# 文件发布管理

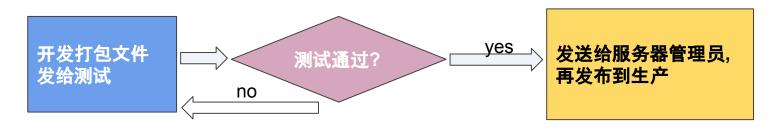
# 苏桂云



#### 起源

- 1. 发文件过程比较繁琐,得通过邮件方式传输
- 2. 各个环节响应结果时间较长
- 3. 代码测试不通过又得重复上面流程







#### 来点情怀

\_\_\_\_

所有我们有一个小目标...





#### 自动化部署

- 1. 获取源码
- 2. 获**取依**赖
- 3. 构建软件包
- 4. 生成/<mark>上传</mark>安装包
- 5. 目标平台安装/配置



#### 提纲

- □ 文件类型、文件管理工具、文件发布流程
- □ 技术实现

#### 常见文件类型

\_\_\_\_

html、jsp、css、js、图片、字体、音视频文件, 等等。

这些文件都不需要编译,直接发布。



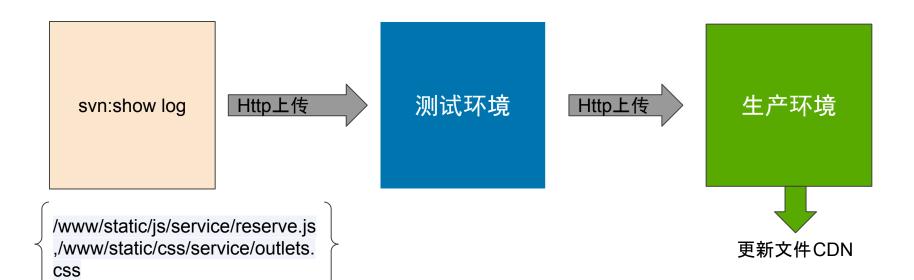
#### SVII项目管理工具

利用show log获取文件信息,拿到文件路径。

<u>D</u>	Filter by Messages, Paths, Authors, Revisions, Bug-IDs, Date, Date Range					From:	2016/ 9
Revision	Actions	Author	Date	Message			
9244	<b>6</b>	suguiyun	2016年9月12日 16:17:30	官网购买链接修改指向360商城			
9243	•	suguiyun	2016年9月12日 16:17:22	官网购买链接修改指向360商城			
9242	<b>₽₽</b>	suguiyun	2016年9月12日 16:16:02	官网购买链接修改指向360商城			
9241	6	hewendong	2016年9月12日 15:10:56	update banner img			
官网购买链技 Path	姜修改指向360	商城			Action	Copy from path	Revision

#### 文件发布流程

\_\_\_\_





#### 技术实现

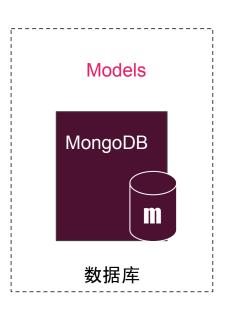
\_\_\_\_







Server



Database



### Nodejs:利用 Child Process获取文件信息

```
const spawn = require('child_process').spawn;
const ls = spawn('svn', ['log', '-v','-1',50]);
ls.stdout.on('data', (data) => {
    console.log(`stdout: ${data}`);
});
ls.stderr.on('data', (data) => {
    console.log(`stderr: ${data}`);
});
ls.on('close', (code) => {
    console.log(`child process exited with code ${code}`);
});
```

## Nodejs:利用formidable上传文件

```
let formidable = require('formidable');
function uploadfiles(res, req){
    let form = new formidable.IncomingForm();
    form.parse(req,function(error, fields, files){
        var types = files.upload.name;
        var date = new Date();
        var ms = Date.parse(date);
        fs.renameSync(files.upload.path,"/tmp/files"+ ms +"."+types;
    });
}
```

#### 项目中用到的Linux命令

```
#!/bin/sh
rm -rf view build/;
mkdir view build;
cp -r view/* view_build;
tar zcf ../output.tar.gz *;
netstat -antp;
ps -ef|grep node;
In -s;
find / -name node*;
kill -2 `pgrep mongod`;
sudo rz -y;
```

不会运维的前端开发不是好的架构师



#### 总结:整个发布流程做三件事

\_\_\_\_

- 1. 文件获取
- 2. 上传文件

3.实时监听"文件获取"和"上传文件"并把结果立即反馈给用户





# Q&A

