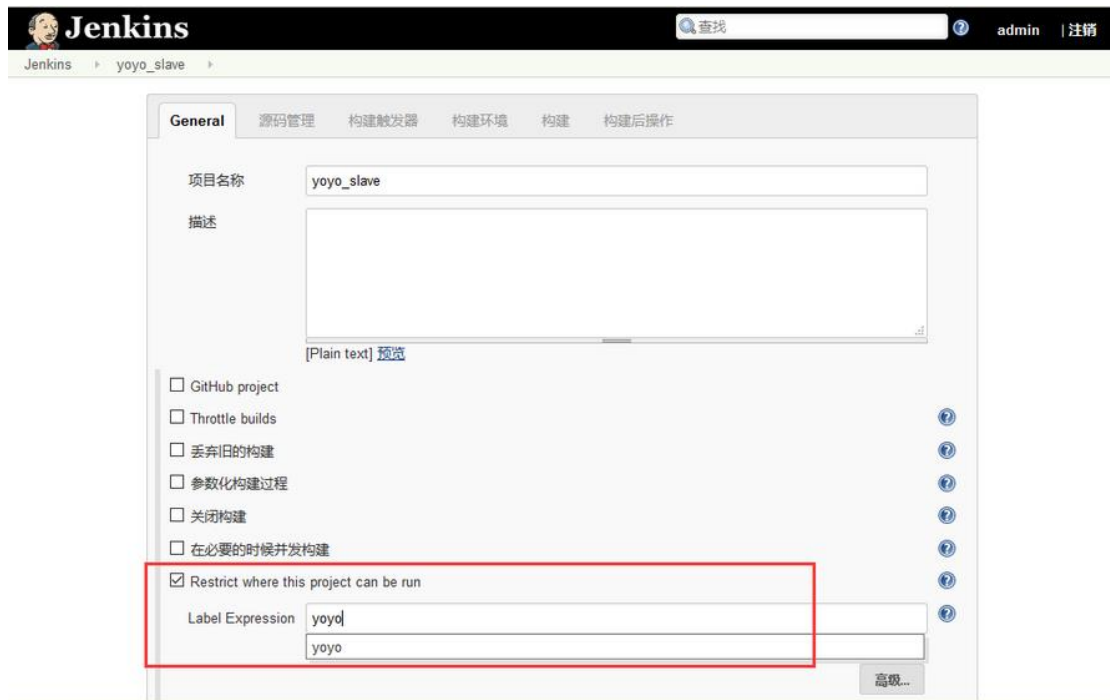
4. 双击 slave-agent.jnlp 启动，启动后桌面上会多个图标



5. 回到jenkins首页后，这个节点就是在线状态了

五、新建 Job

1. 新建的 Job，指定到刚才新增的这个节点机器上运行，勾选 Restrict where this project can be run
2. Label Expression 这个地方输入刚才节点机器的名称：yoyo



如何设置定时任务？

跑自动化用例每次用手工点击 `jenkins` 出发自动化用例太麻烦了，我们希望能每天固定时间跑，这样就不用管了，坐等收测试报告结果就行。

一、定时构建语法

* * * * *

(五颗星，中间用空格隔开)

第一颗*表示分钟，取值 0~59

第二颗*表示小时，取值 0~23

第三颗*表示一个月的第几天，取值 1~31

第四颗*表示第几月，取值 1~12

第五颗*表示一周中的第几天，取值 0~7，其中 0 和 7 代表的都是周日

1. 每 30 分钟构建一次：

H/30 * * * *

2. 每 2 个小时构建一次

H H/2 * * *

3. 每天早上 8 点构建一次

0 8 * * *

4. 每天的 8 点, 12 点, 22 点, 一天构建 3 次

0 8,12,22 * * *

(多个时间点, 中间用逗号隔开)

二、Build periodically

1. Build periodically: 周期性进行项目构建, 这个是在指定的时间必须触发构建任务

2. 比如我想在每天的 9 点, 17 点, 朝九晚五各构建一次, 在 Build periodically 里设置如下



3. 上面红色字体: Spread load evenly by using 'H 9,17 * * *' rather than '0 9,17 * * *', 这句话大概意思就是说, 用这个语法会比后面那个好: H 9,17 * * *

General 源码管理 **构建触发器** 构建环境 构建 构建后操作

☐ Subversion

构建触发器

☐ 触发远程构建 (例如,使用脚本)

☐ Build after other projects are built

☒ Build periodically

日程表

Would last have run at 2017年8月30日 星期三 下午05时48分06秒 CST; would next run at 2017年8月31日 星期四 上午09时48分06秒 CST.

☐ GitHub hook trigger for GITScm polling

☒ Poll SCM

4. 下一次构建时间是 05 时 48 分 06 秒, 然后再下次是 09 时 48 分 06 秒

二、Poll SCM

1. Poll SCM: 定时检查源码变更 (根据 SCM 软件的版本号), 如果有更新就 checkout 最新 code 下来, 然后执行构建动作

2. 如果我想每隔 30 分钟检查一次源码变化, 有变化就执行

General 源码管理 **构建触发器** 构建环境 构建 构建后操作

☐ Subversion

构建触发器

☐ 触发远程构建 (例如,使用脚本)

☐ Build after other projects are built

☐ Build periodically

☐ GitHub hook trigger for GITScm polling

☒ Poll SCM

日程表

Would last have run at 2017年8月30日 星期三 下午10时18分42秒 CST; would next run at 2017年8月30日 星期三 下午10时48分42秒 CST.

Ignore post-commit hooks ☐

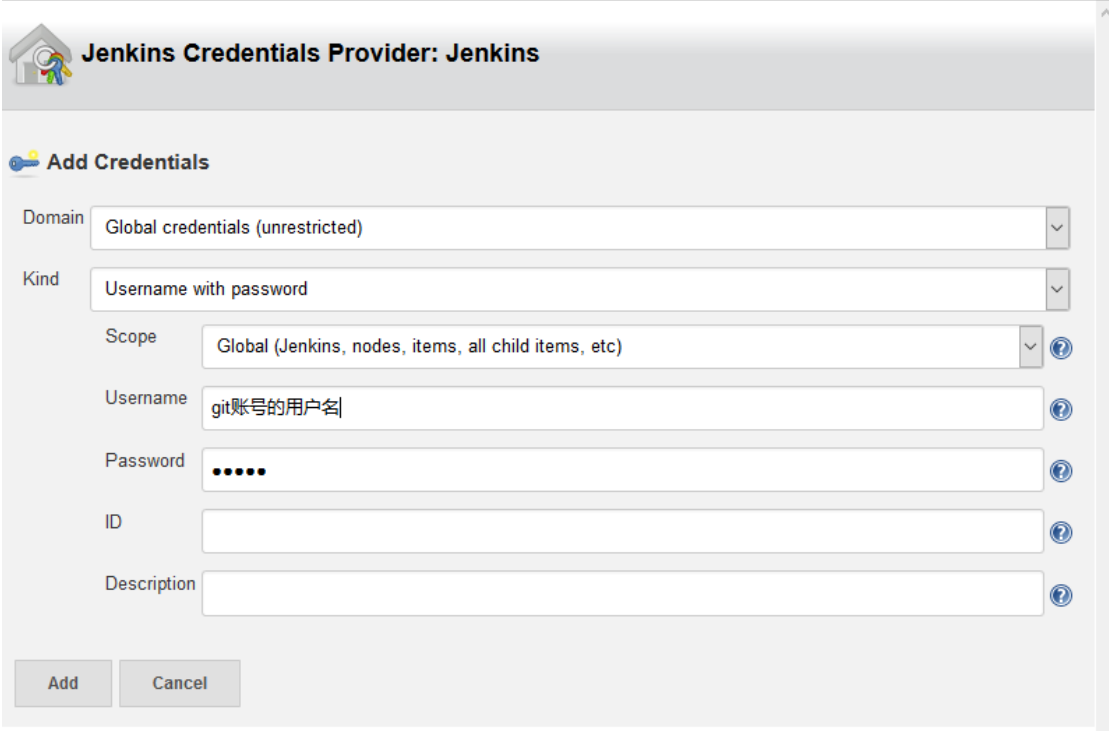
如何结合 git 拉取代码？

源码管理：git

1. 源码管理里面勾选 Git 选项
2. Repository URL 这个地址就是我 git 仓库的地址

The screenshot shows the Jenkins configuration interface for 'Source Management' (源码管理). The 'General' tab is active. Under 'Source Management', the 'Git' option is selected. The 'Repository URL' field contains 'https://github.com/yoyoketang/yoyo.git'. The 'Credentials' dropdown menu shows 'yoyoketang/*****' and an 'Add' button. The 'Branches to build' section has a 'Branch Specifier (blank for \'any\')' field set to '*/master'. The 'Source Code Browser' is set to '(自动)'. At the bottom, there are '保存' (Save) and 'Apply' buttons.

3. Credentials 点 add 添加 git 的账号和密码



The image shows the 'Jenkins Credentials Provider: Jenkins' interface. It features a header with the Jenkins logo and title. Below is a section titled 'Add Credentials' with a key icon. The form includes several fields: 'Domain' (a dropdown menu set to 'Global credentials (unrestricted)'), 'Kind' (a dropdown menu set to 'Username with password'), 'Scope' (a dropdown menu set to 'Global (Jenkins, nodes, items, all child items, etc)' with a help icon), 'Username' (a text input field containing 'git账号的用户名' with a help icon), 'Password' (a password input field with masked dots and a help icon), 'ID' (an empty text input field with a help icon), and 'Description' (an empty text input field with a help icon). At the bottom are 'Add' and 'Cancel' buttons.

4.Branch Specifier (blank for 'any')这个地方的地址是分支名, 如果没指定其它分支, 就是默认: */master

2019 课程推荐

2019 年《python 全栈自动化测试课程》2 月 16 号开学!

主讲老师: 上海-悠悠

上课方式: QQ 群视频在线教学

上课时间: 每周六、周日晚上 20:30-22:30

报名费: 3 个课程费用一起 2888

联系 QQ: 283340479

上课安排:

python 语法讲解, 讲课过程中用到的语法都会讲解、 selenium (web 自动化)、 git+jenkins 持续集成、

面试专题 2 节课(专为金三银四面试而准备)、python 接口自动化、appium 自动化

注意：python 零基础的同学提前报名给资料预习，用到的语法课程中会讲解

课表详情：<https://mp.weixin.qq.com/s/paywUGdkasx-t9S0b08Jtg>

好书推荐

学完此文档觉得对你有帮助的，可以关注我的更多作品！

Selenium+python

要想学好 appium，必先学 selenium，这里推荐一本 葵花和海 上海-悠悠一起写的 selenium+python 的书。

《selenium webdriver 基于 Python 源码案例》本书上架百度阅读以来，获得了非常好的评价，10 分好评，近五万的阅读量，是一本适合初学者的从入门到精通。

<https://yuedu.baidu.com/ebook/0f6a093b7dd184254b35eefdc8d376eeaeaa17e3>



在数十万册图书中搜索

搜索图书

何所冬暖，何所夏凉 戮刺灼亮 无证之罪 万历十五年 我们仨 冰与火之歌

首页

分类

榜单

独家作品

机构专区

客户端



selenium webdriver基于Python源码案例

百度阅读作者计划 selenium webdriver 自动化测试 Python

★★★★★ 10 (145人评论) | 49947人在读

本书概述：本书是非常适合小白入门的，是接地气，符合人类阅读习惯的，书中大量的实际案例分析，图文详解（seleniumpython读书QQ群372471871，凭订单号加入可获取一份对应PDF文档）小编是个实在的人，没太多的理论道... 查看全部

价格：¥50.00 已购买

同步到手机

开始阅读

作者最近更新

已累计 950 +1 次催更新



葵花、上海-悠悠 2017-10-29 有更新 更新12.5 截取某个元素的图

Python 接口自动化

第二本是本人写的《python 接口自动化》的书，虽然没 10 分好评，9.8 分也是获得了不错的评价，20w 的阅读量

<https://yuedu.baidu.com/ebook/585ab168302b3169a45177232f60ddccda38e695>

首页

分类

榜单

独家作品

机构专区

客户端



Python接口自动化测试

百度阅读作者计划 python接口 接口测试 接口 接口自动化 selenium appium requests

★★★★★ 9.8 (178人评论) | 205934人在读

本书概述：本书是非常适合小白入门接口自动化的（购买此书（活动赠送的无PDF）联系作者领取一份对应PDF版QQ交流群：226296743）。本书涉及内容：fiddler抓包、http协议详解、json、python+requests写接口请求、unittest单元... 查看全部

价格：¥40.00

购买全本

开始阅读

加入购物车

交流 qq 群: 779429633

python 自动化框架 pytest

pytest 是最强大最好用的 python 自动化框架，没有之一。本书详细讲解 pytest 框架使用方法，fixture 功能是 pytest 的精髓，书中有详细的案例讲解。另外最后会有项目实战代码，灵活用到 selenium 自动化项目上。声明：本书不适合 python 零基础，不适合 selenium 零基础的同学，本书不讲元素定位，不讲 python 基础，主要讲 pytest 框架运用，基于 python3.6 版本。<https://yuedu.baidu.com/ebook/902224ab27fff705cc1755270722192e4536582b>

[首页](#)[分类](#)[榜单](#)[独家作品](#)[机构专区](#)[客户端](#)

python自动化框架pytest

百度阅读作者计划 python pytest 自动化 测试 selenium

★★★★★ 10 (12条评论) | 27361人在读

本书概述：pytest是最强大最好用的python自动化框架，没有之一。本书详细讲解pytest框架使用方法，fixture功能是pytest的精髓，书中有详细的案例讲解。另外最后会有项目实战代码，灵活用到selenium自动化项目上。声明：本书... [查看全部](#)

价格：¥25.00

[购买全本](#)[开始阅读](#)[加入购物车](#)

Appium+python

第四本 Appium 的自动化，只能算个半成品，需结合 selenium 这本看，是 selenium 的续集！

《appium 自动化入门（图文教程）-python》

<https://yuedu.baidu.com/ebook/7d75728ca0c7aa00b52acfc789eb172ded63991c>



在数十万册图书中搜索

搜索图书

何所冬暖，何所夏凉 数刻尔亮 无证之罪 万历十五年 我们仨 冰与火之歌

首页 分类 榜单 独家作品 机构专区 客户端



Appium自动化入门

上海-悠悠 著

CD 百度网盘

Appium自动化入门级（图文教程）-python 更新中

百度阅读作者计划 appium selenium python 自动化测试 app自动化

★★★★★ 10 (13人评价) | 9486人在读

本书概述：appium环境搭建初级入门教程，本教程只是初级入门，从环境搭建到一个简单的demo。小编目前暂时没精力去更新appium教程，目前只写了前面两章内容，分享出来，希望帮助更多的appium爱好者解决环境入门问题，另外本... [查看全部](#)

价格：¥2.00 已购买

同步到手机

开始阅读

作者最近更新

已累计 49 +1 次催更新



上海-悠悠 2017-11-03 有更新 更新2.13 模拟手势动作，点击元素坐标tap方法