

目录

Part I - 开源项目

项目导航	1.1
联系方式	1.2
我的博客	1.3
开源项目	1.4
SpringBoot集成MyBatisPlus	1.4.1
基于SpringBoot + Shiro + MyBatisPlus的权限管理框架	1.4.2
微信小程序+微信管理后台+微信用户前台	1.4.3
自制spring boot starter for fastjson	1.4.4
一个会动的简历	1.4.5

Part II - Java

SpringBoot2.0更新记录	2.1
SpringBoot集成Druid不支持多条SQL	2.2
SpringBoot设置默认首页	2.3
Ajax请求设置Type为Form-Data	2.4

Part III - 前端开发笔记

前端开发笔记	3.1
移动Web开发技巧汇总	3.1.1
前端开发注意事项	3.1.2
nodejs项目配置文件介绍	3.1.3
gulpfile.js	3.1.3.1
package.json	3.1.3.2
config.rb	3.1.3.3
.gitignore	3.1.3.4
模块化管理工具及框架	3.1.4
Webpack	3.1.4.1
seajs	3.1.4.2
js模块	3.1.5
时间相关模块	3.1.5.1

common.js	3.1.5.2
js及jquery获取页面和元素的宽高	3.1.5.3
js常用函数	3.1.5.4
ajax写法	3.1.5.5
常用插件	3.1.6
常用弹层插件	3.1.6.1
常用效果插件	3.1.6.2

Part IV - Hexo博客

Hexo博客yilia主题添加背景音乐	4.1
Hexo博客添加helper-live2d动态模型插件	4.2
Hexo博客添加访问量统计	4.3
使用Hexo+GitHub搭建免费个人博客	4.4
新增Hexo博客文章置顶功能	4.5

Part V - 日记本

Git分支的创建、删除、切换	5.1
Git彻底删除历史提交记录的方法	5.2
GitBook的安装、卸载、常见问题	5.3
GitBook相关配置及优化	5.4
IDEA+JRebel全自动热部署	5.5
IDEA安装lombok及使用详解	5.6
IDEA设置后端代码开发规范	5.7
解决Tomcat在IDEA控制台中文乱码的问题	5.8
玩转MySQL优化命令	5.9
BitComet服务器列表	5.10

Part VI - Linux

Linux下安装GLIBC_2.15	6.1
Linux安装python2-7-9	6.2
Linux下安装rabbitMq	6.3
Linux下安装Erlang	6.4
Linux常用命令	6.5
Linux查看硬件信息	6.6

Linux下同步网络时间	6.7
Linux或Win下安装node和npm	6.8
Linux下安装Git	6.9
Linux下Tomcat指定JDK和设置内存大小	6.10
nohup和>/dev/null 2>&1	6.11
Linux临时和永久关闭selinux	6.12
vi-vim-粘贴内容时,保留原文件内容格式	6.13
CentOS6.5-离线安装MySQL5.6.26	6.14
更改MySQL的存储目录	6.15
Linux开启MySQL日志	6.16
解决MariaDB中文乱码	6.17
CentOS-6.X-安装VirtualBox-5.1	6.18
CentOS-6.X-安装VNC-Server实现图形化访问	6.19
CentOS-6.X-图形界面安装	6.20
Linux中Apache(httpd)安装、配置	6.21
Linux-Apache服务不能访问文件解决方案	6.22
Linux安装OpenResty-api-网关-Orange	6.23
Linux搭建GitLab并汉化	6.24
Linux配置SSH-Key到GitHub	6.25
Linux下fsck-ext4-Unable-to-resolve问题记录	6.26
Linux-卸载分区	6.27
Linux-新建扩展分区	6.28
Centos用parted分区超过2TB硬盘-分区格式化	6.29
fdisk分区不超过2T的磁盘	6.30
Linux中bond网卡绑定方法	6.31
Linux-查看网卡UUID	6.32

Part VII - ELK日志分析

Linux-搭建-ElasticSearch5.x	7.1
搭建ElasticSearch-6.1.3分布式集群	7.2
Elasticsearch-5.5.0安装head插件	7.3
ElasticSearch-2-x-x版本安装bigdesk插件	7.4
logstash收集tomcat_log	7.5
Kibana使用高德地图	7.6
ELK之本地Grok-Debugger配置	7.7

- **1. myBook**
 - [1.1. 简介](#)
 - [1.2. 安装运行](#)
 - [1.3. 交流 & 提问](#)
 - [1.4. 捐赠](#)

1. myBook

一直在努力,从未放弃

| pdf版 <http://zhousiwei.gitee.io/mybook/mybook.pdf>

1.1. 简介

- [开源项目](#)
- [我的博客](#)
- [联系方式](#)

1.2. 安装运行

```
# gitbook init  
# gitbook install  
# gitbook serve
```

| 关于GitBook的安装和使用 <https://www.jianshu.com/p/1f78d8018ea7>

1.3. 交流 & 提问

| <https://gitee.com/zhousiwei/myBook/issues>

1.4. 捐赠

如果感觉对您有帮助, 请作者喝杯咖啡吧, 请注明您的名字或者昵称, 方便作者感谢o(￣▽￣)o

微信	支付宝

推荐使用微信支付



試毅-思伟(**伟)

支付就用支付宝



周思伟(*思伟)

打开支付宝[扫一扫]

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 18:19:35

- 1. 联系方式
- 2. 打赏
- 3. 开源推荐

1. 联系方式

微信	QQ
	 <p>試毅-思伟  已认证 扫一扫二维码，加我QQ。</p>

2. 打赏

微信	支付宝



特别说明：

1. 打赏-赞助请三思，虚拟产品赞助打赏后恕不退款
2. 打赏 68元 以上可以加个人微信畅聊30分钟，非诚勿扰。
3. 打赏 199元 以上可以一对一答疑。
4. 打赏名单默认不会在网站上显示，如需显示请将付款截屏发送至邮箱 2434387555@qq.com

3. 开源推荐

TIP：不分先后顺序，需加入列表的同学可以联系我。

- 基于 SpringBoot + Shiro + MyBatisPlus 的权限管理框架：<https://gitee.com/zhouxiwei/bootplus>
- 自制spring boot starter for fastjson：<https://gitee.com/zhouxiwei/fastjson-spring-boot-starter>
- SpringBoot集成MyBatisPlus：https://gitee.com/zhouxiwei/springboot_mybatisplus
- 一个会动的简历：<https://gitee.com/zhouxiwei/anires>
- 微信小程序+微信管理后台+微信用户前台 https://gitee.com/zhouxiwei/hqc_mp

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间： 2019-07-01 10:58:56

- 1. 我的博客
- 2. 試毅-思伟的技术博客 <https://zhousiwei.gitee.io/>
 - 2.1. 归档文章
 - 2.2. 捐赠

1. 我的博客

- 个人博客
- GitHub
- 码云
- 简书
- CSDN
- 知乎
- 微博
- 开源项目
- 主要涉及技术: Java后端开发、聚合支付、公众号开发、开源爱好者、Linux

博客将不定期更新，请小伙伴们随时关注哦！

2. 試毅-思伟的技术博客 <https://zhousiwei.gitee.io/>

欢迎来到 試毅-思伟 的技术博客，这里将深入探讨相关技术，包括行业动态，架构设计，设计模式，框架使用等。

- 博客集成了 个人简历、不蒜子访问量统计、404页面、百度提交收录链接、GitBook文档、评论系统Gitment、helper-live2d动态模型插件、背景音乐（网易云音乐）、文章置顶功能

不浮躁，不偏激，不守旧，爱编程，爱后端，爱新技术，爱运动，爱旅行，执行力和学习能力都棒棒哒！

2.1. 归档文章

1. [Java](#)
2. [Linux](#)
3. [ELK日志分析](#)
4. [MySQL](#)
5. [Hexo](#)
6. [日记本](#)

2.2. 捐赠

如果感觉对您有帮助，请作者喝杯咖啡吧，请注明您的名字或者昵称，方便作者感谢o(—~—)o

微信	支付宝
----	-----



Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-01 10:58:56

- 1. 开源项目
 - 1.1. 捐赠

1. 开源项目

- SpringBoot集成MyBatisPlus
- 基于SpringBoot + Shiro + MyBatisPlus的权限管理框架
- 微信小程序+微信管理后台+微信用户前台
- 自制spring boot starter for fastjson
- 一个会动的简历

1.1. 捐赠

如果感觉对您有帮助，请作者喝杯咖啡吧，请注明您的名字或者昵称，方便作者感谢o(￣▽￣)o



Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-01 10:58:56

- 1. SpringBoot集成MyBatisPlus
 - 1.1.1. 项目结构
 - 1.1.2. 技术选型:
 - 1.1.3. 本地部署
 - 1.1.4. 演示效果图:

1. SpringBoot集成MyBatisPlus

GitHub: https://github.com/JoeYBling/SpringBoot_MyBatisPlus

码云: https://gitee.com/zhouxiwei/springboot_mybatisplus

欢迎使用和Star支持, 如使用过程中碰到问题, 可以提出Issue, 我会尽力完善

1.1.1. 项目结构

```
wstro
├── sql   -- 项目SQL语句
|
├── App   -- 项目启动类
|
├── config -- 配置信息
|
├── controller -- 控制器
│   ├── admin -- 后台管理员控制器
|
├── service -- 业务逻辑接口
│   ├── impl -- 业务逻辑接口实现类
|
├── dao   -- 数据访问接口
|
├── entity -- 数据持久化实体类
|
├── datasources -- 多数据源工具类
|
├── shiro  -- Shiro验证框架
|
├── task   -- Quartz定时任务
|
├── util   -- 工具类
│   ├── FreeMarker -- 自定义FreeMarker标签
|
├── resources
│   ├── mapper -- SQL对应的XML文件
│   ├── templates -- FreeMarker模版
|
└── webapp
    ├── statics -- 静态资源
```

```
|   └── upload -- 上传文件  
|   └── WEB-INF  
|       └── templates -- 页面FreeMarker模版
```

1.1.2. 技术选型：

- 核心框架: Spring Boot 1.5.1
- 安全框架: Apache Shiro
- 视图框架: Spring MVC
- 持久层框架: MyBatis 、 MyBatisPlus
- 缓存技术: EhCache 、 Redis
- 定时器: Quartz
- 数据库连接池: Druid
- 日志管理: SLF4J 、 Log4j
- 模版技术: FreeMarker
- 页面交互: Bootstrap 、 Layer 等

1.1.3. 本地部署

- 创建数据库wstro，数据库编码为UTF-8
- 执行sql/wstro.sql文件，初始化数据
- 修改application-dev.properties，更新MySQL账号和密码
- 修改application-dev.properties，更改Redis连接信息
- 如果不想要Redis服务,注解掉 RedisConfig.java 的 @Configuration 注解
- Eclipse、IDEA运行App.java，则可启动项目
- 项目访问路径: <http://localhost:8080/admin>
- 账号密码: admin/admin
- 多数据源配置：需要自己实现，参考 DataSourceConfig.java

1.1.4. 演示效果图：

swagger

admin (/v2/api-docs?group=admin) ▾ Explore

后台相关接口

后台相关接口，主要用于测试。

Created by Joey
See more at <https://github.com/JoeyBling>
Contact the developer
[江西益海微盈信息科技有限公司](#)

接口名	描述	Show/Hide	List Operations	Expand Operations
sys-login-controller : Sys Login Controller		Show/Hide	List Operations	Expand Operations
sys-menu-controller : Sys Menu Controller		Show/Hide	List Operations	Expand Operations
sys-page-controller : Sys Page Controller		Show/Hide	List Operations	Expand Operations
sys-role-controller : Sys Role Controller		Show/Hide	List Operations	Expand Operations
sys-user-controller : Sys User Controller		Show/Hide	List Operations	Expand Operations
sys-user-login-log-controller : Sys User Login Log Controller		Show/Hide	List Operations	Expand Operations

[BASE URL: /WSTRO , API VERSION: 1.0]

2017年7月11日8时55分5秒

首页 管理员列表 ▾ 角色管理 ▾ 监控管理 ▾ Swagger管理 ▾ 菜单管理 ▾ 退出

菜单管理

+ 创建菜单 × 批量删除菜单

菜单名称	上级菜单	菜单图标	菜单URL	授权标识	类型	排序号	操作
系统管理	-	◆	-	sys:user:list,sys:user:info	按钮	0	
查看	管理员列表	-	-	sys:user:save,sys:role:select	按钮	0	
新增	管理员列表	-	-	sys:user:update,sys:role:select	按钮	0	
修改	管理员列表	-	-	sys:user:delete	按钮	0	
删除	管理员列表	-	-	-	菜单	2	
角色管理	系统管理	▲	admin/sys/role.html	-	菜单	3	
菜单管理	系统管理	■	admin/sys/menu.html	-	菜单	3	
监控管理	系统管理	❖	druid/druid-manager	druid:druid:manager	菜单	4	
Swagger管理	系统管理	▷	swagger-ui.html	swaggerui:swaggerui:manager	菜单	1	
管理员列表	系统管理	▲	admin/sys/user.html	-	菜单	0	
查看	菜单管理	-	-	sys:menu:list,sys:menu:info	按钮	0	
新增	菜单管理	-	-	sys:menu:save,sys:menu:select	按钮	0	
修改	菜单管理	-	-	sys:menu:update,sys:menu:select	按钮	0	
删除	菜单管理	-	-	sys:menu:delete	按钮	0	

显示第 1 到第 18 条记录，总共 18 条记录

© 2017-2018 ##

首页 管理员列表 ▾ 退出

新建用户

用户名: ✕
用户名不能为空

密码:

邮箱: ✕
电子邮箱不能为空

手机号: ✕
手机号码不能为空

性别: 男 女 保密

角色: admin

状态: 正常 禁用

提交

新建角色

角色名称： ✕
角色名称不能为空

备注： ✕
备注不能为空

授权：

- 系统管理
 - 管理员列表
 - 查看
 - 新增
 - 修改
 - 删除
 - 角色管理
 - 查看
 - 新增
 - 修改
 - 删除
 - 菜单管理
 - 查看
 - 新增
 - 修改
 - 删除
 - 站控管理

新建菜单

类型: 目录 菜单 按钮

菜单名称:

上级菜单: 一级菜单

菜单URL:

授权标识: 多个用逗号分隔, 如: user:list,user:create

排序号:

图标: [获取图标: http://fontawesome.io/icons/](http://fontawesome.io/icons/)

登录时间	登录IP	操作系统	浏览器
2017-7-11 09:30	192.168.1.12	WINDOWS_7	CHROME49
2017-7-11 08:43	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 17:13	192.168.1.12	WINDOWS_7	CHROME49
2017-7-10 16:55	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 16:35	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 16:34	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 16:32	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 16:30	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 14:58	192.168.1.12	WINDOWS_7	CHROME49
2017-7-10 14:42	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 14:38	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 14:32	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 14:18	192.168.1.88	WINDOWS_7	CHROME49
2017-7-10 14:18	192.168.1.88	WINDOWS_7	CHROME49

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-01 10:58:56

- 1. 基于SpringBoot + Shiro + MyBatisPlus的权限管理框架
- 2. bootplus
 - 2.1.1. 项目结构
 - 2.1.2. 技术选型:
 - 2.1.3. 本地部署:
 - 2.1.4. 效果展示

1. 基于SpringBoot + Shiro + MyBatisPlus的权限管理框架

GitHub: <https://github.com/JoeYBling/bootplus>

码云: <https://gitee.com/zhouxiwei/bootplus>

欢迎使用和Star支持, 如使用过程中碰到问题, 可以提出Issue, 我会尽力完善

2. bootplus

- bootplus 是基于 SpringBoot + Shiro + MyBatisPlus 的真正 restful URL 资源无状态认证权限管理框架

2.1.1. 项目结构

```
bootplus
├── sql   -- 项目SQL语句
|
├── App   -- 项目启动类
|
├── config -- 配置信息
|
├── controller -- 控制器
│   ├── admin -- 后台管理员控制器
|
├── service -- 业务逻辑接口
│   ├── impl -- 业务逻辑接口实现类
|
├── dao   -- 数据访问接口
|
├── entity -- 数据持久化实体类
|
├── shiro  -- Shiro验证框架
|
├── util   -- 工具类
│   ├── FreeMarker -- 自定义FreeMarker标签
|
├── resources
│   ├── mapper -- SQL对应的XML文件
|
└── webapp
```

```
|   └── statics -- 静态资源
|   └── upload -- 上传文件
|   └── WEB-INF
|       └── templates -- 页面FreeMarker模版
```

2.1.2. 技术选型：

- 核心框架: Spring Boot 1.5.1
- 安全框架: Apache Shiro
- 视图框架: Spring MVC
- 持久层框架: MyBatis 、 MyBatisPlus
- 缓存技术: EhCache 、 Redis
- 定时器: Quartz
- 数据库连接池: Druid
- 日志管理: SLF4J 、 Log4j
- 模版技术: FreeMarker
- 页面交互: BootStrap 、 Layer 等

2.1.3. 本地部署：

- mysql执行 sql/bootplus.sql 文件, 初始化数据
- 修改 application.yml , 更新MySQL连接信息
- 项目访问路径: <http://localhost/admin>
- 管理员账号密码: admin/admin

2.1.4. 效果展示

1. 个人资料

用户名：admin
性别：男
联系手机：13647910412
电子邮箱：2434387555@qq.com
注册时间：2017-06-30 13:45:11
最后登录时间：2018-06-14 16:43:57
最后登录IP：0:0:0:0:0:1

登录时间	登录IP	操作系统	浏览器
2018-6-14 17:16	0.0.0.0.0.0:1	WINDOWS_81	CHROME
2018-6-14 17:14	0.0.0.0.0.0:1	WINDOWS_81	CHROME
2018-6-14 16:43	0.0.0.0.0.0:1	WINDOWS_81	CHROME
2018-6-14 16:27	0.0.0.0.0.0:1	WINDOWS_81	CHROME
2018-6-14 15:38	0.0.0.0.0.0:1	WINDOWS_81	CHROME
2018-6-14 15:09	0.0.0.0.0.0:1	WINDOWS_81	CHROME
2018-6-11 17:08	0.0.0.0.0.0:1	WINDOWS_81	CHROME
2017-7-14 15:54	192.168.1.88	WINDOWS_7	CHROME49
2017-7-14 15:50	192.168.1.88	WINDOWS_7	CHROME49

© 2018-2019 試驗-思伟

2. 菜单管理

菜单名称：新建菜单
类型：
菜单名称：
上级菜单：
排序号：
图标：

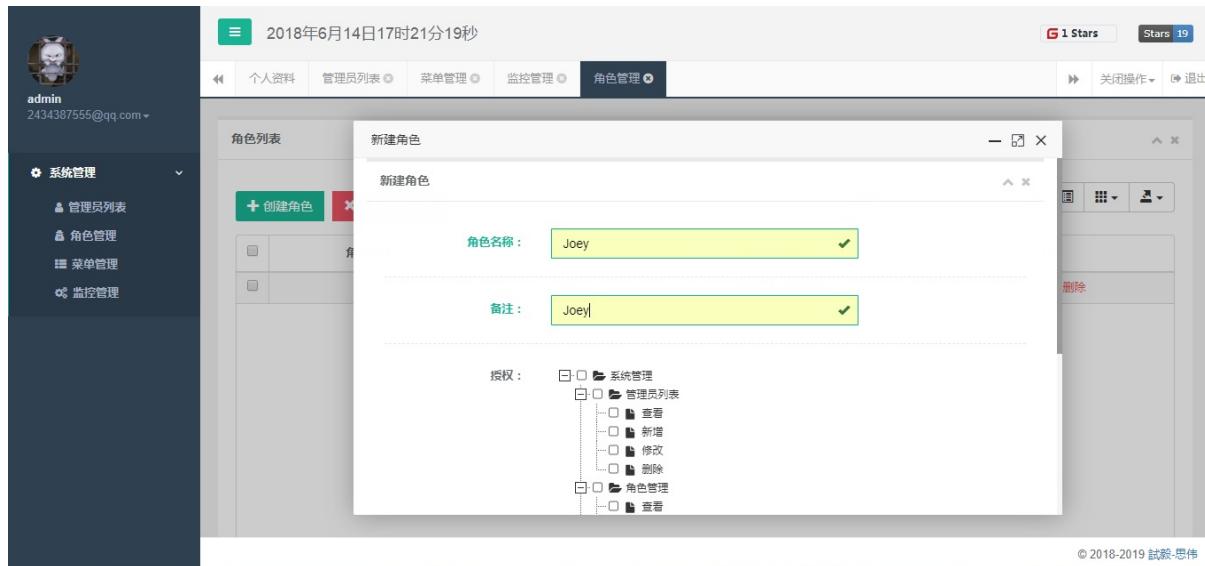
选择菜单

- 一级菜单
 - 系统管理
 - 管理员列表
 - 角色管理
 - 菜单管理
 - 监控管理

3. 菜单列表

菜单名称	上级菜单	菜单图标	菜单URL	授权标识	类型	排序号	操作
系统管理	-	⚙️			目录	0	编辑 删除
查看	管理员列表			sys.user.list,sys.user.info	按钮	0	编辑 删除
新增	管理员列表			sys.user.save,sys.role.select	按钮	0	编辑 删除
修改	管理员列表			sys.user.update,sys.role.select	按钮	0	编辑 删除
删除	管理员列表			sys.user.delete	按钮	0	编辑 删除
管理员列表	系统管理	👤	admin/sys/admin.html		菜单	1	编辑 删除
角色管理	系统管理	🔒	admin/sys/role.html		菜单	2	编辑 删除

© 2018-2019 試驗-思伟



Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-01 10:58:56

- 1. 微信小程序+微信管理后台+微信用户前台
 - 1.1. 项目结构:
 - 1.2. 技术选型:
 - 1.3. 本地部署
 - 1.4. 启动说明:

1. 微信小程序+微信管理后台+微信用户前台

GitHub: https://github.com/JoeYBling/hqc_mp

码云: https://gitee.com/zhouxiwei/hqc_mp

欢迎使用和Star支持, 如使用过程中碰到问题, 可以提出Issue, 我会尽力完善

1.1. 项目结构:

hqc_mini_app 微信小程序相关文件

```
hqc_mp
├── sql    -- 项目SQL语句
│
├── config -- 配置信息
│
├── controller -- 控制器
│   ├── map    -- 3D地图
│   ├── wx     -- 微信前台控制器
│   └── octopus -- 后台管理员控制器
│
├── service -- 业务逻辑接口
│   ├── impl   -- 业务逻辑接口实现类
│
├── dao    -- 数据访问接口
│
├── entity -- 数据持久化实体类
│
├── datasources -- 多数据源工具类
│
├── shiro   -- Shiro验证框架
│
├── task    -- Quartz定时任务
│
├── util    -- 项目所用的所有工具类
│   ├── FreeMarker -- 自定义FreeMarker标签
│
├── payforparking -- 停车付费相关接口WebService
│
├── weather  -- 第三方天气查询相关接口WebService
│
└── ws      -- 提供给小程序的相关接口WebService
```



1.2. 技术选型：

- 核心框架：Spring
- 安全框架：Apache Shiro
- 视图框架：Spring MVC
- 持久层框架：MyBatis MyBatis通用Mapper
- 缓存技术：Redis
- 定时器：Quartz
- 数据库连接池：Druid
- 日志管理：SLF4J、Log4j
- 模版技术：FreeMarker
- 页面交互：BootStrap、Layer等

1.3. 本地部署

- 创建数据库 hqc_mp，数据库编码为 UTF-8
- 执行 sql/hqc_mp.sql 文件，初始化数据
- 修改 db.properties，更新MySQL账号和密码
- 修改 redis.properties，更改Redis连接信息
- Eclipse、IDEA部署Tomcat，则可启动项目
- 项目访问路径：http://localhost:8080/hqc_mp
- 账号密码： admin/admin
- 前台登录用户： 13647910412 123

1.4. 启动说明：

项目依赖mysql、Redis服务。

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-01 10:58:56

- 1. 自制spring boot starter for fastjson
- 2. 简介
- 3. 版本基础
- 4. 如何使用
- 5. 参数配置
 - 5.1. 配置示例
 - 5.2. 配置说明
 - 5.2.1. 默认配置
 - 5.3. 贡献者

1. 自制spring boot starter for fastjson

GitHub: <https://github.com/JoeYBling/fastjson-spring-boot-starter>

码云: <https://gitee.com/zhouxiwei/fastjson-spring-boot-starter>

欢迎使用和Star支持, 如使用过程中碰到问题, 可以提出Issue, 我会尽力完善

2. 简介

该项目主要利用Spring Boot的自动化配置特性来实现快速的将fastjson引入spring boot使用fastjson解析框架。

- 源码地址
 - GitHub: <https://github.com/JoeYBling/fastjson-spring-boot-starter>
 - 码云: <https://gitee.com/zhouxiwei/fastjson-spring-boot-starter>
- 我的博客: <https://zhouxiwei.gitee.io/>

小工具一枚, 欢迎使用和Star支持, 如使用过程中碰到问题, 可以提出Issue, 我会尽力完善该Starter

3. 版本基础

- Spring Boot: 1.5.x
- FastJson: 1.2.15

4. 如何使用

在该项目的帮助下, 我们的Spring Boot可以轻松的引入fastjson, 主要做下面两个步骤:

- 在 `pom.xml` 中引入依赖:
 - 当前最新版本 10.10.10.RELEASE
- 由于spring-boot-starter-parent版本更新问题 spring-boot-starter-parent 2.0.0.RELEASE之前使用

```
<dependency>
```

```
<groupId>io.gitee.zhouxiwei</groupId>
<artifactId>fastjson-spring-boot-starter</artifactId>
<version>8.8.8.RELEASE</version>
</dependency>
```

- spring-boot-starter-parent 2.0.0.RELEASE之后使用

```
<dependency>
    <groupId>io.gitee.zhouxiwei</groupId>
    <artifactId>fastjson-spring-boot-starter</artifactId>
    <version>10.10.10.RELEASE</version>
</dependency>
```

默认情况下就能在springboot项目中使用fastjson

5. 参数配置

更细致的配置内容参考如下：

5.1. 配置示例

```
fastjson:
    enabled: true
    media-type: application/json;charset=UTF-8
    date-format: yyyy-MM-dd HH:mm:ss
```

5.2. 配置说明

5.2.1. 默认配置

```
fastjson:
    enabled: true  是否启用fastjson(默认为:true)
    media-type: application/json;charset=UTF-8 媒体类型(默认为:application/json;charset=UTF-8)
    date-format: yyyy-MM-dd HH:mm:ss  日期格式(默认为:yyyy-MM-dd HH:mm:ss)
```

8.8.8.RELEASE 新增：date-format 属性，且配置成功会打印控制台信息：starter for fastJson----fastJson init success.

5.3. 贡献者

- 思伟

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间：2019-07-01 10:58:56

- 1. 一个会动的简历
 - 1.1. 使用方法
 - 1.2. 部署方法

1. 一个会动的简历

GitHub: <https://github.com/JoeyBling/anires>

码云: <https://gitee.com/zhouxiwei/anires>

欢迎使用和Star支持, 如使用过程中碰到问题, 可以提出Issue, 我会尽力完善

本项目源于: <https://github.com/jirengu-inc/animating-resume>。向作者表示深深的敬意。

This is my resume

预览

1.1. 使用方法

```
git clone https://gitee.com/zhouxiwei/anires.git
cd anires
npm install
npm run dev
```

1.2. 部署方法

1. 编辑 config/index.js, 修改第 10 行的 assetsPublicPath, 值为 项目名/public 。如果你没有修改项目名 anires, 则可跳过此步骤。
2. 编译、上传

```
npm run build
git add .
git commit -m "update"
git push
```

3. 开启 GitHub Pages 功能, 把生成的目标文件 public/* 放在你的web服务器上。
4. 访问地址: <https://your-github-username.github.io/anires/public>

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-01 10:58:56

- 1. SpringBoot2.0更新记录

1. SpringBoot2.0更新记录

1、 Spring Boot 2.0 要求Java 版本必须8以上

2、配置属性的重定位

Old property	New property
server.context-path	server.servlet.context-path
server.context-parameters.*	server.servlet.context-parameters.*
server.jsp.class-name	server.servlet.jsp.class-name
server.jsp.init-parameters.*	server.servlet.jsp.init-parameters.*
server.jsp.registered	server.servlet.jsp.registered
server.servlet-path	server.servlet.path

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:26

- 1. SpringBoot集成Druid不支持多条SQL

1. SpringBoot集成Druid不支持多条SQL

在 DataSource 初始化Bean 添加

```
List<Filter> proxyFilters = new ArrayList<Filter>();
WallFilter statFilter = new WallFilter();
WallConfig config = new WallConfig();
config.setMultiStatementAllow(true); // 批量操作
statFilter.setConfig(config);
proxyFilters.add(statFilter);
druidDataSource.setProxyFilters(proxyFilters);
```

贴上自己的 Config

```
package com.wstro.config;

import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.sql.DataSource;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.boot.web.servlet.FilterRegistrationBean;
import org.springframework.boot.web.servlet.ServletRegistrationBean;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Primary;
import org.springframework.transaction.annotation.EnableTransactionManagement;

import com.alibaba.druid.filter.Filter;
import com.alibaba.druid.pool.DruidDataSource;
import com.alibaba.druid.support.http.StatViewServlet;
import com.alibaba.druid.support.http.WebStatFilter;
import com.alibaba.druid.wall.WallConfig;
import com.alibaba.druid.wall.WallFilter;

/**
 * Druid数据源配置
 *
 * @author Joey
 * @Email 2434387555@qq.com
 */
@Configuration
@EnableTransactionManagement
```

```
@EnableTransactionManagement // 启注解事务管理
public class DataSourceConfig {
    private Logger logger = LoggerFactory.getLogger(DataSourceConfig.class);

    @Value("${spring.datasource.url}")
    private String dbUrl;

    @Value("${spring.datasource.username}")
    private String username;

    @Value("${spring.datasource.password}")
    private String password;

    @Value("${spring.datasource.driverClassName}")
    private String driverClassName;

    @Value("${spring.datasource.initialSize}")
    private int initialSize;

    @Value("${spring.datasource.minIdle}")
    private int minIdle;

    @Value("${spring.datasource.maxActive}")
    private int maxActive;

    @Value("${spring.datasource.maxWait}")
    private int maxWait;

    @Value("${spring.datasource.timeBetweenEvictionRunsMillis}")
    private int timeBetweenEvictionRunsMillis;

    @Value("${spring.datasource.minEvictableIdleTimeMillis}")
    private int minEvictableIdleTimeMillis;

    @Value("${spring.datasource.validationQuery}")
    private String validationQuery;

    @Value("${spring.datasource.testWhileIdle}")
    private boolean testWhileIdle;

    @Value("${spring.datasource.testOnBorrow}")
    private boolean testOnBorrow;

    @Value("${spring.datasource.testOnReturn}")
    private boolean testOnReturn;

    @Value("${spring.datasource.poolPreparedStatements}")
    private boolean poolPreparedStatements;

    @Value("${spring.datasource.maxPoolPreparedStatementPerConnectionSize}")
    private int maxPoolPreparedStatementPerConnectionSize;

    @Value("${spring.datasource.filters}")
}
```

```

private String filters;

@Value("${spring.datasource.connectionProperties}")
private String connectionProperties;

/**
 * 注册DruidServlet
 *
 * @return ServletRegistrationBean
 */
@Bean
public ServletRegistrationBean druidServletRegistrationBean() {
    ServletRegistrationBean servletRegistrationBean = new ServletRegistrationBean();
    servletRegistrationBean.setServlet(new StatViewServlet());
    servletRegistrationBean.addUrlMappings("/druid/*");
    // 白名单:
    servletRegistrationBean.addInitParameter("allow", "127.0.0.1");
    // IP黑名单 (存在共同时, deny优先于allow) : 如果满足deny的话提示:Sorry, you are not
    // permitted to view this page.
    // 登录查看信息的账号密码.
    servletRegistrationBean.addInitParameter("loginUsername", "joey");
    servletRegistrationBean.addInitParameter("loginPassword", "jay");
    // 是否能够重置数据.
    servletRegistrationBean.addInitParameter("resetEnable", "false");
    return servletRegistrationBean;
}

/**
 * 注册DruidFilter拦截
 *
 * @return FilterRegistrationBean
 */
@Bean
public FilterRegistrationBean druidFilterRegistrationBean() {
    FilterRegistrationBean filterRegistrationBean = new FilterRegistrationBean();
    filterRegistrationBean.setFilter(new WebStatFilter());
    Map<String, String> initParams = new HashMap<String, String>();
    // 设置忽略请求
    initParams.put("exclusions", "*.js,*.gif,*.jpg,*.bmp,*.png,*.css,*.ico,/druid/*");
    filterRegistrationBean.setInitParameters(initParams);
    filterRegistrationBean.addUrlPatterns("/");
    return filterRegistrationBean;
}

/**
 * 配置DataSource
 *
 * @return DataSource
 * @throws SQLException

```

```

    */
    @Bean(initMethod = "init", destroyMethod = "close")
    @Primary
    public DataSource dataSource() throws SQLException {
        DruidDataSource druidDataSource = new DruidDataSource();
        druidDataSource.setUrl(dbUrl);
        druidDataSource.setUsername(username);
        druidDataSource.setPassword(password);
        druidDataSource.setDriverClassName(driverClassName);
        // configuration
        druidDataSource.setInitialSize(initialSize);
        druidDataSource.setMinIdle(minIdle);
        druidDataSource.setMaxActive(maxActive);
        druidDataSource.setMaxWait(maxWait);
        druidDataSource.setTimeBetweenEvictionRunsMillis(timeBetweenEvictionRunsMil
lis);
        druidDataSource.setMinEvictableIdleTimeMillis(minEvictableIdleTimeMillis);
        druidDataSource.setValidationQuery(validationQuery);
        druidDataSource.setTestWhileIdle(testWhileIdle);
        druidDataSource.setTestOnBorrow(testOnBorrow);
        druidDataSource.setTestOnReturn(testOnReturn);
        druidDataSource.setPoolPreparedStatements(poolPreparedStatements);
        druidDataSource.setMaxPoolPreparedStatementPerConnectionSize(maxPoolPrepare
dStatementPerConnectionString);
        try {
            List<Filter> proxyFilters = new ArrayList<Filter>();
            WallFilter statFilter = new WallFilter();
            WallConfig config = new WallConfig();
            config.setMultiStatementAllow(true); // 批量操作
            statFilter.setConfig(config);
            proxyFilters.add(statFilter);
            druidDataSource.setProxyFilters(proxyFilters);
            druidDataSource.setFilters(filters);
        } catch (SQLException e) {
            logger.error("druid configuration initialization filter", e);
        }
        druidDataSource.setConnectionProperties(connectionProperties);
        return druidDataSource;
    }
}

```

properties

```

#JDBC
# \u6570\u6362\u5E93\u8BBF\u95EE\u914D\u7F6E
# \u4E3B\u6570\u6362\u6E90\uFF0C\u9ED8\u8BA4\u7684
spring.datasource.type=com.alibaba.druid.pool.DruidDataSource
spring.datasource.driverClassName=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://192.168.1.88:3306/wstro?allowMultiQueries=true&
seUnicode=true&characterEncoding=UTF-8
spring.datasource.username=root
spring.datasource.password=root

```

```

# \u4E0B\u9762\u4E3A\u8FDE\u63A5\u6C60\u7684\u8865\u5145\u8BBE\u7F6E\uFF0C\u5E94\u7523\u5230\u4E0A\u9762\u6240\u6709\u6570\u636E\u6E90\u4E2D
# \u521D\u59CB\u5316\u5927\u5C0F\uFF0C\u6700\u5C0F\uFF0C\u6700\u5927
spring.datasource.initialSize=5
spring.datasource.minIdle=5
spring.datasource.maxActive=20

# \u914D\u7F6E\u83B7\u53D6\u8FDE\u63A5\u7B49\u5F85\u8D85\u65F6\u7684\u65F6\u95F4
spring.datasource.maxWait=60000

# \u914D\u7F6E\u95F4\u9694\u591A\u4E45\u624D\u8FDB\u884C\u4E00\u6B21\u68C0\u6D4B\uF0C\u68C0\u6D4B\u9700\u8981\u5173\u95ED\u7684\u7A7A\u95F2\u8FDE\u63A5\uFF0C\u5355\u4F4D\u662F\u6BEB\u79D2
spring.datasource.timeBetweenEvictionRunsMillis=60000

# \u914D\u7F6E\u4E00\u4E2A\u8FDE\u63A5\u5728\u6C60\u4E2D\u6700\u5C0F\u751F\u5B58\u7684\u65F6\u95F4\uFF0C\u5355\u4F4D\u662F\u6BEB\u79D2
spring.datasource.minEvictableIdleTimeMillis=300000
spring.datasource.validationQuery=SELECT 1 FROM DUAL
spring.datasource.testWhileIdle=true
spring.datasource.testOnBorrow=false
spring.datasource.testOnReturn=false

# \u6253\u5F00PSCache\uFF0C\u5E76\u4E14\u6307\u5B9A\u6BCF\u4E2A\u8FDE\u63A5\u4E0APSCache\u7684\u5927\u5C0F
spring.datasource.poolPreparedStatements=true
spring.datasource.maxPoolPreparedStatementPerConnectionSize=20

# \u914D\u7F6E\u76D1\u63A7\u7EDF\u8BA1\u62E6\u622A\u7684filters\uFF0C\u53BB\u6389\u540E\u76D1\u63A7\u754C\u9762sql\u65E0\u6CD5\u7EDF\u8BA1\uFF0C'wall'\u7528\u4E8E\u9632\u706B\u5899
spring.datasource.filters=stat,wall,log4j

# \u901A\u8FC7connectProperties\u5C5E\u6027\u6765\u6253\u5F00mergeSql\u529F\u80FD\uFF1B\u6162SQL\u8BB0\u5F55
spring.datasource.connectionProperties=druid.stat.mergeSql=true;druid.stat.slowSqlMillis=5000;druid.stat.logSlowSql=true;

# \u5408\u5E76\u591A\u4E2ADruidDataSource\u7684\u76D1\u63A7\u6570\u636E
#spring.datasource.useGlobalDataSourceStat=true

```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:26

- 1. SpringBoot设置默认首页

1. SpringBoot设置默认首页

```
@Configuration
public class DefaultView extends WebMvcConfigurerAdapter {

    @Override
    public void addViewControllers(ViewControllerRegistry registry) {
        super.addViewControllers(registry);
        //主页
        registry.addViewController("/").setViewName("forward:/index");
    }

}
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:26

- 1. Ajax请求设置Type为Form-Data

1. Ajax请求设置Type为Form-Data

在你的Ajax请求中加上

```
headers : {  
    'Content-Type' : 'application/x-www-form-urlencoded'  
},
```

示例：

```
$.ajax({  
    type : "POST",  
    url : "../user/repwd",  
    headers : {  
        'Content-Type' : 'application/x-www-form-urlencoded'  
    },  
    data : data,  
    dataType : "json",  
});
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间： 2019-07-02 13:44:26

- 1. 前端开发相关笔记

1. 前端开发相关笔记

- 记录前端开发过程中容易出错的地方，及浏览器的一些兼容解决方法

| 前端开发相关笔记预览：<http://zhousiwei.gitee.io/mybook/web/>

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-03 14:11:35

- 1. 移动Web开发技巧汇总

- 1.1. META相关

- 1.1.1. 1. 添加到主屏后的标题 (IOS)
 - 1.1.2. 2. 启用 WebApp 全屏模式 (IOS)
 - 1.1.3. 3. 百度禁止转码
 - 1.1.4. 4. 设置状态栏的背景颜色 (IOS)
 - 1.1.5. 5. 移动端手机号识别 (IOS)
 - 1.1.6. 6. 移动端邮箱识别 (Android)
 - 1.1.7. 7. 添加智能 App 广告条 Smart App Banner (IOS 6+ Safari)
 - 1.1.8. 8. IOS Web app启动动画
 - 1.1.9. 9. 添加到主屏后的APP图标
 - 1.1.10. 10. 优先使用最新版本 IE 和 Chrome
 - 1.1.11. 11.viewport模板

- 1.2. 常见问题

- 1.2.1. 1、移动端如何定义字体font-family
 - 1.2.2. 2、移动端字体单位font-size选择px还是rem
 - 1.2.3. 3、移动端touch事件(区分webkit 和 winphone)
 - 1.2.4. 4、移动端click屏幕产生200-300 ms的延迟响应
 - 1.2.5. 5、什么是Retina显示屏，带来了什么问题
 - 1.2.6. 6、ios系统中元素被触摸时产生的半透明灰色遮罩怎么去掉
 - 1.2.7. 7、部分android系统中元素被点击时产生的边框怎么去掉
 - 1.2.8. 8、winphone系统a、input标签被点击时产生的半透明灰色背景怎么去掉
 - 1.2.9. 9、webkit表单元素的默认外观怎么重置
 - 1.2.10. 10、webkit表单输入框placeholder的颜色值能改变么
 - 1.2.11. 11、webkit表单输入框placeholder的文字能换行么
 - 1.2.12. 12. 关闭iOS键盘首字母自动大写
 - 1.2.13. 13. 关闭iOS输入自动修正
 - 1.2.14. 14. 禁止文本缩放
 - 1.2.15. 15. 移动端如何清除输入框内阴影
 - 1.2.16. 16. 快速回弹滚动
 - 1.2.17. 17. 移动端禁止选中内容
 - 1.2.18. 18. 移动端取消touch高亮效果
 - 1.2.19. 19. 如何禁止保存或拷贝图像 (IOS)
 - 1.2.20. 20.模拟按钮hover效果
 - 1.2.21. 21.屏幕旋转的事件和样式
 - 1.2.22. 22.audio元素和video元素在ios和andriod中无法自动播放
 - 1.2.23. 23.摇一摇功能
 - 1.2.24. 24.手机拍照和上传图片
 - 1.2.25. 25. 消除transition闪屏
 - 1.2.26. 26. android 上去掉语音输入按钮

- 1.3. 框架

- 1.3.1. 1. 移动端基础框架
 - 1.3.2. 2. 滑屏框架
 - 1.3.3. 3.瀑布流框架

- [1.4. 工具推荐](#)
- [1.5. 关于我](#)

1. 移动Web开发技巧汇总

1.1. META相关

1.1.1. 1. 添加到主屏后的标题 (IOS)

```
<meta name="apple-mobile-web-app-title" content="标题">
```

1.1.2. 2. 启用 WebApp 全屏模式 (IOS)

当网站添加到主屏幕后再点击进行启动时，可隐藏地址栏（从浏览器跳转或输入链接进入并没有此效果）

```
<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="apple-touch-fullscreen" content="yes" />
```

1.1.3. 3. 百度禁止转码

通过百度手机打开网页时，百度可能会对你的网页进行转码，往你页面贴上它的广告，非常之恶心。不过我们可以通过这个meta标签来禁止它

```
<meta http-equiv="Cache-Control" content="no-siteapp" />
```

- [百度SiteApp转码声明](#)

1.1.4. 4. 设置状态栏的背景颜色 (IOS)

设置状态栏的背景颜色，只有在 "apple-mobile-web-app-capable" content="yes" 时生效

```
<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />
```

content 参数：

- default : 状态栏背景是白色。
- black : 状态栏背景是黑色。
- black-translucent : 状态栏背景是半透明。如果设置为 default 或 black ,网页内容从状态栏底部开始。如果设置为 black-translucent ,网页内容充满整个屏幕，顶部会被状态栏遮挡。

1.1.5. 5. 移动端手机号码识别 (IOS)

在 iOS Safari (其他浏览器和Android均不会) 上会对那些看起来像是电话号码的数字处理为电话链接，比如：

- 7位数字，形如：1234567
- 带括号及加号的数字，形如：(+86)123456789
- 双连接线的数字，形如：00-00-00111
- 11位数字，形如：13800138000

可能还有其他类型的数字也会被识别。我们可以通过如下的meta来关闭电话号码的自动识别：

```
<meta name="format-detection" content="telephone=no" />
```

- 开启电话功能

```
<a href="tel:123456">123456</a>
```

- 开启短信功能：

```
<a href="sms:123456">123456</a>
```

1.1.6. 6. 移动端邮箱识别 (Android)

与电话号码的识别一样，在安卓上会对符合邮箱格式的字符串进行识别，我们可以通过如下的meta来管别邮箱的自动识别：

```
<meta content="email=no" name="format-detection" />
```

同样地，我们也可以通过标签属性来开启长按邮箱地址弹出邮件发送的功能：

```
<a href="mailto:dooyoe@gmail.com">dooyoe@gmail.com</a>
```

1.1.7. 7. 添加智能 App 广告条 Smart App Banner (IOS 6+ Safari)

```
<meta name="apple-itunes-app" content="app-id=myAppStoreID, affiliate-data=myAffiliateData, app-argument=myURL">
```

1.1.8. 8. IOS Web app启动动画

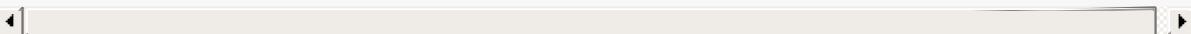
由于iPad的启动画面是不包括状态栏区域的。所以启动图片需要减去状态栏区域所对应的方向上的20px大小，相应地在retina设备上要减去40px的大小

```
<!-- iPhone -->
<link href="apple-touch-startup-image-320x460.png" media="(device-width: 320px)" rel="apple-touch-startup-image">
```

```

<!-- iPhone (Retina) -->
<link href="apple-touch-startup-image-640x960.png" media="(device-width: 320px) and
(-webkit-device-pixel-ratio: 2)" rel="apple-touch-startup-image">
<!-- iPad (portrait) -->
<link href="apple-touch-startup-image-768x1004.png" media="(device-width: 768px) an
d (orientation: portrait)" rel="apple-touch-startup-image">
<!-- iPad (landscape) -->
<link href="apple-touch-startup-image-748x1024.png" media="(device-width: 768px) an
d (orientation: landscape)" rel="apple-touch-startup-image">
<!-- iPad (Retina, portrait) -->
<link href="apple-touch-startup-image-1536x2008.png" media="(device-width: 1536px)
and (orientation: portrait) and (-webkit-device-pixel-ratio: 2)" rel="apple-touch-s
tartup-image">
<!-- iPad (Retina, landscape) -->
<link href="apple-touch-startup-image-2048x1496.png" media="(device-width: 1536px)
and (orientation: landscape) and (-webkit-device-pixel-ratio: 2)" rel="apple-touch
-startup-image">

```



(landscape: 横屏 | portrait: 竖屏)

1.1.9. 9. 添加到主屏后的APP图标

指定web app添加到主屏后的图标路径，有两种略微不同的方式：

```

<!-- 设计原图 -->
<link href="short_cut_114x114.png" rel="apple-touch-icon-precomposed">
<!-- 添加高光效果 -->
<link href="short_cut_114x114.png" rel="apple-touch-icon">

```

图标尺寸：

- 可通过指定 `size` 属性来为不同的设备提供不同的图标（但通常来说，我们只需提供一个114 x 114 pixels大小的图标即可）
- 官方说明如下：

Create different sizes of your app icon for different devices. If you're creating a universal app, you need to supply app icons in all four sizes.
For iPhone and iPod touch both of these sizes are required:

57 x 57 pixels

114 x 114 pixels (high resolution)

For iPad, both of these sizes are required:

72 x 72 pixels

144 x 144 (high resolution)

1.1.10. 10. 优先使用最新版本 IE 和 Chrome

```
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
```

1.1.11. 11.viewport模板

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no" name="viewport">
<meta content="yes" name="apple-mobile-web-app-capable">
<meta content="black" name="apple-mobile-web-app-status-bar-style">
<meta content="telephone=no" name="format-detection">
<meta content="email=no" name="format-detection">
<title>标题</title>
<link rel="stylesheet" href="index.css">
</head>
<body>
这里开始内容
</body>
</html>
```

1.2. 常见问题

1.2.1. 1、移动端如何定义字体font-family

三大手机系统的字体：

ios 系统

- 默认中文字体是Heiti SC
- 默认英文字体是Helvetica
- 默认数字字体是HelveticaNeue
- 无微软雅黑字体

android 系统

- 默认中文字体是Droidsansfallback
- 默认英文和数字字体是Droid Sans
- 无微软雅黑字体

winphone 系统

- 默认中文字体是Dengxian(方正等线体)
- 默认英文和数字字体是Segoe
- 无微软雅黑字体

各个手机系统有自己的默认字体，且都不支持微软雅黑。如无特殊需求，手机端无需定义中文字体，使用系统默认 英文字体和数字字体可使用 Helvetica，三种系统都支持

```
* 移动端定义字体的代码 */
body{font-family:Helvetica;}
```

1.2.2. 2、移动端字体单位font-size选择px还是rem

- 对于只需要适配手机设备，使用px即可
- 对于需要适配各种移动设备，使用rem，例如只需要适配iPhone和iPad等分辨率差别比较挺大的设备

rem配置参考：

```
html {font-size:10px}
@media screen and (min-width:480px) and (max-width:639px) {
    html {
        font-size: 15px
    }
}
@media screen and (min-width:640px) and (max-width:719px) {
    html {
        font-size: 20px
    }
}
@media screen and (min-width:720px) and (max-width:749px) {
    html {
        font-size: 22.5px
    }
}
@media screen and (min-width:750px) and (max-width:799px) {
    html {
        font-size: 23.5px
    }
}
@media screen and (min-width:800px) and (max-width:959px) {
    html {
        font-size: 25px
    }
}
@media screen and (min-width:960px) and (max-width:1079px) {
    html {
        font-size: 30px
    }
}
@media screen and (min-width:1080px) {
    html {
        font-size: 32px
    }
}
```

1.2.3. 3、移动端touch事件(区分webkit 和 winphone)

- 当用户手指放在移动设备在屏幕上滑动会触发的touch事件

以下支持 webkit

- touchstart ——当手指触碰屏幕时候发生。不管当前有多少只手指
- touchmove ——当手指在屏幕上滑动时连续触发。通常我们再滑屏页面，会调用event的 preventDefault()可以阻止默认情况的发生：阻止页面滚动
- touchend ——当手指离开屏幕时触发
- touchcancel ——系统停止跟踪触摸时候会触发。例如在触摸过程中突然页面alert()一个提示框，此时会触发该事件，这个事件比较少用

以下支持 winphone 8

- MSPointerDown ——当手指触碰屏幕时候发生。不管当前有多少只手指
- MSPointerMove ——当手指在屏幕上滑动时连续触发。通常我们再滑屏页面，会调用css 的 html{-ms-touch-action: none;} 可以阻止默认情况的发生：阻止页面滚动
- MSPointerUp ——当手指离开屏幕时触发

1.2.4. 4、移动端click屏幕产生200-300 ms的延迟响应

- fastclick可以解决在手机上点击事件的300ms延迟
- zepto的touch模块，tap事件也是为了解决在click的延迟问题

触摸事件的响应顺序

- ontouchstart
- ontouchmove
- ontouchend
- onclick

解决300ms延迟的问题，也可以通过绑定 ontouchstart 事件，加快对事件的响应

1.2.5. 5、什么是Retina显示屏，带来了什么问题

retina：一种具备超高像素密度的液晶屏，同样大小的屏幕上显示的像素点由1个变为多个，如在同样带下的屏幕上，苹果设备的retina显示屏中，像素点1个变为4个

在高清显示屏中的位图被放大，图片会变得模糊，因此移动端的视觉稿通常会设计为传统PC的2倍
那么，前端的应对方案是：

设计稿切出来的图片长宽保证为偶数，并使用 background-size 把图片缩小为原来的1/2

```
//例如图片宽高为：200px*200px，那么写法如下
.css{width:100px;height:100px;background-size:100px 100px;}
```

其它元素的取值为原来的1/2，例如视觉稿40px的字体，使用样式的写法为20px

```
.css{font-size:20px}
```

1.2.6. 6、ios系统中元素被触摸时产生的半透明灰色遮罩怎么去掉

ios用户点击一个链接，会出现一个半透明灰色遮罩，如果想要禁用，可设置 `-webkit-tap-highlight-color` 的 `alpha` 值为0，也就是属性值的最后一位设置为0就可以去除半透明灰色遮罩

```
a,button,input,textarea{-webkit-tap-highlight-color: rgba(0,0,0,0);}
```

1.2.7. 7、部分android系统中元素被点击时产生的边框怎么去掉

android用户点击一个链接，会出现一个边框或者半透明灰色遮罩，不同生产商定义出来的效果不一样，可设置 `-webkit-tap-highlight-color` 的 `alpha` 值为0去除部分机器自带的效果

```
a,button,input,textarea{
    -webkit-tap-highlight-color: rgba(0,0,0,0);
    -webkit-user-modify:read-write-plaintext-only;
}
```

- `-webkit-user-modify` 有个副作用，就是输入法不再能够输入多个字符</p>

1.2.8. 8、winphone系统a、input标签被点击时产生的半透明灰色背景怎么去掉

```
<meta name="msapplication-tap-highlight" content="no">
```

1.2.9. 9、webkit表单元素的默认外观怎么重置

```
.css{-webkit-appearance:none;}
```

1.2.10. 10、webkit表单输入框placeholder的颜色值能改变么

```
input::-webkit-input-placeholder{color:#AAAAAA;}
input:placeholder-shown::-webkit-input-placeholder{color:#EEEEEE;}
```

1.2.11. 11、webkit表单输入框placeholder的文字能换行么

ios可以， android不行~

1.2.12. 12. 关闭iOS键盘首字母自动大写

在iOS中，默认情况下键盘是开启首字母大写的功能的，如果关闭这个功能，可以这样：

```
<input type="text" autocapitalize="off" />
```

1.2.13. 13. 关闭iOS输入自动修正

和英文输入默认自动首字母大写那样，IOS还做了一个功能，默认输入法会开启自动修正输入内容，这样的话，用户经常要操作两次。如果不希望开启此功能，我们可以通过input标签属性来关闭掉：

```
<input type="text" autocorrect="off" />
```

1.2.14. 14. 禁止文本缩放

当移动设备横竖屏切换时，文本的大小会重新计算，进行相应的缩放，当我们不需要这种情况时，可以选择禁止：

```
html {
    -webkit-text-size-adjust: 100%;
}
```

- 需要注意的是，PC端的该属性已经被移除，该属性在移动端要生效，必须设置 `meta viewport`

1.2.15. 15. 移动端如何清除输入框内阴影

在iOS上，输入框默认有内部阴影，但无法使用 `box-shadow` 来清除，如果不需要阴影，可以这样关闭：

```
input, textarea {
    border: 0; /* 方法1 */
    -webkit-appearance: none; /* 方法2 */
}
```

1.2.16. 16. 快速回弹滚动

在iOS上如果你想让一个元素拥有像 Native 的滚动效果，你可以这样做：

```
.xxx {
    overflow: auto; /* auto | scroll */
    -webkit-overflow-scrolling: touch;
}
```

- PS：iScroll用过之后感觉不是很好，有一些诡异的bug，这里推荐另外一个 `iDangerous Swiper`，这个插件集成了滑屏滚动的强大功能（支持3D），而且还有回弹滚动的内置滚动条，官方地

址: <http://www.idangero.us/sliders/swiper/index.php>

1.2.17. 17. 移动端禁止选中内容

如果你不想用户可以选中页面中的内容, 那么你可以在css中禁掉:

```
.user-select-none {
    -webkit-user-select: none; /* Chrome all / Safari all */
    -moz-user-select: none;     /* Firefox all (移动端不需要) */
    -ms-user-select: none;     /* IE 10+ */
}
```

1.2.18. 18. 移动端取消touch高亮效果

在做移动端页面时, 会发现所有a标签在触发点击时或者所有设置了伪类 :active 的元素, 默认都会在激活状态时, 显示高亮框, 如果不想要这个高亮, 那么你可以通过css以下方法来进行全局的禁止:

```
html {
    -webkit-tap-highlight-color: rgba(0, 0, 0, 0);
}
```

- 但这个方法在三星的机子上无效, 有一种妥协的方法是把页面非真实跳转链接的a标签换成其它标签, 可以解决这个问题。

1.2.19. 19. 如何禁止保存或拷贝图像 (IOS)

通常当你在手机或者pad上长按图像 img , 会弹出选项 存储图像 或者 拷贝图像, 如果你不想让用户这么操作, 那么你可以通过以下方法来禁止:

```
img { -webkit-touch-callout: none; }
```

1.2.20. 20. 模拟按钮hover效果

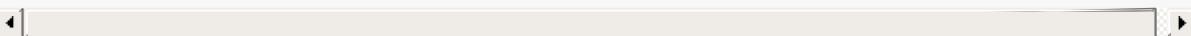
移动端触摸按钮的效果, 可明示用户有些事情正要发生, 是一个比较好体验, 但是移动设备中并没有鼠标指针, 使用css的hover并不能满足我们的需求, 还好国外有个激活css的active效果, 代码如下:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" name="viewport">
<meta content="yes" name="apple-mobile-web-app-capable">
<meta content="black" name="apple-mobile-web-app-status-bar-style">
```

```

<meta content="telephone=no" name="format-detection">
<meta content="email=no" name="format-detection">
<style type="text/css">
a{-webkit-tap-highlight-color: rgba(0,0,0,0);}
.btn-blue{display: block; height: 42px; line-height: 42px; text-align: center; border-radius: 4px; font-size: 18px; color: #FFFFFF; background-color: #4185F3;}
.btn-blue:active{background-color: #357AE8;}
</style>
</head>
<body>
<div class="btn-blue">按钮</div>
<script type="text/javascript">
document.addEventListener("touchstart", function(){}, true)
</script>
</body>
</html>

```



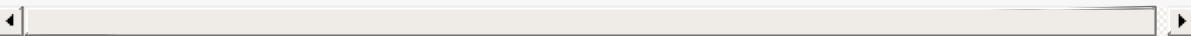
- 兼容性ios5+、部分android 4+、winphone 8

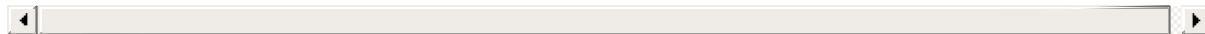
要做到全兼容的办法，可通过绑定 ontouchstart 和 ontouchend 来控制按钮的类名

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no" name="viewport">
<meta content="yes" name="apple-mobile-web-app-capable">
<meta content="black" name="apple-mobile-web-app-status-bar-style">
<meta content="telephone=no" name="format-detection">
<meta content="email=no" name="format-detection">
<style type="text/css">
a{-webkit-tap-highlight-color: rgba(0,0,0,0);}
.btn-blue{display: block; height: 42px; line-height: 42px; text-align: center; border-radius: 4px; font-size: 18px; color: #FFFFFF; background-color: #4185F3;}
.btn-blue-on{background-color: #357AE8;}
</style>
</head>
<body>
<div class="btn-blue">按钮</div>
<script type="text/javascript">
var btnBlue = document.querySelector(".btn-blue");
btnBlue.ontouchstart = function(){
    this.className = "btn-blue btn-blue-on"
}
btnBlue.ontouchend = function(){
    this.className = "btn-blue"
}
</script>
</body>
</html>

```





1.2.21. 21.屏幕旋转的事件和样式

事件: `window.orientation`, 取值: 正负90表示横屏模式、0和180表现为竖屏模式

```
window.onorientationchange = function(){
    switch(window.orientation){
        case -90:
        case 90:
            alert("横屏:" + window.orientation);
        case 0:
        case 180:
            alert("竖屏:" + window.orientation);
            break;
    }
}
```

样式

```
//竖屏时使用的样式
@media all and (orientation:portrait) {
    .css{}
}

//横屏时使用的样式
@media all and (orientation:landscape) {
    .css{}
}
```

1.2.22. 22.audio元素和video元素在ios和andriod中无法自动播放

应对方案: 触屏即播

```
($('html').one('touchstart',function(){
    audio.play()
}))
```

1.2.23. 23.摇一摇功能

HTML5 deviceMotion: 封装了运动传感器数据的事件, 可以获取手机运动状态下的运动加速度等数据。

1.2.24. 24.手机拍照和上传图片

`<input type="file">` 的 accept 属性

```
<!-- 选择照片 -->
```

```
<input type="file" accept="image/*">
<!-- 选择视频 -->
<input type="file" accept="video/*">
```

使用总结：

- ios 有拍照、录像、选取本地图片功能
- 部分android只有选取本地图片功能
- winphone不支持
- input控件默认外观丑陋

1.2.25. 25. 消除transition闪屏

```
.css{
    /*设置内嵌的元素在 3D 空间如何呈现：保留 3D*/
    -webkit-transform-style: preserve-3d;
    /* (设置进行转换的元素的背面在面对用户时是否可见：隐藏) */
    -webkit-backface-visibility: hidden;
}
```

开启硬件加速

- 解决页面闪白
- 保证动画流畅

```
.css {
    -webkit-transform: translate3d(0, 0, 0);
    -moz-transform: translate3d(0, 0, 0);
    -ms-transform: translate3d(0, 0, 0);
    transform: translate3d(0, 0, 0);
}
```

设计高性能CSS3动画的几个要素

- 尽可能地使用合成属性transform和opacity来设计CSS3动画
- 不使用position的left和top来定位
- 利用translate3D开启GPU加速

1.2.26. 26. android 上去掉语音输入按钮

```
input::-webkit-input-speech-button {display: none}
```

1.3. 框架

1.3.1. 1. 移动端基础框架

- [zepto.js](#) 语法与jquery几乎一样，会jquery基本会zepto~
- [iscroll.js](#) 解决页面不支持弹性滚动，不支持fixed引起的问题~ 实现下拉刷新，滑屏，缩放等功能~
- [underscore.js](#) 该库提供了一整套函数式编程的实用功能，但是没有扩展任何JavaScript内置对象。
- [fastclick](#) 加快移动端点击响应时间
- [animate.css](#) CSS3动画效果库

1.3.2. 2. 滑屏框架

适合上下滑屏、左右滑屏等滑屏切换页面的效果

- [slip.js](#)
- [iSlider.js](#)
- [fullpage.js](#)

1.3.3. 3.瀑布流框架

- [masonry](#)

1.4. 工具推荐

- [caniuse](#) 各浏览器支持html5属性查询
- [paletton](#)调色搭配

1.5. 关于我

- 个人博客：<https://zhousiwei.gitee.io/>
- 开发笔记：<https://zhousiwei.gitee.io/myBook>

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-03 14:11:36

- 1. 前端开发注意事项

1. 前端开发注意事项

1、清除float浮动

```
.clearfix {  
    *zoom: 1;  
}  
.clearfix:before, .clearfix:after {  
    content: ' ';  
    display: table;  
}  
.clearfix:after {  
    clear: both;  
}
```

2、radio,checkbox写法 (：这种写法点后面的文字也可选中前面的选择框)

```
<input type="radio" id="idRadio" /> <label for="idRadio" >今天</label>  
<input type="checkbox" id="idCheck" /> <label for="idCheck" >今天</label>
```

3、在手机端去掉以下标签的各种自带样式

```
a,select,span,i,input {  
    -webkit-tap-highlight-color: transparent; //无色  
    -webkit-tap-highlight-color: rgba(0, 0, 0, 0); //这个地方的设置最好是用rgba  
    -webkit-appearance: none;  
    outline: none;  
    border: 0;  
    line-height: normal;  
    background: none;  
}
```

4、解决手机web字号不一

```
body{ -webkit-text-size-adjust:none; }
```

5、去除ie11输入框的X

```
input[type=text]::-ms-clear { display: none; width : 0; height: 0; }  
input[type=text]::-ms-reveal { display: none; width : 0; height: 0; }
```

6、去掉chrome input自动填充颜色

```
input:-webkit-autofill {  
    box-shadow: 0px 0px 0px 1000px #ffff inset !important;  
}
```

7、清除chrome下input自动记住密码

```
<input style="display: none">
<input type="password" autocomplete="off" placeholder="请输入密码">
```

8、hack的使用

```
.box{
    background-color:#f1ee18; /*所有识别*/
    background-color:#00deff\9; /*'\'9'IE6、7、8、9、10识别*/
    background-color:#00deff\0; /*'\0'IE8、9、10识别*/
    background-color:#00deff\9\0; /*'\0'IE9、10识别*/
    +background-color:#a200ff; /*'+IE6、7识别*/
    _background-color:#1e0bd1; /*'_IE6识别*/
}
IE6能识别*, 不能识别 !important;
IE7能识别*, 能识别!important;
FF不能识别*, 但能识别!important;
"" IE6、IE7可以识别.IE8、FireFox不能.

:root #test { background-color:purple\9; } :root是给ie9的, 网上流传了个版本是 :root #te
st { background- color:purple\0;}, 这个, 新版opera也认识, 所以经笔者反复验证最终ie9特有的为
:root 选择符 {属性\9;};
@media all and (min-width:0px){ #test {background-color:black\0;} } 这个是老是跟ie抢着
认\0的神奇的opera, 必须加个\0, 不然firefox, chrome, safari也都认识。。
@media screen and (-webkit-min-device-pixel-ratio:0){ #test {background-color:gray;} }
}最后这个是浏览器新贵chrome和safari的。

.test:lang(zh-cn){color:#f00; /* For IE8+ and not IE */}
.test:nth-child(1){color:#0ff;} /* For IE9+ and not IE */

"- "减号是IE6专有的hack
"\9" IE6/IE7/IE8/IE9/IE10都生效
"\0" IE8/IE9/IE10都生效, 是IE8/9/10的hack
"\9\0" 只对IE9/IE10生效, 是IE9/10的hack

&lt;!--[if IE]&gt; 这段文字只在IE浏览器显示&lt;![endif]--&gt;;
&lt;!--[if IE 6]&gt; 这段文字只在IE6浏览器显示&lt;![endif]--&gt;;
&lt;!--[if gte IE 6]&gt; 这段文字只在IE6以上(包括)版本IE浏览器显示&lt;![endif]--&gt;;
&lt;!--[if !IE 8]&gt; 这段文字在非IE8浏览器显示&lt;![endif]--&gt;;
&lt;!--[if !IE]&gt; 这段文字只在非IE浏览器显示&lt;![endif]--&gt;;
```

9、web网站屏幕适配CSS写法

```
@media screen and (max-width: 800px) {
    body{
        background: #ccc;
    }
}
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-03 14:11:36

- 1. nodejs项目配置文件介绍

1. nodejs项目配置文件介绍

在nodejs项目开发时几个主要的配置文件写法

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-03 14:11:36

- 1. gulpfilejs
 - 1.1.1. 项目运行需要的配置文件
 - 1.1.2. 把jade编译成html，编译过程会寻找对应名字的json文件
 - 1.1.3. 监听jade的任务需要跟编译的分开
 - 1.1.4. 把scss编译成css
 - 1.1.5. 压缩js和css
 - 1.1.6. 启动服务器，进行监听

1. gulpfilejs

1.1.1. 项目运行需要的配置文件

- 根据每个人不同的需求自己添加删除

```
var gulp = require('gulp');
var browserSync = require('browser-sync');
var reload = browserSync.reload;
var plumber = require('gulp-plumber');
var sass = require('gulp-sass');
var path = require('path');
var fs = require('fs');
var data = require('gulp-data');
var jade = require('gulp-jade');
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
var uglifycss = require('gulp-uglifycss');
var rename = require("gulp-rename");
var gulpCopy = require('gulp-file-copy');
```

1.1.2. 把jade编译成html，编译过程会寻找对应名字的json文件

```
gulp.task('templates', function() {
  return gulp.src('srcs/jade/*.jade')
    .pipe(plumber())
    .pipe(data(function(file) {
      var json = JSON.parse(fs.readFileSync('srcs/json/' + path.basename(file.path, '.jade') + '.json'));
      return json;
    }))
    .pipe(jade({
      pretty: true
    }))
    .pipe(gulp.dest('builds'));
});
```

1.1.3. 监听jade的任务需要跟编译的分开

命令语句： `gulp jade-watch` 只运行这个命令是执行 jade 编译

```
gulp.task('jade-watch', ['templates'], reload);
gulp.task('scss-watch', ['compass'], reload);
```

1.1.4. 把scss编译成css

```
gulp.task('sass', function() {
  return gulp.src("srcs/scss/*.scss")
    .pipe(plumber({
      errorHandler: function(error) {
        console.log(error.message);
        this.emit('end');
      }
    }))
    .pipe(sass())
    .on('error', function(err) {})
    .pipe(gulp.dest("builds/css"))
    .pipe(reload({
      stream: true
    }));
});
```

1.1.5. 压缩js和css

```
gulp.task('jimin', function() {
  return gulp.src('builds/js/*.js')
    // .pipe(concat('main.js')) 把所有的js合并压缩到main.js里面
    .pipe(uglify({
      mangle: {except: ["require"]} // 压缩时过滤掉不想被压缩的关键词
    }))
    .pipe(rename(function(path) {
      path.basename
    }))
    .pipe(gulp.dest('builds/compressed/js'));
});

gulp.task('cssmin', function() {
  gulp.src('builds/css/*.css')
    .pipe(uglifycss())
    .pipe(rename(function(path) {
      path.basename += '.min'
    }))
    .pipe(gulp.dest('builds/compressed/css'));
});
gulp.task('compress', ['jsmin', 'cssmin']);
```

单独运行命令：

- `gulp jimin` 只执行js压缩
- `gulp cssmin` 只执行css压缩
- `gulp compress` js和css压缩都执行

注：在不运行gulp时，开户config.rb文件内的 output_style = :compressed 运行compass watch 同样时压缩css

1.1.6. 启动服务器，进行监听

运行命令：`gulp` 默认启动nodejs项目服务器，时时更新 sass 和 jade 编译

```
gulp.task('default', ['sass', 'templates','jimin','cssmin'], function() {  
  
  browserSync({  
    server: 'builds'  
  });  
  
  gulp.watch('srcs/scss/*.scss', ['sass']);  
  gulp.watch('srcs/jade/**/*.jade', ['jade-watch']);  
  gulp.watch('srcs/json/*.json', ['jade-watch']);  
});
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-03 14:11:36

- 1. package.json

1. package.json

这个文件只需要注意配置 `dependencies` 例:

```
"browser-sync": "^2.8.2",
"fs": "0.0.2",
"gulp": "^3.9.0",
"gulp-data": "^1.2.0",
"gulp-jade": "^1.1.0",
"gulp-plumber": "^1.0.1",
"gulp-rename": "^1.2.2",
"gulp-sass": "^2.0.4",
"gulp-concat": "v2.6.0",
"gulp-uglify": "^1.2.0",
"gulp-uglifycss": "^1.0.4",
"gulp-wrap-amd": "^0.5.0",
"path": "^0.11.14",
"gulp-file-copy": "v0.0.1"
```

所需文件版本更新可以查看: <https://www.npmjs.com/>

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-03 14:11:36

- 1. config.rb

1. config.rb

配置好这个文件里的文件路径，如：

```
http_path = "/"
css_dir = "builds/css"
sass_dir = "srcs/scss"
images_dir = "builds/images"
javascripts_dir = "builds/js"
```

直接运行命令语句： compass watch 可编译 scss

放开代码： # output_style = :compressed; 去掉前面的“#”这时css能被时时压塑

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-03 14:11:35

- 1. 移动Web开发技巧汇总
- 2. .gitignore

1. 移动Web开发技巧汇总

2. .gitignore

这个文件是设置忽略不想上传到 git 上的文件

把不想上传的文件名称直接写进这个文件内就可以

示例:

```
# Logs
logs
*.log
npm-debug.log*
yarn-debug.log*
yarn-error.log*

# Runtime data
pids
*.pid
*.seed
*.pid.lock

# Directory for instrumented libs generated by jscoverage/JSCover
lib-cov

# Coverage directory used by tools like istanbul
coverage

# nyc test coverage
.nyc_output

# Grunt intermediate storage (http://gruntjs.com/creating-plugins#storing-task-files)
.grunt

# Bower dependency directory (https://bower.io/)
bower_components

# node-waf configuration
.lock-wscript

# Compiled binary addons (http://nodejs.org/api/addons.html)
build/Release

# Dependency directories
node_modules/
```

```
jspm_packages/  
  
# Typescript v1 declaration files  
 typings/  
  
# Optional npm cache directory  
.npm  
  
# Optional eslint cache  
.eslintcache  
  
# Optional REPL history  
.node_repl_history  
  
# Output of 'npm pack'  
*.tgz  
  
# Yarn Integrity file  
.yarn-integrity  
  
# dotenv environment variables file  
.env  
  
_book  
.DS_Store
```

- 1. 模块化管理工具

1. 模块化管理工具

常用模块化管理工具的介绍

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-03 14:11:35

- 1. Webpack

1. Webpack

介绍：Webpack 是当下最热门的前端资源模块化管理和打包工具。它可以将许多松散的模块按照依赖和规则打包成符合生产环境部署的前端资源。还可以将按需加载的模块进行代码分隔，等到实际需要的时候再异步加载。通过 Loader 的转换，任何形式的资源都可以视作模块，比如 CommonJs 模块、AMD 模块、ES6 模块、CSS、图片、JSON、Coffeescript、LESS 等。

- 详细讲解请查看：<http://webpackdoc.com/>

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间：2019-07-03 14:11:35

- 1. seajs
 - 1.1.1. config.js 文件配置
 - 1.1.2. 文件引用

1. seajs

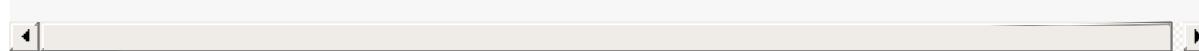
介绍: SeaJS 是一个遵循CMD规范的JavaScript模块加载框架, 可以实现JavaScript的模块化开发及加载机制。

详细API请查看:

- <http://www.zhangxinxu.com/sp/seajs/docs/zh-cn/module-definition.html>
- <http://www.zhangxinxu.com/sp/seajs/docs/zh-cn/cheatsheet.html>

1.1.1. config.js 文件配置

```
seajs.config({
  base: lemon.cdn_asset_url + "/static/" + lemon.builds_path,
  map: [
    ['.js', '.js?v=' + ("undefined"!=typeof lemon&&lemon.version?lemon.version:(new Date).getTime())]
  ],
  alias: {
    "jquery": "js/libs/min/jquery.1.11.1.min",
    "cookie": "js/libs/jquery.cookie",
    "easing": "js/libs/jquery.easing",
    "utils": "js/components/common/utils",
    "component": "js/components/common/component",
    "validate": "js/libs/jquery.validate",
    "validate.add": "js/libs/jquery.validate.add",
    "pager": "js/libs/jquery.pager",
    "layer": "js/libs/layer/layer.min",
    "unslider": "js/libs/min/unslider.min",
    "layer.ext": "js/libs/layer/extend/layer.ext",
    "highStock": "js/libs/min/highstock",
    "mousewheel": "js/libs/min/jquery.mousewheel.min",
    "countUp": "js/libs/countUp"
  }
});
```



注:

- base 是js文件存放的目录
- map 是js文件的后缀和随机版本号
- alias 命名: 地址

1.1.2. 文件引用

```
define(function (require, exports, module) {
    var $ = require('jquery'),
        $layer = require('layer'),
        LemonUtils = require('utils'),
        form = $('#loginForm'),
        popCodeFrom = $('#popCodeFrom'),
        layerNo = 0;

    //引用文件
    require('validate')($);
    require('validate.add')($);
    require('component')($);

    $("#mobile").telephone();

    require("./checkcode.js");

    //其他jq用法都一样
    $("#btnLogin").on("click", function () {
        }
    });
    注: define(function (require, exports, module){}) 跟 $(function(){}) 功能一样
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-03 14:11:35

- 1. js模块

1. js模块

平时用到的一些常见js模块记录笔记

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-03 14:11:35

- 1. 时间相关模块

- 1.1.1. 获得服务器时间
- 1.1.2. 时间格式转化，两者时间相比较
- 1.1.3. 把时间转化为准备时间
- 1.1.4. 计算已经开团时间

1. 时间相关模块

1.1.1. 获得服务器时间

```

function get_server_time(){
    var xmlhttp = false;
    //获取服务器时间
    try {
        xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e2) {
            xmlhttp = false;
        }
    }

    if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
        xmlhttp = new XMLHttpRequest();
    }

    xmlhttp.open("GET", window.location.href.toString(), false);
    xmlhttp.setRequestHeader("If-None-Match", "bytes=-1");
    xmlhttp.setRequestHeader("Cache-Control", "no-cache");
    xmlhttp.send(null);

    return new Date(xmlhttp.getResponseHeader("Date"));
}

```

1.1.2. 时间格式转化，两者时间相比较

```

// 对Date的扩展，将 Date 转化为指定格式的String
// 月(M)、日(d)、小时(h)、分(m)、秒(s)、季度(q) 可以用 1-2 个占位符,
// 年(y)可以用 1-4 个占位符, 毫秒(S)只能用 1 个占位符(是 1-3 位的数字)
// 例子:
// (new Date()).Format("yyyy-MM-dd hh:mm:ss.S") ==> 2006-07-02 08:09:04.423
// (new Date()).Format("yyyy-M-d h:m:s.S")      ==> 2006-7-2 8:9:4.18

Date.prototype.Format = function(fmt){ //author: meizz
    var o = {
        "M+" : this.getMonth()+1,                 //月份
        "d+" : this.getDate(),                   //日

```

```

    "h+" : this.getHours(), //小时
    "m+" : this.getMinutes(), //分
    "s+" : this.getSeconds(), //秒
    "q+" : Math.floor((this.getMonth()+3)/3), //季度
    "S" : this.getMilliseconds() //毫秒
};

if(/(y+)/.test(fmt))
    fmt=fmt.replace(RegExp.$1, (this.getFullYear()+"").substr(4 - RegExp.$1.length));
for(var k in o)
    if(new RegExp("(^"+k+"")").test(fmt))
        fmt = fmt.replace(RegExp.$1, (RegExp.$1.length==1) ? (o[k]) : ((("00"+ o[k]).substr((""+ o[k]).length)));
return fmt;
}

$(function(){
    //两个时间之间的比较
    function dateCompare(dateStart,dateEnd){
        dateStart = dateStart.replace(/\-/gi,"/");
        dateEnd = dateEnd.replace(/\-/gi,"/");
        var time1 = new Date(dateStart).getTime();
        var time2 = new Date(dateEnd).getTime();
        if(time2 > time1){
            return true;
        }
        return false;
    } //eg
})

```

运用场景如：

```

if(dateCompare(new Date($.cookie("dataTime")).Format("yyyy-MM-dd"),new Date().Format("yyyy-MM-dd"))){
    //返回true执行这里面的内容
}
})

```

1.1.3. 把时间转化为准备时间

```

//把时间转换为标准时间
function get_unix_time(dateStr){
    var newstr = dateStr.replace(/\-/g,'/');
    var date = new Date(newstr);
    //var time_str = date.getTime().toString();
    //return time_str.substr(0, 10);
    return date;
}

```

1.1.4. 计算已经开团时间

```
var indate = $("#start_date").val();
var todate = get_server_time(); //服务器时间
var olddate = get_unix_time(indate); //调用上面的函数把时间转为标准时间
var timepass = 0;
function showTime(){
    var timelong = (todate - olddate) + timepass;
    var h = parseInt(timelong/3600000,10);
    var m = parseInt(timelong%3600000/60000,10);
    var s = parseInt(timelong%60000/1000);
    document.getElementById("datetime").innerHTML = (h < 10 ? "0" + h : h) + ":" + (
m < 10 ? "0" + m : m) + ":" + (s < 10 ? "0" + s : s);
    setTimeout(function(){
        timepass += 1000;
        showTime();
    },1000);
}
showTime();

// 运用实例：抱团活动
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-03 14:11:35

- 1. common.js
 - 1.1.1. 相关的正则表达式
 - 1.1.2. 获取url中参数值

1. common.js

1.1.1. 相关的正则表达式

```
function LEMONUtils() {}

LEMONUtils.testFn = {
  isPassword : function(s) {
    return /(?![0-9]+$)(?![A-z]+$)(?!^[_A-z0-9]+$)^.{6,20}$.test(s);
  },
  //邮箱验证
  isEmail : function(s) {
    var isEmailText = /^(?=\w+([-+.'])\w+@\w+([-.\n])\w+\.\w+([-.\n])*$).{0,30}$/;
    return isEmailText.test(s);
  },
  //校验手机号码
  isMobile : function(s) {
    var re=/^1[3-8]\d{9}$/;
    return re.test(s);
  },
  //校验邮政编码
  isZipCode : function(s) {
    var re = /^[1-9][0-9]{5}$/;
    return re.test(s);
  },
  //数字
  isNumber:function(s) {
    var re = /^[0-9]*$/;
    return re.test(s);
  },
  isQQ:function(s) {
    var re = /^[1-9]*$/;
    return re.test(s);
  },
  isNickname:function(s){
    var re =/^\u4E00-\u9FA5]{2,5}$/;
    return re.test(s);
  },
  //是否为金额
  isMoney: function(s) {
    var re = /\d*(\.\d{1,2})?|$|^d*\.\d{1,2})?$/;
    return re.test(s);
  },
  // 全中文
  isChinese: function(value) {
```

```

        return /^[\\u4e00-\\u9fa5]+$/ .test(value);
    },
    // 全数字
    isNum: function(value) {
        return /^[0-9]+$/ .test(value);
    },
    // 全英文
    isEnglish: function(value) {
        return /^[a-zA-Z]+$/ .test(value);
    },
    // 英文、数字
    isPwd: function(value) {
        return /^[a-zA-Z0-9]+$/ .test(value);
    },
    // 英文、数字、汉字
    isUname: function(value) {
        return /^[a-zA-Z0-9\\u4E00-\\u9FA5]+$/ .test(value);
    },
    // 是否为RMB
    isMoney: function(data, isPositive) {
        return isPositive ? /^[\\d+(\\.\\d{1,2})?$/ .test(data) && parseFloat(data) > 0 :
        /^(-)?\\d+(\\.\\d{1,2})?$/ .test(data);
    },
    // 身份证
    isIdCard: function(idCard) {
        //15位和18位身份证号码的正则表达式
        var regIdCard = /^[^1-9]\\d{7}((\\d{2}|1[0-2]))(([0|1|2]\\d)|3[0-1])\\d{3}$)|(^[1-9]\\d{5}[1-9]\\d{3}((\\d{2}|1[0-2]))(([0|1|2]\\d)|3[0-1])(\\d{4})|\\d{3}[Xx])$/;

        //如果通过该验证，说明身份证格式正确，但准确性还需计算
        if (regIdCard .test(idCard)) {
            if (idCard.length == 18) {
                var idCardWi = new Array(7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8
                , 4, 2); //将前17位加权因子保存在数组里
                var idCardY = new Array(1, 0, 10, 9, 8, 7, 6, 5, 4, 3, 2); //这是除以1
                1后，可能产生的11位余数、验证码，也保存成数组
                var idCardWiSum = 0; //用来保存前17位各自乘以加权因子后的总和
                for (var i = 0; i < 17; i++) {
                    idCardWiSum += idCard .substring(i, i + 1) * idCardWi[i];
                }

                var idCardMod = idCardWiSum % 11; //计算出校验码所在数组的位置
                var idCardLast = idCard .substring(17); //得到最后一位身份证号码

                //如果等于2，则说明校验码是10，身份证号码最后一位应该是X
                if (idCardMod == 2) {
                    if (idCardLast == 'X' || idCardLast == 'x') {
                        return true;
                    } else {

                        return false;
                    }
                } else {

```

```

        //用计算出的验证码与最后一位身份证号码匹配，如果一致，说明通过，否则是无效的身
份证号码
        if (idCardLast == idCardY[idCardMod]) {
            return true;
        } else {
            return false;
        }
    }
}
} else {
    return false;
}
return true;
},
//验证银行卡号
isBankCard: function(bankCard) {

    var strBin = "10,18,30,35,37,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,
56,58,60,62,65,68,69,84,87,88,94,95,98,99";

    var lastNum = bankCard.substr(bankCard.length - 1, 1); //取出最后一位 (与luhm进
行比较)
    var first15Num = bankCard.substr(0, bankCard.length - 1); //前15或18位
    var newArr = new Array();
    for (var i = first15Num.length - 1; i > -1; i--) { //前15或18位倒序存进数组
        newArr.push(first15Num.substr(i, 1));
    }
    var arrJiShu = new Array(); //奇数位*2的积 <9
    var arrJiShu2 = new Array(); //奇数位*2的积 >9
    var arrOuShu = new Array(); //偶数位数组
    for (var j = 0; j < newArr.length; j++) {
        if ((j + 1) % 2 == 1) { //奇数位
            if (parseInt(newArr[j]) * 2 < 9)
                arrJiShu.push(parseInt(newArr[j]) * 2);
            else
                arrJiShu2.push(parseInt(newArr[j]) * 2);
        } else //偶数位
            arrOuShu.push(newArr[j]);
    }
    var jishu_child1 = new Array(); //奇数位*2 >9 的分割之后的数组个位数
    var jishu_child2 = new Array(); //奇数位*2 >9 的分割之后的数组十位数
    for (var h = 0; h < arrJiShu2.length; h++) {
        jishu_child1.push(parseInt(arrJiShu2[h]) % 10);
        jishu_child2.push(parseInt(arrJiShu2[h]) / 10);
    }
    var sumJiShu = 0; //奇数位*2 < 9 的数组之和
    var sumOuShu = 0; //偶数位数组之和
    var sumJiShuChild1 = 0; //奇数位*2 >9 的分割之后的数组个位数之和
    var sumJiShuChild2 = 0; //奇数位*2 >9 的分割之后的数组十位数之和
    var sumTotal = 0;
    for (var m = 0; m < arrJiShu.length; m++) {
        sumJiShu = sumJiShu + parseInt(arrJiShu[m]);
    }
}

```

```

        for (var n = 0; n < arrOuShu.length; n++) {
            sumOuShu = sumOuShu + parseInt(arrOuShu[n]);
        }
        for (var p = 0; p < jishu_child1.length; p++) {
            sumJiShuChild1 = sumJiShuChild1 + parseInt(jishu_child1[p]);
            sumJiShuChild2 = sumJiShuChild2 + parseInt(jishu_child2[p]);
        }
        //计算总和
        sumTotal = parseInt(sumJiShu) + parseInt(sumOuShu) + parseInt(sumJiShuChild1)
+ parseInt(sumJiShuChild2);
        //计算Luhm值
        var k = parseInt(sumTotal) % 10 == 0 ? 10 : parseInt(sumTotal) % 10;
        var luhm = 10 - k;

        if (strBin.indexOf(bankCard.substring(0, 2)) == -1) {
            return false;
        }
        if (!(lastNum == luhm)) {
            return false;
        }
        return true;
    },
    // 邮件
    isEmail: function(value) {
        return /^[a-zA-Z0-9.!#$%&'*+\-/=?^_`{|}~-]+@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9]))?$/i.test(value);
    }
}

// 注: 调用 LEMONutils.testFn.isIdCard(value)

```

1.1.2. 获取url中参数值

```

function LEMONutils() {}
LEMONutils.getQueryString = function(name) {
    var reg = new RegExp("(^|&) " + name + "=([^\&]*)(\&|$)", "i");
    var r = decodeURIComponent(location.search).substr(1).match(reg);
    if (r != null) return unescape(r[2]);
    return "";
}

// 注: 运用 var _type = Lemonutils.getQueryString('type');

```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-03 14:11:35

- 1. js及jquery获取页面和元素的宽高

- 1.1.1. js:

- 1.1.2. jquery:

1. js及jquery获取页面和元素的宽高

1.1.1. js:

```

网页可见区域宽: document.body.clientWidth; (不含滚动条)
网页可见区域高: document.body.clientHeight; (不含滚动条)
网页可见区域宽: document.body.offsetWidth; (包括边线的宽);
网页可见区域高: document.body.offsetHeight; (包括边线的宽);
网页正文全文宽: document.body.scrollWidth;
网页正文全文高: document.body.scrollHeight;
网页被卷去的高(ff): document.body.scrollTop;
网页被卷去的高(ie): document.documentElement.scrollTop;
网页被卷去的左: document.body.scrollLeft;
网页正文部分上: window.screenTop;
网页正文部分左: window.screenLeft;
某个元素的宽度: obj.offsetWidth;
某个元素的高度: obj.offsetHeight;
某个元素的上边界到body最顶部的距离: obj.offsetTop; (在元素的包含元素不含滚动条的情况下)
某个元素的左边界到body最左边的距离: obj.offsetLeft; (在元素的包含元素不含滚动条的情况下)
返回当前元素的上边界到它的包含元素的上边界的偏移量: obj.offsetTop; (在元素的包含元素含滚动条的情况下)
返回当前元素的左边界到它的包含元素的左边界的偏移量: obj.offsetLeft; (在元素的包含元素含滚动条的情况下)

屏幕分辨率的高: window.screen.height
屏幕分辨率的宽: window.screen.width
屏幕可用工作区高度: window.screen.availHeight
屏幕可用工作区宽度: window.screen.availWidth

HTML精确定位: scrollLeft, scrollWidth, clientWidth, offsetWidth
scrollHeight: 获取对象的滚动高度。
scrollLeft 设置或获取位于对象左边界和窗口中目前可见内容的最左端之间的距离
scrollTop 设置或获取位于对象最顶端和窗口中可见内容的最顶端之间的距离
scrollWidth 获取对象的滚动宽度
offsetHeight 获取对象相对于版面或由父坐标 offsetParent 属性指定的父坐标的高度
offsetLeft 获取对象相对于版面或由 offsetParent 属性指定的父坐标的计算左侧位置
offsetTop 获取对象相对于版面或由 offsetTop 属性指定的父坐标的计算顶端位置
event.clientX 相对文档的水平座标
event.clientY 相对文档的垂直座标
event.offsetX 相对容器的水平坐标
event.offsetY 相对容器的垂直坐标
document.documentElement.scrollTop 垂直方向滚动的值
event.clientX+document.documentElement.scrollTop 相对文档的水平座标+垂直方向滚动的量

```

1.1.2. jquery:

```
获取浏览器显示区域（可视区域）的高度 : $(window).height();
获取浏览器显示区域（可视区域）的宽度 : $(window).width();
获取页面的文档高度: $(document).height();
获取页面的文档宽度 : $(document).width();
浏览器当前窗口文档body的高度: $(document.body).height();
浏览器当前窗口文档body的宽度: $(document.body).width();
获取滚动条到顶部的垂直高度（即网页被卷上去的高度）: $(document).scrollTop();
获取滚动条到左边的垂直宽度 : $(document).scrollLeft();
获取或设置元素的宽度: $(obj).width();
获取或设置元素的高度: $(obj).height();

获取或设置元素的宽度: $(obj).innerWidth(); (height + padding)
获取或设置元素的高度: $(obj).innerHeight(); (height + padding)

获取或设置元素的宽度: $(obj).outerWidth(); (height + padding + border)
获取或设置元素的高度: $(obj).outerHeight(); (height + padding + border)

获取或设置元素的宽度: $(obj).outerWidth(true); (height + padding + border + margin)
获取或设置元素的高度: $(obj).outerHeight(true); (height + padding + border + margin)
某个元素的上边界到body最顶部的距离: obj.offsetTop().top; (在元素的包含元素不含滚动条的情况下)
某个元素的左边界到body最左边的距离: obj.offset().left; (在元素的包含元素不含滚动条的情况下)
返回当前元素的上边界到它的包含元素的上边界的偏移量: obj.offsetTop().top (在元素的包含元素含滚动条的情况下)
返回当前元素的左边界到它的包含元素的左边界的偏移量: obj.offset().left (在元素的包含元素含滚动条的情况下)
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-03 14:11:35

- 1. js常用函数
 - 1.1.1. JS Math 对象

1. js常用函数

1、split() 方法用于把一个字符串分割成字符串数组。如：

```
var str="how are you doing today?"
console.log(str.split(" "));      //["how", "are", "you", "doing", "today?"]
console.log(str.split(" ",3));    //["how", "are", "you"]
```

2、join() 方法用于把数组中的所有元素放入一个字符串。如：

```
var arr = ["George", "John", "Thomas"];
console.log(arr.join());          //"George,John,Thomas"
console.log(arr.join("."));       //"George.John.Thomas"
```

3、concat() 方法用于连接两个或多个数组。如：

```
var a = [1, 2, 3], b=[4, 5, 6]
console.log(a.concat(4,5));   // [1, 2, 3, 4, 5]
console.log(a.concat(b));     // [1, 2, 3, 4, 5, 6]
```

4、pop() 方法用于删除并返回数组的最后一个元素。如：

```
var arr = ["George", "John", "Thomas"];
console.log(arr.pop());        //"Thomas"
console.log(arr);              // ["George", "John"]
```

5、push() 方法可向数组的末尾添加一个或多个元素，并返回新的长度。如：

```
var arr = ["George", "John", "Thomas"];
console.log(arr.push("tom"));   //4
console.log(arr);              // ["George", "John", "Thomas", "tom"]
```

6、reverse() 方法用于颠倒数组中元素的顺序。如：

```
var arr = ["George", "John", "Thomas"];
console.log(arr.reverse());     // ["tom", "Thomas", "John", "George"]
```

7、shift() 方法用于把数组的第一个元素从其中删除，并返回第一个元素的值。(如果数组是空的，那么shift() 方法将不进行任何操作，返回 undefined 值。请注意，该方法不创建新数组，而是直接修改原有的 arrayObject)如：

```
var arr = ["George", "John", "Thomas"];
```

```
console.log(arr.shift());      //"tom"
```

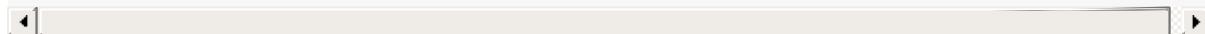
8、slice() 方法可从已有的数组中返回选定的元素。语法：arrayObject.slice(start,end) start 必需。规定从何处开始选取。如果是负数，那么它规定从数组尾部开始算起的位置。也就是说，-1 指最后一个元素，-2 指倒数第二个元素，以此类推。

end 可选。规定从何处结束选取。该参数是数组片断结束处的数组下标。如果没有指定该参数，那么切分的数组包含从 start 到数组结束的所有元素。如果这个参数是负数，那么它规定的是从数组尾部开始算起的元素。

```
var arr = ["George", "John", "Thomas"];
console.log(arr.slice(1));    //["John", "George"]
console.log(arr.slice(0,2)); //["Thomas", "John"]
```

9、sort() 方法用于对数组的元素进行排序。 (对数组元素首字母或首数字排序) 如：

```
var arr=["George", "John", "Thomas", "James", "Adrew", "Martin"];
console.log(arr.sort()); //["Adrew", "George", "James", "John", "Martin", "Thomas"]
```



对数字进行排序需借助函数

```
var arr=[10, 5, 40, 25, 1000, 1];
function sortNumber(a, b){
  return a - b
}
console.log(arr.sort(sortNumber)); // [1, 5, 10, 25, 40, 1000]
```

10、splice() 方法向/从数组中添加/删除项目，然后返回被删除的项目。如：

```
var arr=["George", "John", "Thomas", "James", "Adrew", "Martin"];
//删除从 index 2 ("Thomas") 开始的三个元素，并添加一个新元素 ("William") 来替代被删除的元素
console.log(arr.splice(2,3,"William")); // ["Thomas", "James", "Adrew"]
console.log(arr); // ["George", "John", "William", "Martin"]
```

11、toString() 方法可把数组转换为字符串，并返回结果。如：

```
var arr=["George", "John", "Thomas", "James", "Adrew", "Martin"];
console.log(arr.toString()); // "George, John, Thomas, James, Adrew, Martin"
```

12、toLocaleString() 把数组转换为本地字符串。如：

```
var arr=["George", "John", "Thomas", "James", "Adrew", "Martin"];
console.log(arr.toLocaleString()); // "George, John, Thomas, James, Adrew, Martin"
```

13、`unshift()` 方法可向数组的开头添加一个或更多元素，并返回新的长度。 (注：`unshift()` 方法无法在 Internet Explorer 中正确地工作！) 如：

```
var arr=["George", "John", "Thomas"];
console.log(arr.unshift("William")); //4
console.log(arr); //["William", "George", "John", "Thomas"]
```

14、`valueOf()` 方法返回 Array 对象的原始值。如：

```
var arr=["George", "John", "Thomas"];
console.log(arr.valueOf()); //["George", "John", "Thomas"]
```

15、js数据类型 字符串、数字、布尔、数组、对象、Null、Undefined

1.1.1. JS Math 对象

具体的可查看：http://www.w3school.com.cn/jsref/jsref_obj_math.asp，下面列举几个主要的

1、`Math.abs(x)` 方法可返回数的绝对值。如：

```
document.write(Math.abs(7.25)) //7.25
document.write(Math.abs(-7.25)) //7.25
document.write(Math.abs(7.25-10)) //2.75
```

2、`Math.ceil(x)` 方法可对一个数进行上舍入。如：

```
document.write(Math.ceil(0.60)) // 1
document.write(Math.ceil(5.1)) // 6
document.write(Math.ceil(-5.1)) // -5
document.write(Math.ceil(-5.9)) // -5
```

3、`Math.floor(x)` 方法可对一个数进行下舍入。如：

```
document.write(Math.floor(0.60)) //0
document.write(Math.floor(0.40)) //0
document.write(Math.floor(5)) //5
document.write(Math.floor(-5.9)) //-6
```

4、`Math.max(x,y)` 方法可返回两个指定的数中带有较大的值的那个数。如：

```
document.write(Math.max(5,7)) //7
document.write(Math.max(-3,5)) //5
document.write(Math.max(-3,-5)) //-3
```

5、`Math.min(x,y)` 方法可返回指定的数字中带有最低值的数字。如：

```
document.write(Math.max(5,7))          //5
document.write(Math.max(-3,5))         //-3
document.write(Math.max(-3,-5))        //-5
```

6、Math.random() 方法可返回介于 0 ~ 1 之间的一个随机数。如： document.write(Math.random())

//0.17967150011099875 7、Math.round(x) 方法可把一个数字舍入为最接近的整数。如：

```
document.write(Math.round(0.60))      //1
document.write(Math.round(0.50))      //1
document.write(Math.round(0.49))      //0
document.write(Math.round(-4.40))     //-4
document.write(Math.round(-4.60))     //-5
```

8、object.toSource() 方法返回表示对象源代码的字符串。如：

```
function employee(name,job,born){
  this.name=name;
  this.job=job;
  this.born=born;
}
var bill=new employee("Bill Gates","Engineer",1985);
document.write(bill.toSource());      //({name:"Bill Gates", job:"Engineer", born:1985})
```

9、valueOf() 方法可返回 Boolean 对象的原始值。如：

对象	返回值
Array	数组实例对象。
Boolean	布尔值。
Date	以毫秒数存储的时间值，从 UTC 1970 年 1 月 1 日午夜开始计算。
Function	函数本身。
Number	数字值。
Object	对象本身。这是默认设置。
String	字符串值。

```
// Array: 返回数组对象本身
var array = ["CodePlayer", true, 12, -5];
document.writeln( array.valueOf() === array ); // true

// Date: 当前时间距1970年1月1日午夜的毫秒数
var date = new Date(2013, 7, 18, 23, 11, 59, 230);
document.writeln( date.valueOf() ); // 1376838719230

// Number: 返回数字值
var num = 15.26540;
document.writeln( num.valueOf() ); // 15.2654

// 布尔: 返回布尔值true或false
var bool = true;
document.writeln( bool.valueOf() === bool ); // true
```

```
// new一个Boolean对象
var newBool = new Boolean(true);
// valueOf()返回的是true, 两者的值相等
document.writeln( newBool.valueOf() === newBool ); // true
// 但是不全等, 两者类型不相等, 前者是boolean类型, 后者是object类型
document.writeln( newBool.valueOf() === newBool ); // false

// Function: 返回函数本身
function foo(){
}
document.writeln( foo.valueOf() === foo ); // true
var foo2 = new Function("x", "y", "return x + y;");
document.writeln( foo2.valueOf() === foo2 ); // true

// Object: 返回对象本身
var obj = {name: "张三", age: 18};
document.writeln( obj.valueOf() === obj ); // true

// String: 返回字符串值
var str = "http://www.365mini.com";
document.writeln( str.valueOf() === str ); // true
// new一个字符串对象
var str2 = new String("http://www.365mini.com");
// 两者的值相等, 但不全等, 因为类型不同, 前者为string类型, 后者为object类型
document.writeln( str2.valueOf() === str2 ); // false
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-03 14:11:35

- 1. ajax写法

1. ajax写法

```
$(function(){
    $.ajax({
        url: '/path/to/file',
        type: 'get or post',
        dataType: 'default: Intelligent Guess (Other values: xml, json, jsonp, script, or html)',
        data: {param1:'value1', param2:'value2'},
        jsonp: "jsoncallback", //ajax跨域写法
        async: true, //默认值: true。默认设置下，所有请求均为异步请求。如果需要发送同步请求，请将此选项设置为false。
        processData: false, // 告诉jQuery不要去处理发送的数据
        contentType: false, // 告诉jQuery不要去设置Content-Type请求头
        success: function (data){
            /*当请求之后调用。传入返回后的数据，以及包含成功代码的字符串。*/
        },
        dataFilter: function(){
            /*在请求成功之后调用。传入返回的数据以及 "dataType" 参数的值。并且必须返回新的数据(可能是处理过的) 传递给 success 回调函数。*/
        },
        error: function(){
            /*在请求出错时调用*/
        },
        beforeSend: function(){
            /*在发送请求之前调用*/
        },
        complete: function(){
            /*当请求完成之后调用这个函数*/
        }
    });
})
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-03 14:11:35

- 1. 常用插件

1. 常用插件

项目开发中用到的一些插件汇总

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-03 14:11:36

- 1. 常用弹层插件

- 1.1.1. 1、layer 弹出层
- 1.1.2. 2、fancyBox 弹出层
- 1.1.3. 3、jquery.simplemodal.min.js 插件弹出层

1. 常用弹层插件

1.1.1. 1、layer 弹出层

api文档地址: <https://wwwlayui.com/doc/modules/layer.html>

实例: (注: 手机, PC端都比较好用)

```
var layerNo = layer.open({
  type: 1,
  shade: false,
  title: false, //不显示标题
  content: $('.popup-box'), //捕获的元素
  area: ['200px', '100px']
});

layer.close(layerNo); //关闭当前弹层
layer.closeAll(); //关闭当前所有的弹层
```

1.1.2. 2、fancyBox 弹出层

api文档: <http://fancyboxapps.com/fancybox/#docs> <http://simplemodal.plasm.it/> 中文文档: <http://www.weste.net/2013/4-22/90629.html>

实例: (注: 根本网站需求具体运用还是查看API, 主要运用在PC端)

```
var content="sdfs"
function showDiv(){
  var content=<div class='content_box'>+
    "<div class='content'>fancybox弹出内容</div>"+
    "<a class='btn' href='javascript:$._fancybox.close();确定</a>"+
    "</div>";
  return packageFancybox(content);
}
function packageFancybox(template){
  return $.fancybox(template, {
    padding:20,
    margin:10,
    width:200,
    height:40,
    minWidth:200,
    autoSize:false, //默认是:true,自适应, 变为:false后设置的宽和高才有效。
    closeBtn:false, //默认是:true, 变为:false后close键隐藏。
    closeClick:true, //默认是:false, 变为:true后点击弹框内的内容弹框也会消失。
  })
}
```

```
        modal:true           //默认是:false,变为:true后点击外面透明黑背景弹框不会消失。
    })
}
showDiv();
```

1.1.3. 3、jquery.simplemodal.min.js 插件弹出层

api文档地址: http://blog.sina.com.cn/s/blog_6a1fb8420100psj0.html

实例:

```
var _deal = $("#register-notice").modal({width: 800, height: 600, close:false});

_deal.close();      //关闭弹出层
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-03 14:11:36

- 1. 常用效果插件

1. 常用效果插件

- 1、**select**选择下拉效果：<https://select2.github.io/examples.html>
- 2、**jQuery全屏滚动插件fullPage.js**：<http://www.jq22.com/jquery-info1124>
- 3、**Swiper演示**：<http://www.swiper.com.cn/demo/index.html>
- 4、**superslide2插件**：<http://www.superslide2.com/demo.html>
- 5、**ECharts图表数据插件**：<http://echarts.baidu.com/index.html>
- 6、**Highcharts图表数据插件**：<http://www.hcharts.cn/index.php>

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-03 14:11:36

- 1. Hexo博客yilia主题添加背景音乐- (网易云音乐)

1. Hexo博客yilia主题添加背景音乐- (网易云音乐)

- 博主博客地址

1、打开网易云音乐首页，复制选择的背景音乐分享外链

网易云音乐

2、引入播放器代码

在 `themes/yilia/layout/_partial/left-col.ejs` 文件 `nav` 标签中添加代码

注：放在 `<nav>` 和 `</nav>` 之内

笔者添加了一些判断和表达式

```
<!-- 网易云音乐插件 -->
<% if (theme.music && theme.music.enable){ %>
    <div style="position: absolute; bottom: 120px left: auto; width: 85%">
        <iframe frameborder="no" border="0" marginwidth="0" marginheight="0" width="240" height="52" src="//music.163.com/outchain/player?type=2&id=<%=theme.music.id|1334445174%>&auto=<%=theme.music.autoplay?1:0%>&height=32"></iframe>
    </div>
<% } %>
```



```
<nav class="header-nav">
    <div class="social">
        <% for (var i in theme.subnav){ %>
            <a class="<%= i %>" target="_blank" href="<%- url_for(theme.subnav[i]) %>" title="<%= i %>"><i class="icon-<%= i %>">/<% } %>
    </div>
    <!-- 网易云音乐插件 -->
    <% if (theme.music && theme.music.enable){ %>
        <div style="position: absolute; bottom: 120px left: auto; width: 85%">
            <iframe frameborder="no" border="0" marginwidth="0" marginheight="0" width="240" height="52" src="//music.163.com/outchain/player?type=2&id=<%=theme.music.id|1334445174%>&auto=<%=theme.music.autoplay?1:0%>&height=32"></iframe>
        </div>
    <% } %>
</nav>
```

3、在yilia主题配置文件中添加属性

```
# 网易云音乐插件
music:
    enable: true
    #id: 1332647902 # 网易云分享的ID
    autoplay: true # 是否开启自动播放
```


- 1. Hexo博客添加helper-live2d动态模型插件

1. Hexo博客添加helper-live2d动态模型插件

Tips:前导必备

- 博主博客地址
 - 博主使用的是 yilia 主题
 - 插件的github地址hexo-helper-live2d
 - live2d模型仓库
 - 必须有Node环境 Linux下安装node和npm
-

1、安装模块

在hexo根目录执行命令

```
# npm install --save hexo-helper-live2d
```

2、下载模型

作者各种模型包展示 hexo live2d插件 2.0 !

live2d模型仓库

- live2d-widget-model-chitose
- live2d-widget-model-epsilon2_1
- live2d-widget-model-gf
- live2d-widget-model-haru/01 (use `npm install --save live2d-widget-model-haru`)
- live2d-widget-model-haru/02 (use `npm install --save live2d-widget-model-haru`)
- live2d-widget-model-haruto
- live2d-widget-model-hibiki
- live2d-widget-model-hijiki
- live2d-widget-model-izumi
- live2d-widget-model-koharu
- live2d-widget-model-miku
- live2d-widget-model-ni-j
- live2d-widget-model-nico
- live2d-widget-model-nietzsche
- live2d-widget-model-nipsilon
- live2d-widget-model-nito
- live2d-widget-model-shizuku
- live2d-widget-model-tororo
- live2d-widget-model-tsumiki

- live2d-widget-model-unitychan
- live2d-widget-model-wanko
- live2d-widget-model-z16

使用 `npm install {packagename}` 安装模型

-- 笔者使用的模型

```
# npm install live2d-widget-model-hibiki
```

3、详细配置

在Hexo的 `_config.yml` 文件或主题的 `_config.yml` 文件中添加配置. 参阅 [live2d-widget.js API](#)

- 博主配置文件示例

```
# Live2D
## https://github.com/EYHN/hexo-helper-live2d
## https://l2dwidget.js.org/docs/class/src/index.js~L2Dwidget.html#instance-method-init
live2d:
  enable: true
  #enable: false
  scriptFrom: local # 默认
  pluginRootPath: live2dw/ # 插件在站点上的根目录(相对路径)
  pluginJsPath: lib/ # 脚本文件相对与插件根目录路径
  pluginModelPath: assets/ # 模型文件相对与插件根目录路径
  # scriptFrom: jsdelivr # jsdelivr CDN
  # scriptFrom: unpkg # unpkg CDN
  # scriptFrom: https://cdn.jsdelivr.net/npm/live2d-widget@3.x/lib/L2Dwidget.min.js
  # 你的自定义 url
  tagMode: false # 标签模式, 是否仅替换 live2d tag标签而非插入到所有页面中
  debug: false # 调试, 是否在控制台输出日志
  model:
    use: live2d-widget-model-hibiki
    # use: live2d-widget-model-wanko # npm-module package name
    # use: wanko # 博客根目录/live2d_models/ 下的目录名
    # use: ./wives/wanko # 相对于博客根目录的路径
    # use: https://cdn.jsdelivr.net/npm/live2d-widget-model-wanko@1.0.5/assets/wanko.model.json # 你的自定义 url
    display:
      position: right
      width: 145
      height: 315
    mobile:
      show: true # 是否在移动设备上显示
      scale: 0.5 # 移动设备上的缩放
    react:
      opacityDefault: 0.7
      opacityOnHover: 0.8
```

4、作者寄语

Everything is ok. But you have to do by yourself.

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:51:06

- 1. Hexo博客添加访问量统计

1. Hexo博客添加访问量统计

Tips:前导必备

- 博主博客地址
- 博主使用的是 yilia 主题
- 访问量统计使用不蒜子

1、配置是否开启不蒜子访问量统计功能

在 themes/yilia/_config.yml 添加属性

```
# 是否开启访问量统计功能(不蒜子)
```

```
busuanzi:  
  enable: true
```

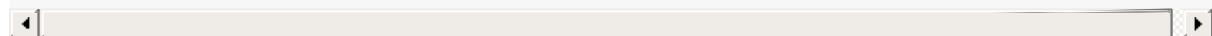
```
#6、是否开启访问量统计功能(不蒜子)  
busuanzi:  
  enable: true
```



2、引入不蒜子并添加站点访问量

在 themes/yilia/layout/_partial/footer.ejs 末尾添加如下代码

```
<% if (theme.busuanzi && theme.busuanzi.enable){ %>  
    <!-- 不蒜子统计 -->  
    <span id="busuanzi_container_pv">  
        本站总访问量<span id="busuanzi_value_site_pv"></span>次  
    </span>  
    <span class="post-meta-divider">|</span>  
    <span id="busuanzi_container_uv" style='display:none'>  
        本站访客数<span id="busuanzi_value_site_uv"></span>人  
    </span>  
    <script async src="//busuanzi.ibruce.info/busuanzi/2.3/busuanzi.pure.mini.js  
    "></script>  
    <% } %>
```



```
<footer id="footer">
  <div class="outer">
    <div id="footer-info">
      <div class="footer-left">
        &copy; <%= date(new Date(), 'YYYY') %> <%= config.author || config.title %>
      </div>
      <div class="footer-right">
        码云:<a href="https://gitee.com/zhouwei" target="_blank">思伟</a> by Litten
      </div>
    </div>
    <% if (theme.busuanzi && theme.busuanzi.enable){ %>
      <!-- 不蒜子统计 -->
      <span id="busuanzi_container_site_pv">
        本站总访问量<span id="busuanzi_value_site_pv"></span>次
      </span>
      <span class="post-meta-divider">|</span>
      <span id="busuanzi_container_site_uv" style='display:none'>
        本站访客数<span id="busuanzi_value_site_uv"></span>人
      </span>
      <script async src="//busuanzi.ibruce.info/busuanzi/2.3/busuanzi.pure.mini.js"></script>
    <% } %>
  </div>
</footer>
```

3、添加文章访问量

在 themes/yilia/layout/_partial/post/date.ejs 开头添加如下代码

```
<% if (theme.busuanzi && theme.busuanzi.enable && !index){ %>
  <!-- 不蒜子统计 -->
  <span id="busuanzi_container_page_pv" style='display:none' class="<%= class_name %>">
    <i class="icon-smile icon"></i> 阅读数: <span id="busuanzi_value_page_pv"></span>次
  </span>
<% } %>

<% if (theme.busuanzi && theme.busuanzi.enable && !index){ %>
  <!-- 不蒜子统计 -->
  <span id="busuanzi_container_page_pv" style='display:none' class="<%= class_name %>">
    <i class="icon-smile icon"></i> 阅读数: <span id="busuanzi_value_page_pv"></span>次
  </span>
<% } %>
<a href="<%- url_for(post.path) %>" class="<%= class_name %>">
  <time datetime="<%= date_xml(post.date) %>" itemprop="datePublished"><i class="icon-calendar icon"></i><%= date(post.date, date_format) %></time>
</a>
~
```

- 重启即可查看效果

4、作者寄语

Everything is ok. But you have to do by yourself.

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:26

- 1. 使用Hexo+GitHub搭建免费个人博客
 - 1.1.1. 1、前言
 - 1.1.2. 2、搭建GitHub博客
 - 1.1.3. 3、使用hexo写博客

1. 使用Hexo+GitHub搭建免费个人博客

- 博主GitHub博客地址:<https://joeybling.github.io/>
- 联系QQ: 2434387555
- 此博客的所有操作均在 Linux 上完成

本文所使用的环境: Centos 6.5Final@64Bit 、 NodeJs@v8.11.1 、 hexo@3.2.2

1.1.1. 1、前言

使用 `github-pages` 服务搭建博客的好处有:

1. 全是静态文件，访问速度快；
2. 免费方便，不用花一分钱就可以搭建一个自由的个人博客，不需要服务器不需要后台；
3. 可以随意绑定自己的域名，不仔细看的话根本看不出来你的网站是基于github的；
4. 数据绝对安全，基于github的版本管理，想恢复到哪个历史版本都行；
5. 博客内容可以轻松打包、转移、发布到其它平台；
6. 等等；

1.1、准备工作

- 有一个github账号，没有的话去注册一个
- 安装了 `node.js` 、 `npm`，并了解相关基础知识
- 传送门 [Linux上安装node和npm](#)

1.1.2. 2、搭建GitHub博客

2.1、创建仓库

建一个名为 你的用户名.github.io 的仓库，比如说，如果你的github用户名是test，那么你就新建 `test.github.io` 的仓库（必须是你的用户名，其它名称无效），将来你的网站访问地址就是 <http://test.github.io> 了，是不是很方便？几个注意的地方：

- 注册的邮箱一定要验证，否则不会成功；
- 仓库名字必须是: `username.github.io`，其中username是你的用户名；
- 仓库创建成功不会立即生效，需要过一段时间，大概3-10分钟左右
- 仓库创建成功后需要开启pages服务 `settings GitHub Pages` 选择分支并开启服务

2.2、绑定域名(看自己选择是否使用)

绑定域名分2种情况：带www和不带www的。

域名配置最常见有2种方式，CNAME和A记录，CNAME填写域名，A记录填写IP，由于不带www方式只能采用A记录，所以必须先ping一下你的用户名.github.io的IP，然后到你的域名DNS设置页，将A记录指向你ping出来的IP，将CNAME指向你的用户名.github.io，这样可以保证无论是否添加www都可以访问，如下：

记录类型	主机记录	解析线路	记录值
CNAME	██████████	默认	██████████
MX	@	默认	██████████
A	@	默认	103.245.222.133
CNAME	www	默认	liuxianan.github.io

然后到你的github项目根目录新建一个名为CNAME的文件（无后缀），里面填写你的域名，加不加www看你自己喜好，因为经测试：

- 如果你填写的是没有www的，比如 mygit.me，那么无论是访问 <http://www.mygit.me> 还是 <http://mygit.me>，都会自动跳转到 <http://mygit.me>
- 如果你填写的是带www的，比如 www.mygit.me，那么无论是访问 <http://www.mygit.me> 还是 <http://mygit.me>，都会自动跳转到 <http://www.mygit.me>
- 如果你填写的是其它子域名，比如 abc.mygit.me，那么访问 <http://abc.mygit.me> 没问题，但是访问 <http://mygit.me>，不会自动跳转到 <http://abc.mygit.me>

另外说一句，在你绑定了新域名之后，原来的你的用户名.github.io并没有失效，而是会自动跳转到你的新域名。

1.1.3. 3、使用hexo写博客

3.1、安装hexo

```
# npm install -g hexo-cli
# 做个软链接
# cd /usr/local/bin
# 这是是你的Node解压安装的路径
# ln -s /opt/node-v8.11.1-linux-x64/bin/hexo
```

3.2、初始化

在某个地方新建一个名为hexo的文件夹

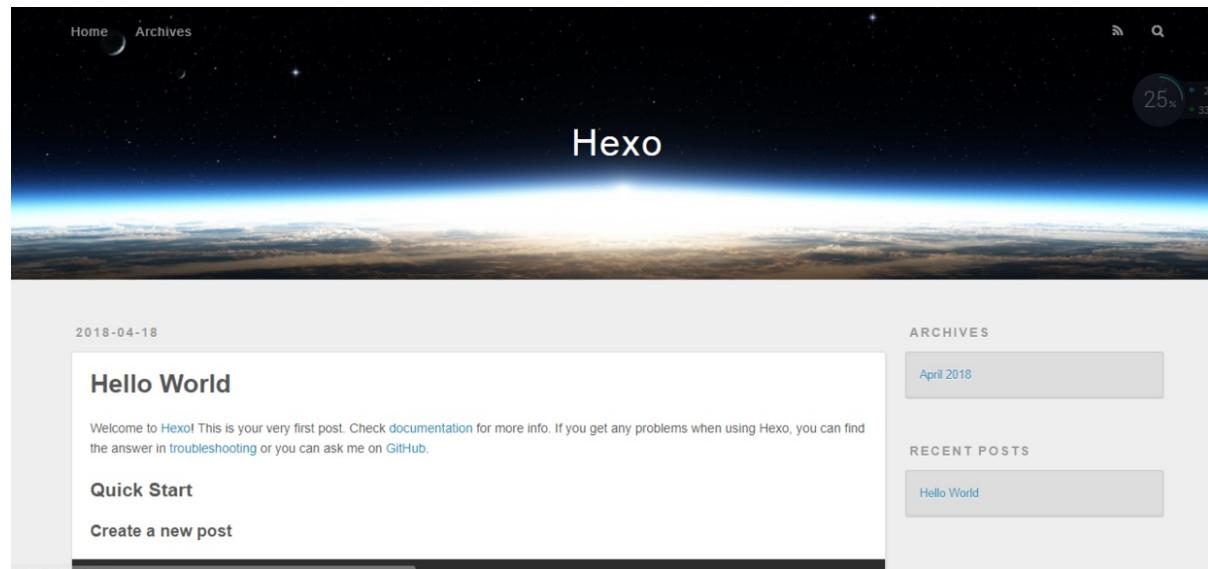
```
# mkdir -p /workspaces/hexo
# cd /workspaces/hexo/
# hexo init
```

目录结构如下图:

```
root@app03 hexo]# ll
total 128
-rw-r--r-- 1 root root 1755 Apr 18 09:46 _config.yml
-rw-r--r-- 1 root root 174 Apr 18 09:49 db.json
-rwxr-xr-x 236 root root 12288 Apr 18 09:46 node_modules
-rw-r--r-- 1 root root 447 Apr 18 09:49 package.json
-rw-r--r-- 1 root root 92430 Apr 18 09:46 package-lock.json
-rwxr-xr-x 2 root root 4096 Apr 18 09:46 scaffolds
-rwxr-xr-x 3 root root 4096 Apr 18 09:46 source
-rwxr-xr-x 3 root root 4096 Apr 18 09:46 themes
root@app03 hexo]#
```

```
# hexo g # 生成
# hexo s # 启动服务
```

打开浏览器访问 <http://IP:4000> 即可看到内容如下:



3.3、修改主题

- 有哪些好看的 Hexo 主题? 传送门 <https://www.zhihu.com/question/24422335>
- 个人比较喜欢的 <https://github.com/litten/hexo-theme-yilia>

```
# cd /workspaces/hexo/themes/
# yum install -y wget
# wget https://codeload.github.com/litten/hexo-theme-yilia/zip/master
# yum install -y unzip
# unzip master
# mv hexo-theme-yilia-master/ yilia
# ll
```

```
root@app03 themes]# ll
total 328
-rwxr-xr-x 6 root root 4096 Apr 18 09:46 landscape
-rw-r--r-- 1 root root 326543 Apr 18 09:55 master
-rwxr-xr-x 6 root root 4096 Nov 15 16:09 yilia
下载后的主题都在这里: root@app03 themes]#
```

修改 `_config.yml` 中的 `theme: landscape` 改为 `theme: yilia`，然后重新执行 `hexo g` 来重新生成。

如果出现一些莫名其妙的问题，可以先执行 `hexo clean` 来清理一下 `public` 的内容，然后再来重新生成和发布。

```
# yum install -y vim
# 修改_config.yml中的theme: landscape改为theme: yilia
# vim /workspaces/hexo/_config.yml
# 重新生成
# hexo g
# hexo s
```

```
# Extensions
## Plugins: https://hexo.io/plugins
## Themes: https://hexo.io/themes/
#theme: landscape
theme: yilia
```

3.4、上传到GitHub

上传 `public` 文件夹里所有的文件，可以拷贝出来使用 `git` 上传到 GitHub

3.5、常用 `hexo` 命令

- `hexo new "postName"` 新建文章
- `hexo new page "pageName"` 新建页面
- `hexo generate` 生成静态页面至 `public` 目录
- `hexo server` 开启预览访问端口（默认端口4000，'ctrl + c'关闭server）
- `hexo deploy` 部署到 GitHub
- `hexo help` 查看帮助
- `hexo version` 查看 Hexo 的版本

缩写：

- `hexo n == hexo new`
- `hexo g == hexo generate`
- `hexo s == hexo server`
- `hexo d == hexo deploy`

组合命令：

- `hexo s -g` 生成并本地预览
- `hexo d -g` 生成并上传

3.6、`_config.yml` 文件配置

需要特别注意的地方是，冒号后面必须有一个空格，否则可能会出问题。

3.7、写博客

定位到我们的hexo根目录，执行命令：

```
# hexo new 'my-first-blog'
```

hexo 会帮我们在`_posts`下生成相关md文件:

```
[root@app03 hexo]#  
[root@app03 hexo]#  
[root@app03 hexo]# hexo new 'my-first-blog'  
INFO  Created: /workspaces/hexo/source/_posts/my-first-blog.md  
[root@app03 hexo]#
```

我们只需要打开这个文件就可以开始写博客了，默认生成如下内容：

当然你也可以直接自己新建md文件，用这个命令的好处是帮我们自动生成了时间

一般完整格式如下：

```
---  
title: postName #文章页面上的显示名称，一般是中文  
date: 2013-12-02 15:30:16 #文章生成时间，一般不改，当然也可以任意修改  
categories: 默认分类 #分类  
tags: [tag1,tag2,tag3] #文章标签，可空，多标签请用格式，注意：后面有个空格  
description: 附加一段文章摘要，字数最好在140字以内，会出现在meta的description里面  
---  
以下是正文.....
```

推荐一个在线写博客工具 <http://pandao.github.io/editor.md/>

Tips:如何让博文列表不显示全部内容

默认情况下，生成的博文目录会显示全部的文章内容，如何设置文章摘要的长度呢？

答案是在合适的位置加上

```
<!-- more -->
```

即可，例如：

```
# 前言
```

使用github pages服务搭建博客的好处有：

1. 全是静态文件，访问速度快；
2. 免费方便，不用花一分钱就可以搭建一个自由的个人博客，不需要服务器不需要后台；
3. 可以随意绑定自己的域名，不仔细看的话根本看不出来你的网站是基于github的；

```
<!--more-->
```

4. 数据绝对安全，基于github的版本管理，想恢复到哪个历史版本都行；
5. 博客内容可以轻松打包、转移、发布到其它平台；
6. 等等；

最终效果：

2016-08-18

HTML、CSS、JS中unicode字符的处理

前言

使用github pages服务搭建博客的好处有：

1. 全是静态文件，访问速度快；
2. 免费方便，不用花一分钱就可以搭建一个自由的个人博客，不需要服务器不需要后台；
3. 可以随意绑定自己的域名，不仔细看的话根本看不出来你的网站是基于github的；

[Read More](#)

- 1. 新增Hexo博客文章置顶功能

1. 新增Hexo博客文章置顶功能

博主博客地址

修改Hexo文件夹下的 `node_modules/hexo-generator-index/lib/generator.js`

需要添加的代码：

```
posts.data = posts.data.sort(function(a, b) {
    if(a.top && b.top) {
        if(a.top == b.top) return b.date - a.date;
        else return b.top - a.top;
    }
    else if(a.top && !b.top) {
        return -1;
    }
    else if(!a.top && b.top) {
        return 1;
    }
    else return b.date - a.date;
});
```

以下是最终的generator.js内容：

```
'use strict';

var pagination = require('hexo-pagination');

module.exports = function(locals) {
    var config = this.config;
    var posts = locals.posts.sort(config.index_generator.order_by);

    posts.data = posts.data.sort(function(a, b) {
        if(a.top && b.top) {
            if(a.top == b.top) return b.date - a.date;
            else return b.top - a.top;
        }
        else if(a.top && !b.top) {
            return -1;
        }
        else if(!a.top && b.top) {
            return 1;
        }
        else return b.date - a.date;
    });

    var paginationDir = config.pagination_dir || 'page';
    var path = config.index_generator.path || '';
}
```

```
return pagination(path, posts, {
  perPage: config.index_generator.per_page,
  layout: ['index', 'archive'],
  format: paginationDir + '/%d/',
  data: {
    __index: true
  }
});
});
```

如何使用：在需要置顶的文章添加top属性即可，排序从小到大

```
---
title: "Linux查看硬件信息"
date: 2018-03-17 16:06:43
tags: Linux
comments: true
top: 1
description: "Linux查看硬件信息"
---
```

博客效果：



Tips: 常用hexo命令

```
# hexo n == hexo new
# hexo g == 生成
# hexo s == 启动服务
```


- 1. Git分支的创建、删除、切换

1. Git分支的创建、删除、切换

1、本地创建新分支并切换到新分支

```
# git checkout -b 分支名
```

2、提交分支到远程仓库

```
# git push origin 分支名
```

3、切换分支

```
# git checkout 分支名
```

4、删除远程分支

```
# git push origin --delete 分支名
```

5、删除本地分支

```
# git branch -D 分支名
```

Tips：其他分支命令

- 查看当前分支 `git branch`
- 查看所属分支 `git branch -a`

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间：2019-07-02 13:44:30

- 1. Git彻底删除历史提交记录的方法

1. Git彻底删除历史提交记录的方法

1、查看Git提交记录

```
# git log
```

2、找到需要回滚到的提交点，复制它的hash值

```
$ git log
commit e1d617c9d0ba6e56a4d9f56ac9f26a3764673e43 (HEAD -> master, origin/master,
origin/HEAD)
Author: zhousiwei <2434387555@qq.com>
Date:   Wed Jun 5 10:13:06 2019 +0800

    添加笔芯gif

commit fa156673d27e8ed6c5b45cd3f116323ce8c4069c
Author: zhousiwei <2434387555@qq.com>
Date:   Mon May 13 17:32:05 2019 +0800

    添加百度网站统计
commit b7341b1fa2f3b1f26a3bd8680b99bbd3e6e9da84
Author: zhousiwei <2434387555@qq.com>
Date:   Mon May 13 16:29:51 2019 +0800
```

```
# git reset --hard 你复制的hash值
```

3、将当前指向的head推到git

```
# git push --force
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:30

- 1. GitBook的安装、卸载、常见问题
 - 1.1. 一、卸载GitBook(没有安装的跳过此步骤)
 - 1.2. 二、安装GitBook
 - 1.3. 三、GitBook插件
 - 1.4. 五、作者寄语

1. GitBook的安装、卸载、常见问题

Tips: 博主使用的版本

- 博主博客地址
- Node v8.11.2 npm v6.9.0 gitbook v3.2.3 服务器 CentOS6.5

官方GitHub:<https://github.com/GitbookIO/gitbook-cli>

GitBook相关配置及优化

1.1. 一、卸载GitBook(没有安装的跳过此步骤)

Tips: 前导摘要：记得有一次不小心卸载了一个东西，后面执行gitbook init报错：`Cannot find module 'internal/util/types'`

找到 `c:\Users\{User}\.gitbook` 找到并删除此文件夹

删除后执行命令

```
# npm uninstall -g gitbook
# npm uninstall -g gitbook-cli
--- 清除npm缓存
# npm cache clean --force
```

1.2. 二、安装GitBook

需要node环境 Linux或Win下安装node和npm 需要Git环境
<https://www.jianshu.com/p/f2da5e76a588>

```
# npm install gitbook -g
# npm install -g gitbook-cli
--- 如果没有安装gitbook,此命令会默认同时安装 GitBook
# gitbook -V
--- 列出本地所有的gitbook版本
# gitbook ls
```

Tips: **GitBook** 常用命令

- `gitbook -v` 查看版本号
 - `gitbook ls` 列出本地所有的gitbook版本
 - `gitbook init` 初始化
 - `gitbook install` 安装插件
 - `gitbook serve` 预览
 - `gitbook build` 生成
 - `gitbook build --gitbook=2.6.7` 生成时指定gitbook的版本, 本地没有会先下载
 - `gitbook uninstall 2.6.7` 卸载指定版本号的gitbook
 - `gitbook fetch [version]` 获取[版本]下载并安装<版本>
 - `gitbook --help` 显示gitbook-cli帮助文档
 - `gitbook help` 列出 gitbook 所有的命令
 - `gitbook ls-remote` 列出NPM上的可用版本:
-

1.3. 三、GitBook插件

1.4. 五、作者寄语

Everything is ok. But you have to do by yourself.

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:30

- 1. GitBook相关配置及优化

- 1.1.1. 1、GitBook配置说明
- 1.1.2. 2、GitBook生成电子书
- 1.1.3. 3、使用gitbook-convert
- 1.1.4. 4、GitBook插件列表
- 1.1.5. 5、博主示例book.json配置

1. GitBook相关配置及优化

1.1.1. 1、GitBook配置说明

1.1、目录结构

```

├── book.json
├── README.md
├── SUMMARY.md
├── chapter-1/
|   ├── README.md
|   └── something.md
└── chapter-2/
    └── README.md

```

1.2、GitBook 特殊文件的功能：

文件名	描述
book.json	配置数据 (optional)
README.md	电子书的前言或简介 (required)
SUMMARY.md	电子书目录 (optional)
GLOSSARY.md	词汇/注释术语列表 (optional)

1.3、book.json的配置说明

变量	描述
root	包含所有图书文件的根文件夹的路径，除了 book.json
structure	指定自述文件，摘要，词汇表等的路径
title	您的书名，默认值是从 README 中提取出来的。在 GitBook.com 上，这个字段是预填的。
description	您的书籍的描述，默认值是从 README 中提取出来的。在 GitBook.com 上，这个字段是预填的。
author	作者名。在 GitBook.com 上，这个字段是预填的。
isbn	国际标准书号 ISBN

language	本书的语言类型——ISO code。默认值是 en
direction	文本阅读顺序。可以是 rtl (从右向左) 或 ltr (从左向右)，默认值依赖于 language 的值。
gitbook	应该使用的GitBook版本，并接受类似于 >=3.0.0 的条件。
links	在左侧导航栏添加链接信息
plugins	要加载的插件列表(官网插件列表)
pluginsConfig	插件的配置

1.4、Gitbook 默认带有 5 个插件：

- highlight - 语法高亮插件
- search - 搜索插件
- sharing - 分享插件
- font-settings - 字体设置插件
- livereload - 热加载插件

1.5、去除自带插件

可以在插件名称前面加 -

```
"plugins": [
  "-search"
]
```

Tips：添加插件后要进行安装 `gitbook install`

1.6、设置PDF输出格式

变量	描述
pdf.pageNumbers	是否添加页码，默认是true
pdf.fontSize	字体大小，默认是12
pdf.fontFamily	字体，默认字体是Arial
pdf.paperSize	Paper size, options are a0 , a1 , a2 , a3 , a4 , a5 , a6 , b0 , b1 , b2 , b3 , b4 , b5 , b6 , legal (is a4)
pdf.margin.top	Top margin (default is 56)
pdf.margin.bottom	Bottom margin (default is 56)
pdf.margin.right	Right margin (default is 62)
pdf.margin.left	Left margin (default is 62)

1.1.2. 2、GitBook生成电子书

- 需要有calibre-ebook环境 [Linux下安装calibre-ebook](#)
- [GitBook生成pdf出现中文不显示或乱码](#)

```

--- 生成PDF文件
# gitbook pdf ./ ./mybook.pdf
--- 生成ePub文件
# gitbook epub ./ ./mybook.epub
--- 生成Mobi文件
# gitbook mobi ./ ./mybook.mobi

```

1.1.3. 3、使用gitbook-convert

说明：把 docx 、 xml 、 html 、 odt 文档转成 GitBook

- 官网地址：<https://github.com/GitbookIO/gitbook-convert>

```

# npm install gitbook-convert -g
--- 如果需要做软链
# ln -s /usr/local/node-v8.11.2-linux-x64/bin/gitbook-convert /usr/local/bin/gitbook-convert
--- 如何使用
# gitbook-convert [options] <file> [export-directory]

```

1.1.4. 4、GitBook插件列表

- `mygitalk` 基于gitalk的评论插件 <https://github.com/snowdreams1006/gitbook-plugin-mygitalk>
- `theme-default` GitBook的默认主题 <https://github.com/GitbookIO/theme-default>
- `autotheme` 自动换肤插件 <https://github.com/willin/gitbook-plugin-autotheme>
- `sharing` 默认的分享插件 <https://github.com/GitbookIO/plugin-sharing>
- `fontsettings` 默认的字体、字号、颜色设置插件 <https://github.com/GitbookIO/plugin-fontsettings>
- `highlight` 默认的代码高亮插件，通常会使用 prism 来替换 <https://github.com/GitbookIO/plugin-highlight>
- `search` 默认搜索插件 <https://github.com/GitbookIO/plugin-search>
- `search-plus` 支持中文搜索插件 <https://github.com/lwdgit/gitbook-plugin-search-plus>
- `prism` 基于 Prism 的代码高亮 <https://github.com/gaaron/gitbook-plugin-prism>
- `favicon` 更改网站的 favicon.ico <https://github.com/menduo/gitbook-plugin-favicon>
- `github` 在右上角显示 github 仓库的图标链接 <https://github.com/GitbookIO/plugin-github>
- `github-buttons` 显示 github 仓库的star和fork按钮 <https://github.com/azu/gitbook-plugin-github-buttons>
- `splitter` 在左侧目录和右侧内容之间添加一个可以拖拽的栏，用来调整两边的宽度 <https://github.com/yoshidax/gitbook-plugin-splitter>
- `copy-code-button` 为代码块添加复制的按钮 <https://github.com/WebEngage/gitbook-plugin-copy-code-button>
- `tbfed-pagefooter` 自定义页脚，显示版权和最后修订时间 <https://github.com/zhj3618/gitbook-plugin-tbfed-pagefooter>

- `tbfed-pagefooter` 自定义页脚，显示版权和最后修订时间 <https://github.com/zjh3618/gitbook-plugin-tbfed-pagefooter>
- `expandable-chapters` 收起或展开章节目录中的父节点
<https://github.com/DomainDrivenArchitecture/gitbook-plugin-expandable-chapters>
- `expandable-chapters-small` 比较好的折叠侧边栏 <https://github.com/lookdczar/gitbook-plugin-expandable-chapters-small-auto>
- `book-summary-scroll-position-saver` 自动保存左侧目录区域导航条的位置
<https://github.com/yoshidax/gitbook-plugin-book-summary-scroll-position-saver>
- `ga` 添加 Google 统计代码 <https://github.com/GitbookIO/plugin-ga>
- `sitemap` 生成站点地图 <https://github.com/GitbookIO/plugin-sitemap>
- `baidu` 使用百度统计 <https://github.com/poppinlp/gitbook-plugin-baidu>
- `Donate` Gitbook 捐赠打赏插件 <https://github.com/willin/gitbook-plugin-donate>
- `anchors` 标题带有 github 样式的锚点 <https://github.com/rilmv/gitbook-plugin-anchors>
- `anchor-navigation-ex` 插件锚导航-EX <https://github.com/zq99299/gitbook-plugin-anchor-navigation-ex>
- `theme-api` 编写 API 文档 <https://github.com/GitbookIO/theme-api>
- `katex` 使用KaTeX进行数学排版 <https://github.com/GitbookIO/plugin-katex>
- `editlink` 内容顶部显示编辑本页链接 <https://github.com/zhaoda/gitbook-plugin-editlink>
- `ad` 在每个页面顶部和底部添加广告或任何自定义内容 <https://github.com/zhaoda/gitbook-plugin-ad>
- `image-captions` 抓取内容中图片的 alt 或 title 属性，在图片下面显示标题
<https://github.com/todvora/gitbook-plugin-image-captions>
- `chart` 使用 C3.js 图表 <https://github.com/csbun/gitbook-plugin-chart>
- `styles-sass` 使用 SASS 替换 CSS <https://github.com/GitbookIO/plugin-styles-sass>
- `styles-less` 使用 LESS 替换 CSS <https://github.com/GitbookIO/plugin-styles-less>
- `disqus` 添加 disqus 评论插件 <https://github.com/GitbookIO/plugin-disqus>
- `latex-codecogs` 使用数学方程式 <https://github.com/GitbookIO/plugin-latex-codecogs>
- `mermaid` 使用流程图 <https://github.com/JozoVilcek/gitbook-plugin-mermaid>
- `atoc` 插入 TOC 目录 <https://github.com/willin/gitbook-plugin-atoc>
- `ace` 插入代码高亮编辑器 <https://github.com/ymcatar/gitbook-plugin-ace>
- `sectionx` 分离各个段落，并提供一个展开收起的按钮 <https://github.com/ymcatar/gitbook-plugin-sectionx>
- `mcqx` 交互式多选插件 <https://github.com/ymcatar/gitbook-plugin-mcqx>
- `include-codeblock` 通过引用文件插入代码 <https://github.com/azu/gitbook-plugin-include-codeblock>
- `fbqx` 使用填空题 <https://github.com/Erwin-Chan/gitbook-plugin-fbqx>
- `spoiler` 隐藏答案，当鼠标划过时才显示 <https://github.com/ymcatar/gitbook-plugin-spoiler>
- `anchor-navigation` 锚点导航 <https://github.com/yaneryou/gitbook-plugin-anchor-navigation>
- `youtubex` 插入 YouTube 视频 <https://github.com/ymcatar/gitbook-plugin-youtubex>
- `redirect` 重定向页面跳转 <https://github.com/ketan/gitbook-plugin-redirect>
- `duoshuo` 使用多说评论 <https://github.com/codepiano/gitbook-plugin-duoshuo>
- `jsfiddle` 插入 JSFiddle 组件 <https://github.com/Mavrin/gitbook-plugin-jsfiddle>
- `jsbin` 插入 JSBin 组件 <https://github.com/jcouyang/gitbook-plugin-jsbin>
- `Advanced Emoji` 支持emoji表情 <https://github.com/codeclou/gitbook-plugin-advanced-emoji>

- `Puml` 使用 PlantUML 展示 uml 图 <https://github.com/GitbookIO/plugin-puml>
- `Graph` 使用 function-plot 绘制数学函数图 <https://github.com/cjam/gitbook-plugin-graph>
- `Todo` 添加 Todo 功能 <https://github.com/ly-tools/gitbook-plugin-todo>
- `include-csv` 展示 csv 文件内容 <https://github.com/TakuroFukamizu/gitbook-plugin-include-csv>
- `musicxml` 支持 musicxml 格式的乐谱渲染 <https://github.com/ymcatar/gitbook-plugin-musicxml>
- `versions-select` 添加版本选择的下拉菜单, 针对文档有多个版本的情况
<https://github.com/prescottprue/gitbook-plugin-versions-select>
- `rss` 添加 rss 订阅功能 <https://github.com/denysdovhan/gitbook-plugin-rss>
- `multipart` 将书籍分成几个部分 <https://github.com/citizenmatt/gitbook-plugin-multipart>
- `url-embed` 嵌入动态内容 <https://github.com/basilvetas/gitbook-plugin-url-embed>

1.1.5.5、博主示例book.json配置

```
{
  "gitbook": "3.x.x",
  "plugins": [
    "favicon", "yahei", "github", "autotheme", "-lunr", "-search",
    "search-plus", "prism", "-highlight", "github-buttons", "splitter",
    "-sharing", "sharing-plus", "tbfed-pagefooter", "expandable-chapters-small",
    "book-summary-scroll-position-saver", "ga", "baidu", "donate",
    "anchors", "anchor-navigation-ex", "sitemap-general",
    "copy-code-button", "mygitalk", "multipart"
  ],
  "pluginsConfig": {
    "theme-default": {
      "showLevel": true
    },
    "fontSettings": {
      "family": "serif",
      "size": 2
    },
    "github": {
      "url": "https://github.com/JoeBling"
    },
    "autotheme": {
      "white": [9, 10, 11, 12, 13, 14, 15, 16, 17],
      "sepia": [6, 7, 8, 18, 19],
      "night": [20, 21, 22, 23, 0, 1, 2, 3, 4, 5]
    },
    "prism": {
      "css": [
        "prismjs/themes/prism-tomorrow.css"
      ]
    },
    "github-buttons": {
      "buttons": [
        {
          "user": "JoeBling",
          "repo": "myBook",
          "type": "star",
          "size": "small"
        }
      ]
    }
  }
}
```

```
},
"sharing": {
    "douban": false,
    "facebook": false,
    "google": true,
    "hatenaBookmark": false,
    "instapaper": false,
    "line": true,
    "linkedin": true,
    "messenger": false,
    "pocket": false,
    "qq": false,
    "qzone": true,
    "stumbleupon": false,
    "twitter": false,
    "viber": false,
    "vk": false,
    "weibo": true,
    "whatsapp": false,
    "all": [
        "facebook", "google", "twitter",
        "weibo", "instapaper", "linkedin",
        "pocket", "stumbleupon"
    ]
},
"tbfed-pagefooter": {
    "copyright": "Copyright &copy <a href='https://zhousiwei.gitee.io' target='_blank'>試毅-思伟</a> 2019",
    "modify_label": "修订时间：",
    "modify_format": "YYYY-MM-DD HH:mm:ss"
},
"ga": {
    "token": "UA-142790104-1"
},
"baidu": {
    "token": "f1f256f18b7899e11647d6ec1f3ded2f"
},
"donate": {
    "wechat": "/assets/weixin.png",
    "alipay": "/assets/alipay.jpeg",
    "title": "",
    "button": "赏",
    "alipayText": "支付宝打赏",
    "wechatText": "微信打赏"
},
"anchor-navigation-ex": {
    "showLevel": true,
    "associatedWithSummary": false,
    "printLog": false,
    "multipleH1": true,
    "mode": "float",
    "showGoTop": true,
    "float": {
        "y": 100
    }
}
```

```
        "floatIcon": "fa fa-navicon",
        "showLevelIcon": false,
        "level1Icon": "fa fa-hand-o-right",
        "level2Icon": "fa fa-hand-o-right",
        "level3Icon": "fa fa-hand-o-right"
    },
    "pageTop": {
        "showLevelIcon": false,
        "level1Icon": "fa fa-hand-o-right",
        "level2Icon": "fa fa-hand-o-right",
        "level3Icon": "fa fa-hand-o-right"
    }
},
"sitemap-general": {
    "prefix": "https://zhousiwei.gitee.io/myBook/"
},
"favicon": {
    "shortcut": "/assets/images/favicon.ico",
    "bookmark": "/assets/images/favicon.ico",
    "appleTouch": "/assets/images/apple-touch-icon.png",
    "appleTouchMore": {
        "120x120": "/assets/images/apple-touch-icon-120x120.png",
        "180x180": "/assets/images/apple-touch-icon-180x180.png"
    }
},
"mygitalk": {
    "clientID": "608e351b48e4d0a3a528",
    "clientSecret": "****加密****",
    "repo": "myBook",
    "owner": "JoeyBling",
    "admin": ["JoeyBling"],
    "distractionFreeMode": false
},
"pdf": {
    "pageNumbers": true,
    "fontFamily": "Arial",
    "fontSize": 12,
    "paperSize": "a4",
    "margin": {
        "right": 62,
        "left": 62,
        "top": 56,
        "bottom": 56
    }
},
"title": "試毅-思伟_开源项目",
"description": "試毅-思伟_开源项目",
"author": "試毅-思伟",
"language": "zh-hans",
"root": ".",
"structure": {
    "readme": "introduction.md"
```

```
"structure": {  
    "readme": "introduction.md"  
},  
"links": {  
    "sidebar": {  
        "GitHub": "https://github.com/JoeyBling",  
        "个人博客": "https://zhousiwei.gitee.io"  
    }  
}
```



Tips: 添加插件后要进行安装 `gitbook install`

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 17:47:57

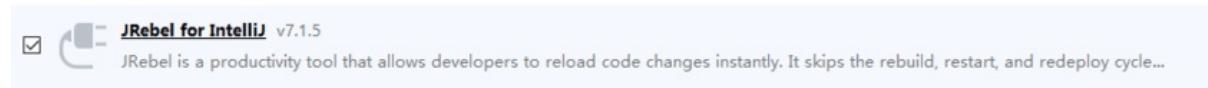
- 1. IDEA+JRebel全自动热部署

1. IDEA+JRebel全自动热部署

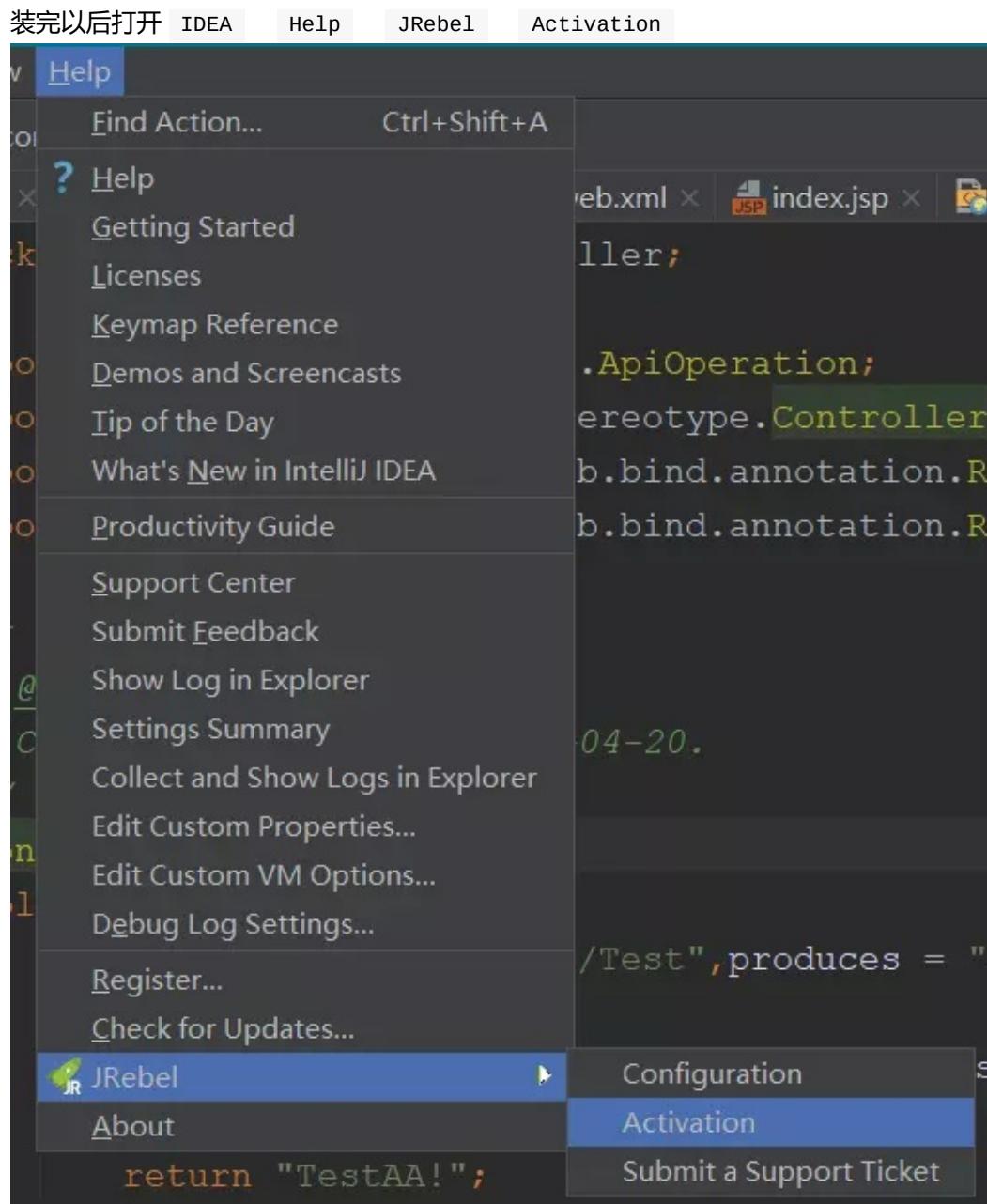
- 链接: <https://pan.baidu.com/s/1ZE5cv1vhFQhdaj91dFD-GA> 密码: b2k8 下载后解压

11ReverseProxy_windows_amd64.exe	2018/7/23 15:18	应用程序	1,50
jr-ide-intellij-7.1.5_13-17.zip	2018/3/29 8:57	好压 ZIP 压缩文件	38,62
ReverseProxy_darwin_amd64	2018/3/29 8:53	文件	1,46
ReverseProxy_windows_amd64.exe	2018/3/29 8:53	应用程序	1,30

1、打开IDEA离线安装插件，选择刚刚下载的ZIP就行了

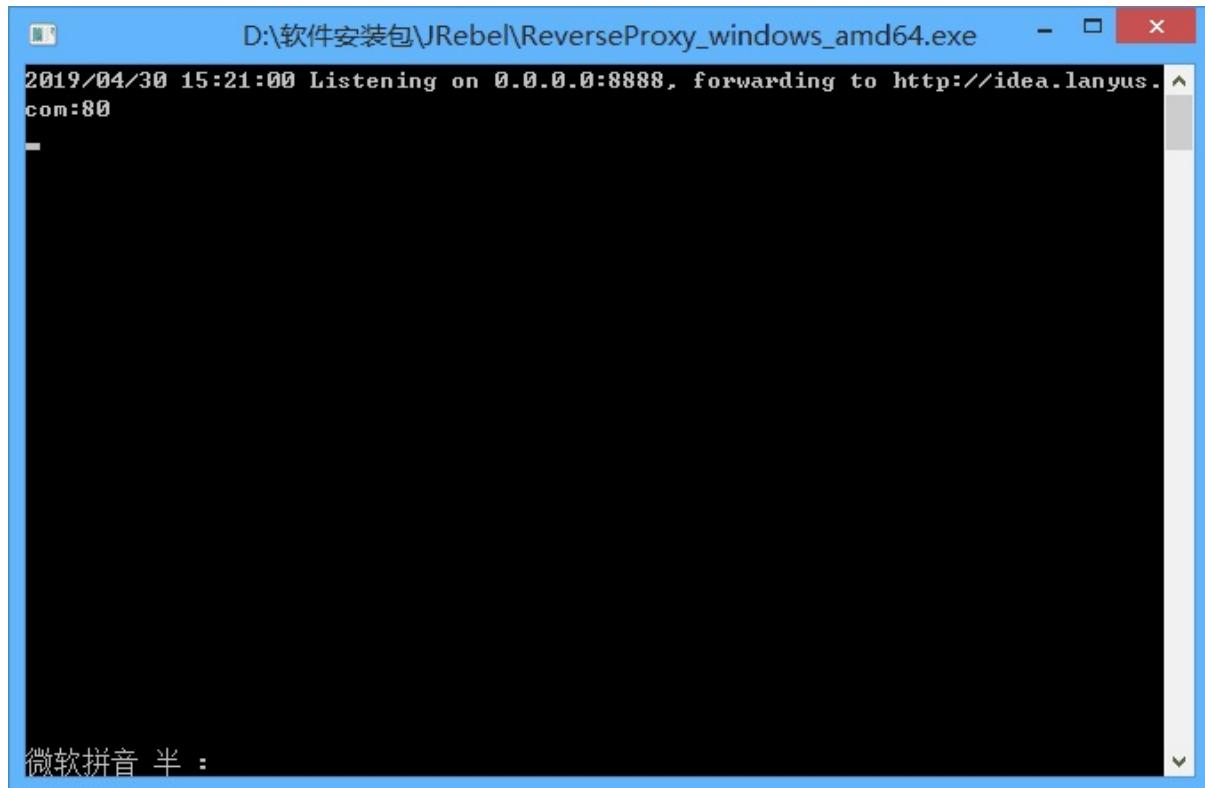


2、激活JRebel

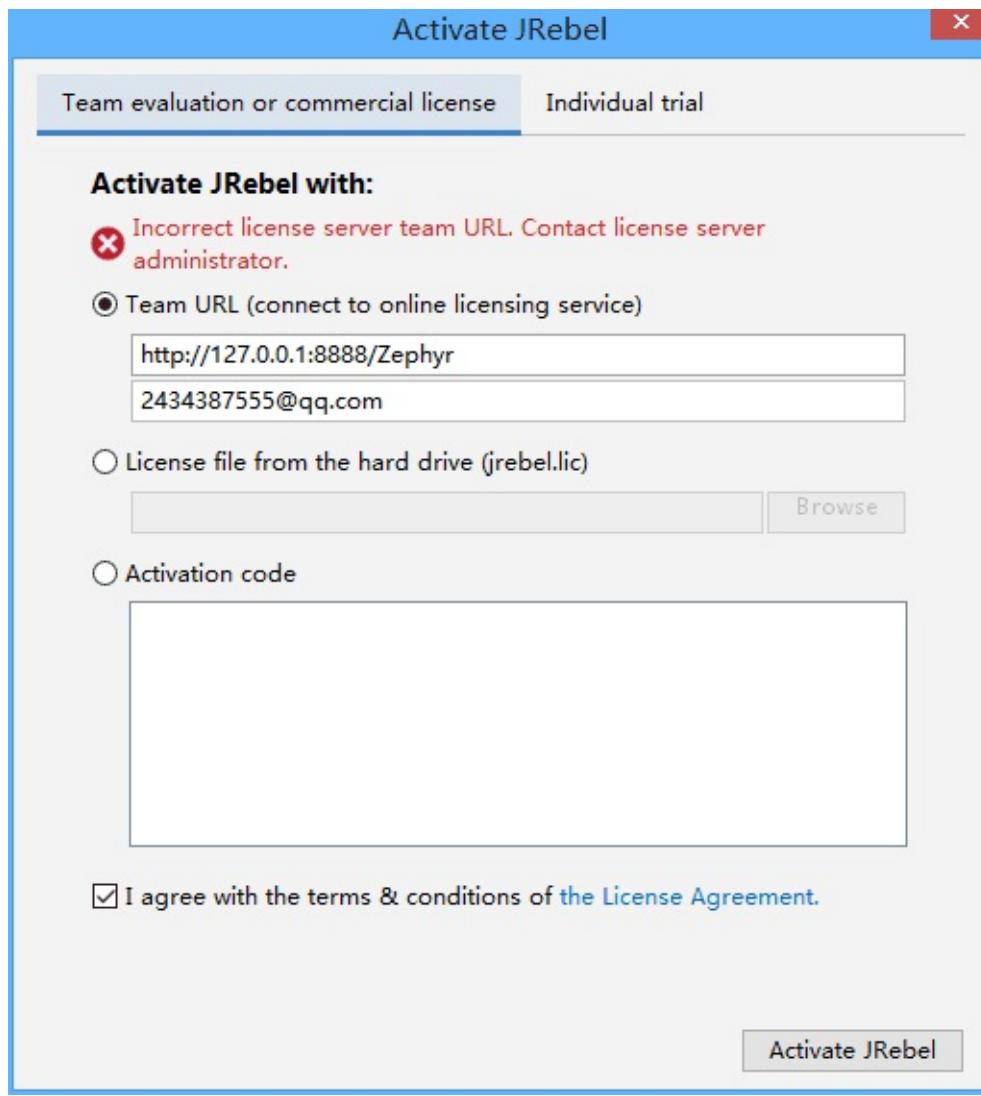


运行激活程序

11ReverseProxy_windows_amd64.exe	2018/7/23 15:18	应用程序	1,50	
jr-ide-intellij-7.1.5_13-17.zip	2018/3/29 8:57	好压 ZIP 压缩文件	38,6%	
ReverseProxy_darwin_amd64	2018/3/29 8:53	文件	1,40	
ReverseProxy_windows_amd64.exe	2018/3/29 8:53	应用程序	1,30	

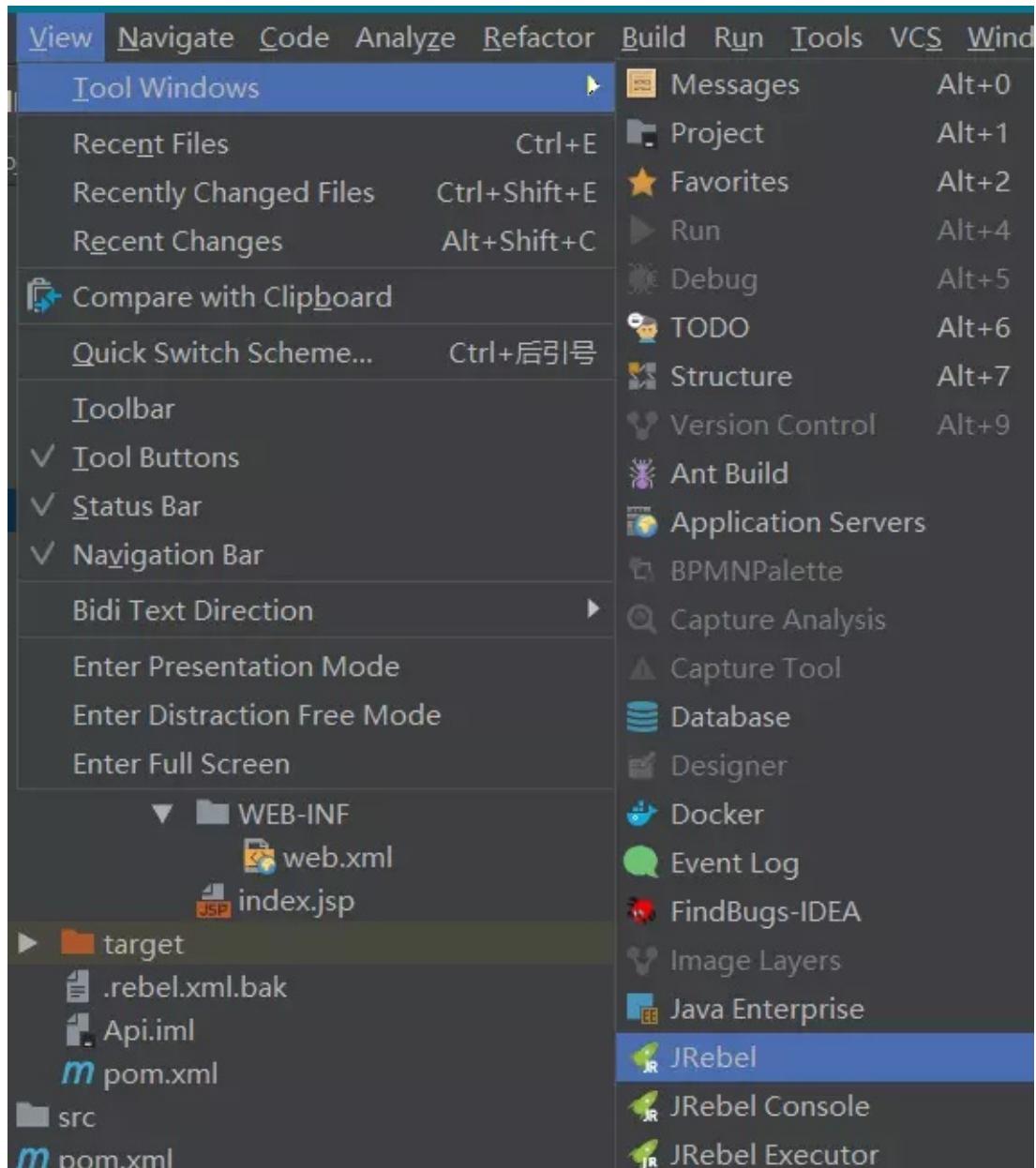


地址填写 `http://127.0.0.1:8888/Zephyr` 邮箱随便填

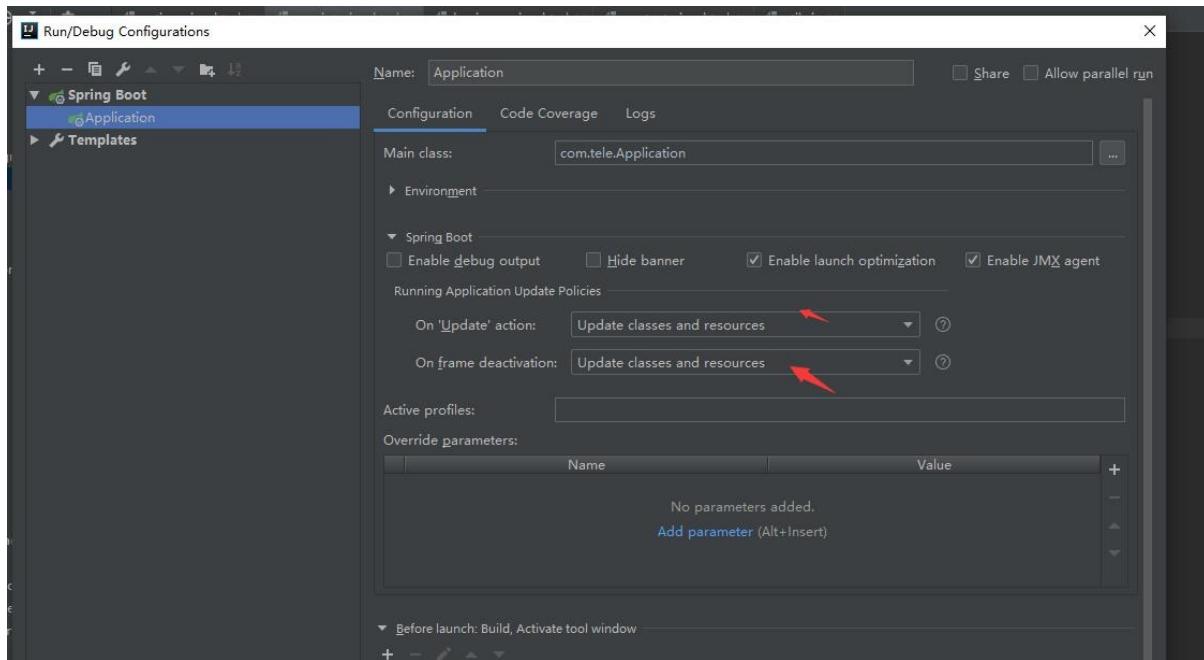


3、使用及自动部署

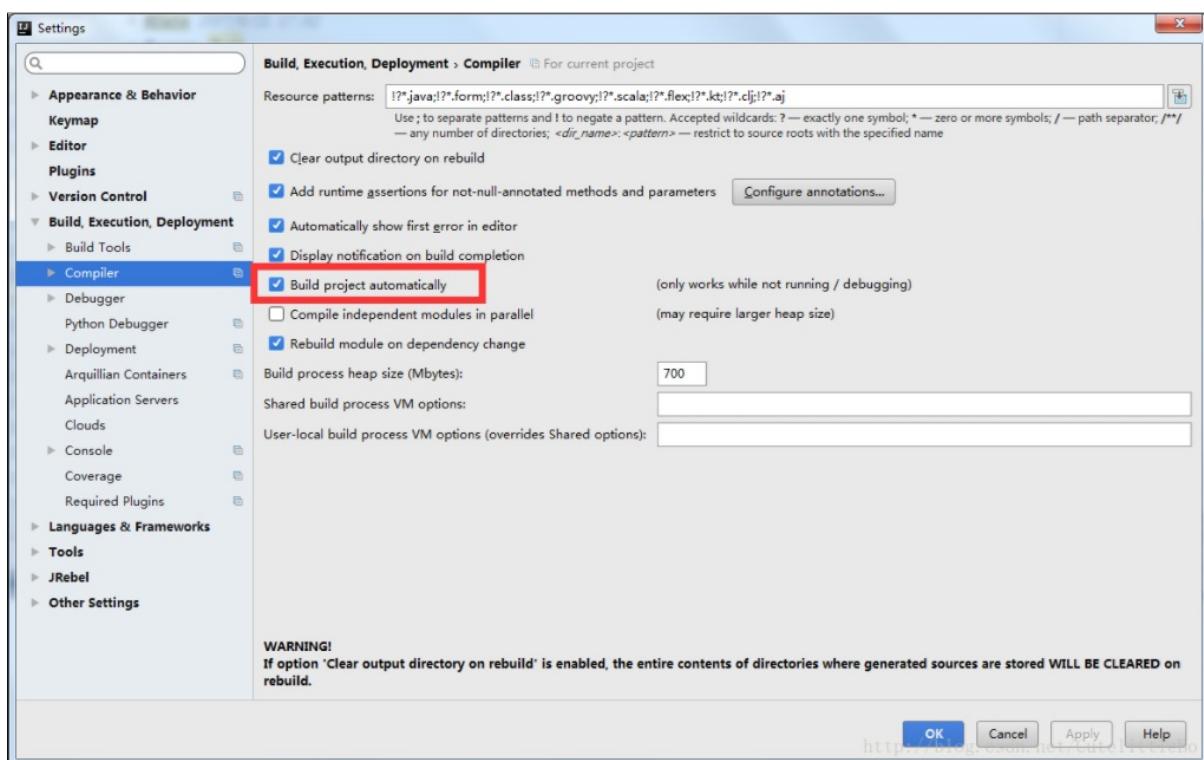
依次打开 view Tool Windows JRebel



这时候已经实现热部署了,不过需要每次修改源码后手动执行UPDATE操作,我们可以结合Idea自身的Debug Hot swap功能,在 运行和调试配置 中,配置自动执行UPDATE操作



4、勾选自动编译功能



Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:30

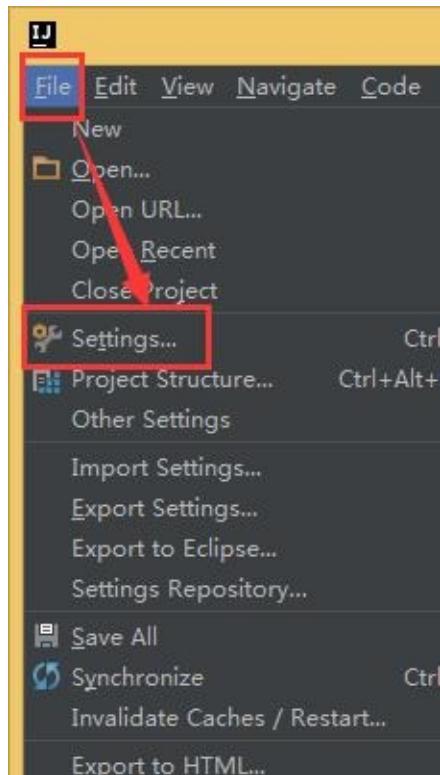
- 1. IDEA安装lombok及使用详解
 - 1.1.1. 2、添加lombok的maven的pom.xml依赖：
 - 1.1.2. 3、示例代码Student.java

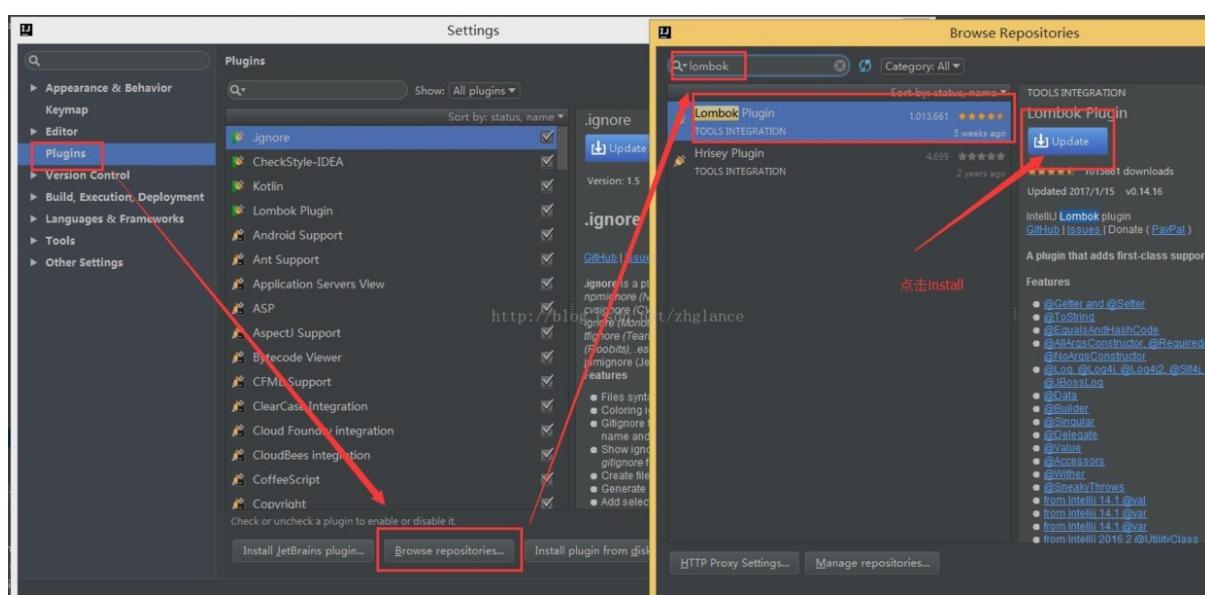
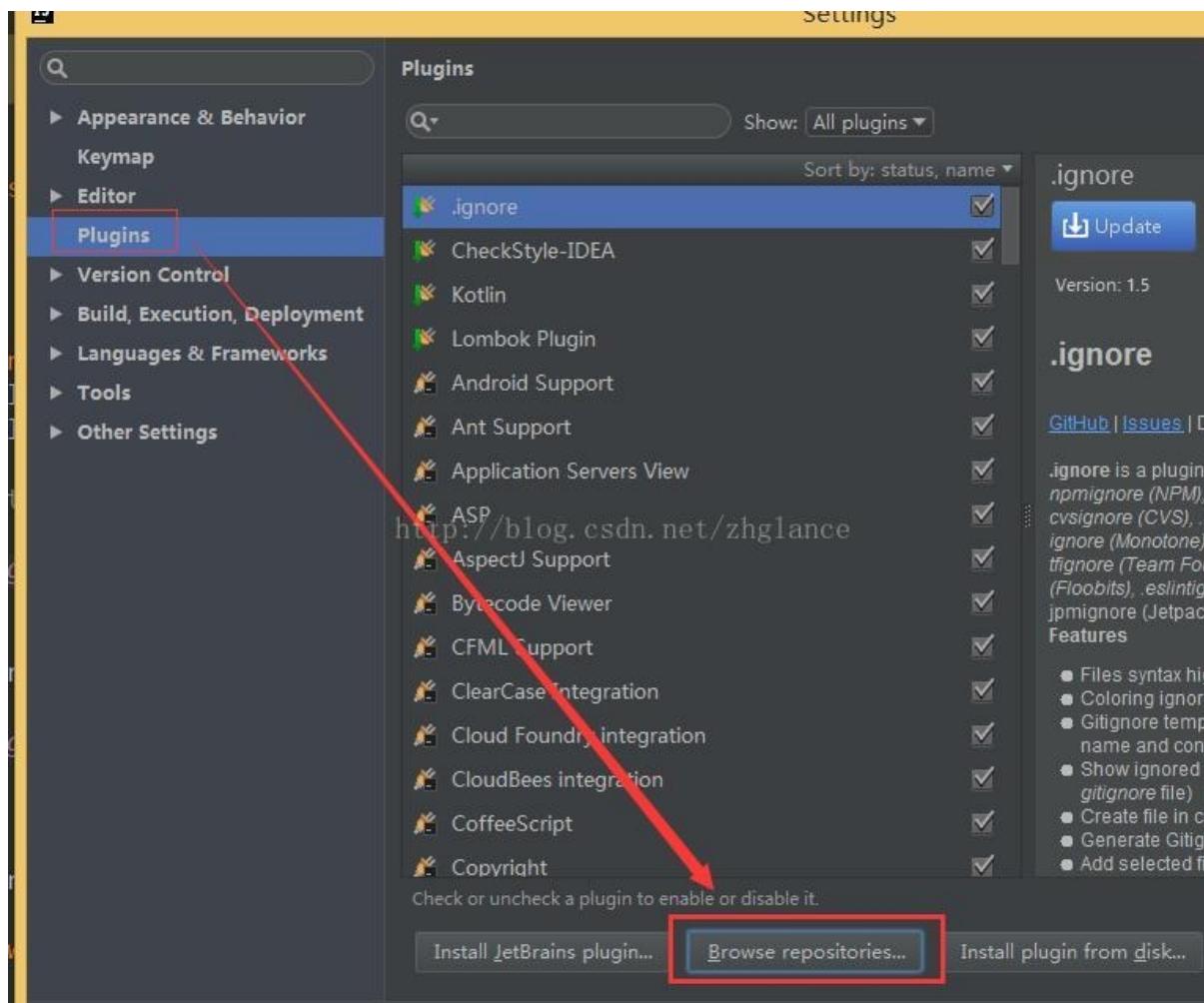
1. IDEA安装lombok及使用详解

项目中经常使用bean, entity等类，绝大部分数据类类中都需要get、set、toString、equals和hashCode方法，虽然eclipse和idea开发环境下都有自动生成的快捷方式，但自动生成这些代码后，如果bean中的属性一旦有修改、删除或增加时，需要重新生成或删除get/set等方法，给代码维护增加负担。而使用了lombok则不一样，使用了lombok的注解

(@Setter , @Getter , @ToString , @RequiredArgsConstructor , @EqualsAndHashCode 或 @Data)之后，就不需要编写或生成get/set等方法，很大程度上减少了代码量，而且减少了代码维护的负担。故强烈建议项目中使用 lombok，去掉bean中get、set、toString、equals和hashCode等方法的代码。

1、安装lombok插件





1.1.1. 2、添加lombok的maven的pom.xml依赖：

```
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
```

```
<version>1.16.10</version>
</dependency>
```

1.1.2. 3、示例代码Student.java

```
package com.lombok.demo;

import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.Setter;
import lombok.ToString;

/**
 * Created by zhousiwei on 2017/2/8
 */
@Data
public class Student {
    private String name;
    private int age;
    private String male;
    private String studentNo;
}
```

Tips: `@Data` 注解的作用相当

于 `@Getter` , `@Setter` , `@RequiredArgsConstructor` , `@ToString` , `@EqualsAndHashCode` 的合集。

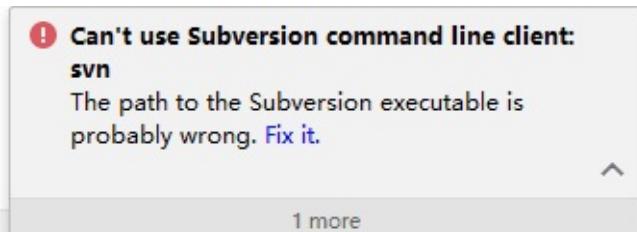
Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:31

- 1. IDEA设置后端代码开发规范

1. IDEA设置后端代码开发规范

1、IDEA 配置svn

提交svn报错：



1.1、下载：<https://www.visualsvn.com/downloads/>



Apache Subversion command line tools

A standalone and lightweight package to install Apache Subversion command line binaries only. This redistributable package is great for build scripting and other administrative needs.

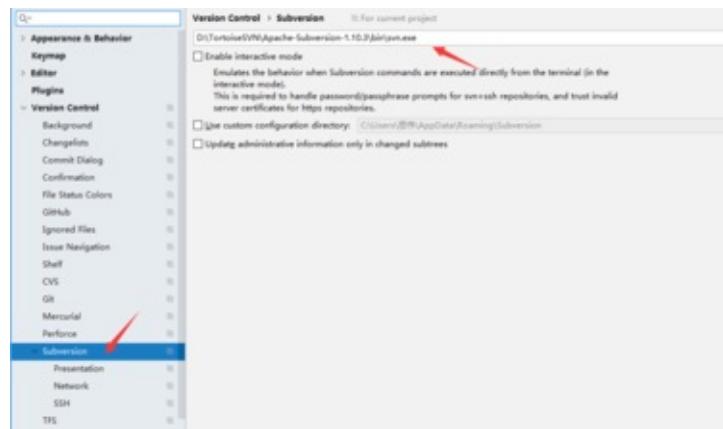
[Read more about Subversion command line tools](#)

[Download](#)

Version: 1.10.3

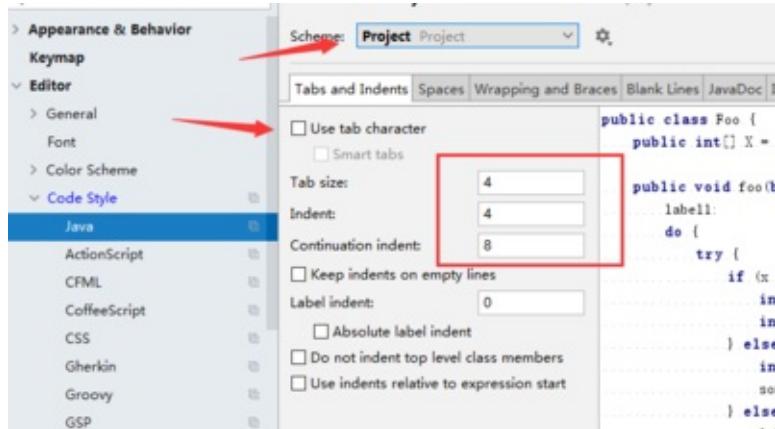
Size: ~3 MB

1.2、修改IDEA配置指向的路径：

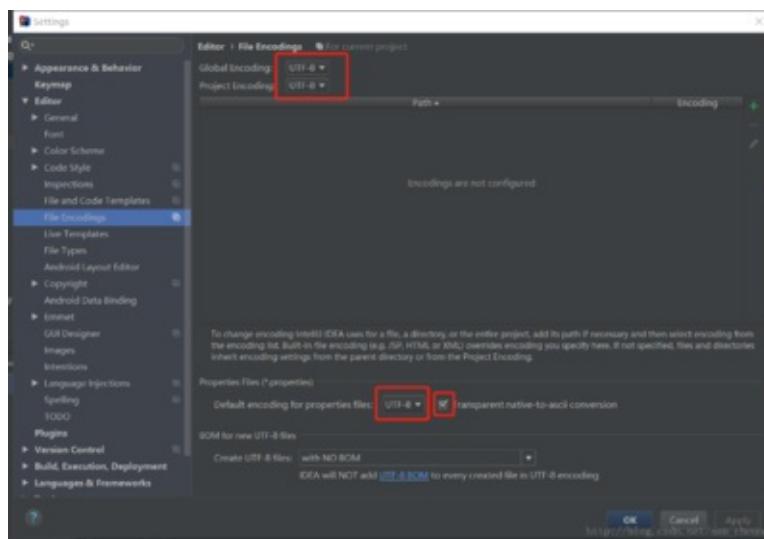


2、IDEA 设置 Tab 键为 4 个空格

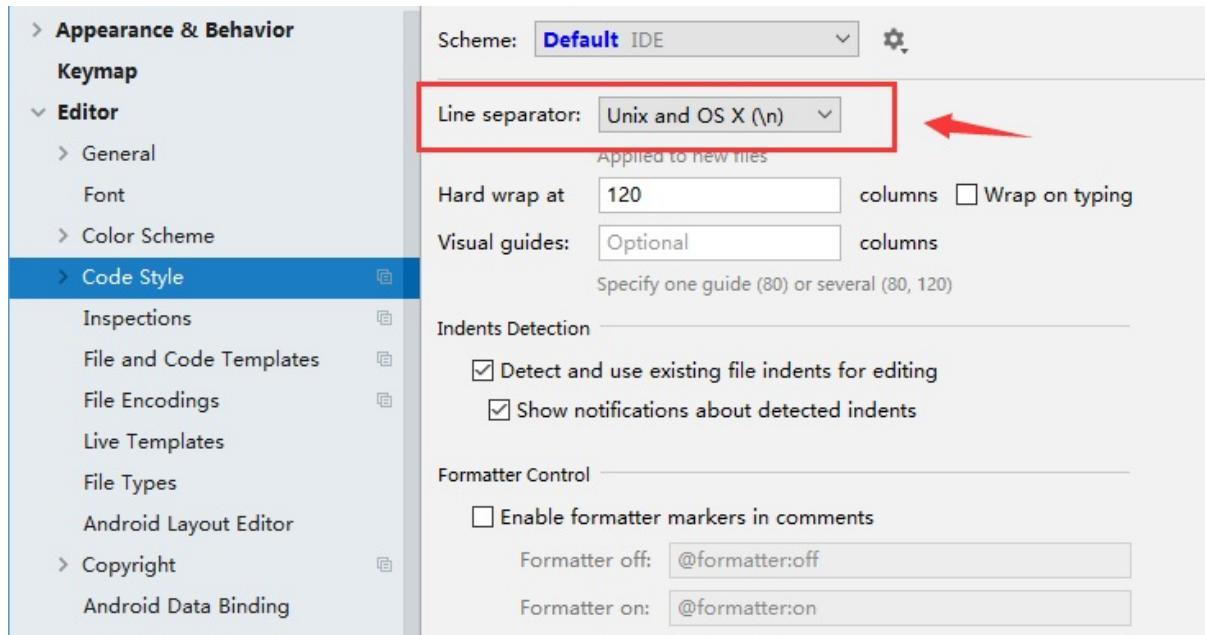
Scheme 选择 **Project**, 在下面把取消勾选 **Use tab character**



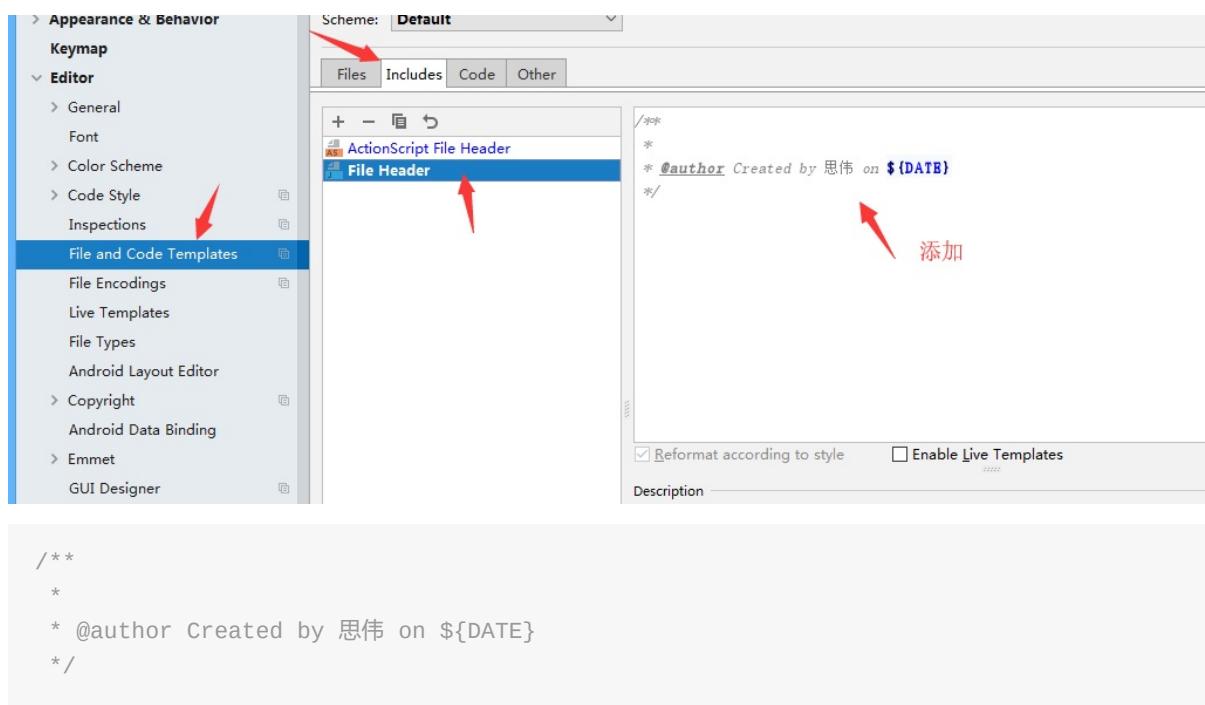
3、IDEA 设置编码格式为 UTF-8



4、IDEA 设置换行符为 Unix 编码格式



5、IDEA 设置类注释模板



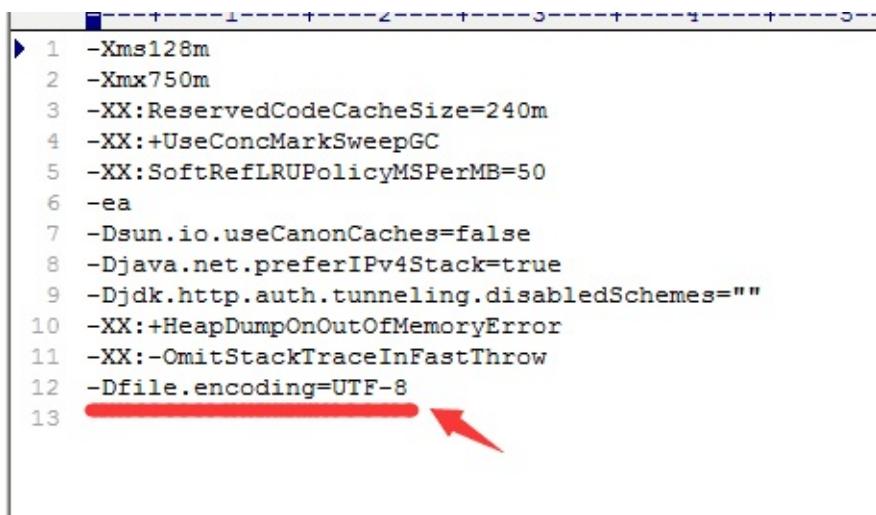
Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:31

- 1. 解决Tomcat在IDEA控制台中文乱码的问题

1. 解决Tomcat在IDEA控制台中文乱码的问题

在idea的安装目录下的 `bin/idea.exe.vmoptions` 文件和 `idea64.exe.vmoptions` 文件的末尾另起一行添加

```
-Dfile.encoding=UTF-8
```



```
1 -Xms128m
2 -Xmx750m
3 -XX:ReservedCodeCacheSize=240m
4 -XX:+UseConcMarkSweepGC
5 -XX:SoftRefLRUPolicyMSPerMB=50
6 -ea
7 -Dsun.io.useCanonCaches=false
8 -Djava.net.preferIPv4Stack=true
9 -Djdk.http.auth.tunneling.disabledSchemes=""
10 -XX:+HeapDumpOnOutOfMemoryError
11 -XX:-OmitStackTraceInFastThrow
12 -Dfile.encoding=UTF-8
13
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:31

- 1. 玩转MySQL优化命令

1. 玩转MySQL优化命令

1、MySQL更改为只能本地登录

```
# use mysql;
# select user,host,Password from user where user='root';
# delete from user where user='root' and host='%';
```

2、MySQL授予外网登陆权限

```
grant all privileges on *.* to 授予外网登陆的用户名@'%' identified by '密码' WITH GRANT OPTION;
```

```
# grant all privileges on *.* to root@'%' identified by 'root' WITH GRANT OPTION;
# flush privileges;
```

3、更改MySQL的存储目录

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:31

- 1. BitComet服务器列表

1. BitComet服务器列表

udp://mgtracker.org:2710/announce <http://share.camoe.cn:8080/announce> udp://11.rarbg.me:80/announce
<http://tracker.tfile.me/announce> <http://open.acgtracker.com:1096/announce>
<http://mgtracker.org:2710/announce> udp://tracker.ex.ua:80/announce
<http://tracker.longwangbt.com:2710/announce> udp://tracker.openbittorrent.com:80/announce
<http://tracker.publicbt.com/announce> <http://tracker.prq.to/announce> udp://9.rarbg.me:2710/announce
<http://tracker1.wasabii.com.tw:6969/anno> <http://tracker2.wasabii.com.tw:6969/announce>
udp://explodie.org:6969/announce udp://shadowshq.eddie4.nl:6969/announce
udp://shadowshq.yi.org:6969/announce udp://eddie4.nl:6969/announce
udp://tracker.publicbt.com:80/announce udp://tracker.istole.it:80/announce
<http://trackers.ibzu.me/announce.php> <http://tracker1.torrentino.com/announce>
<http://tracker2.torrentino.com/announce> <http://tracker3.torrentino.com/announce>
<http://bt.careland.com.cn:6969/announce> <http://bt2.careland.com.cn:6969/announce>
<http://bt3.careland.com.cn:6969/announce> <http://94.228.192.98/announce>
<http://121.14.98.151:9090/announce> <http://henbt.com:2710/announce>
<http://server1.9sheng.cn:6969/announce> <http://anisaishuu.de:2710/announce> udp://tracker.leechers-paradise.org:6969/announce <http://tracker.shuntv.net/announce.php>
<http://tracker.tvunderground.org.ru:3218/announce> udp://10.rarbg.me:80/announce
udp://9.rarbg.com:2710/announce udp://tracker.yify-torrents.com/announce
udp://tracker.coppersurfer.tk:6969/announce <http://siambit.org/announce.php>
udp://open.demonii.com:1337/announce udp://tracker.torrenty.org:6969/announce
<http://tracker.torrenty.org:6969/announce> udp://tracker.publichd.eu:80/announce
<http://tracker.pimp4003.net/announce> <http://tracker.xfsub.com:6868/announce>
<http://mgtracker.org:6969/announce> <http://tr.cili001.com:6666/announce>
udp://tr.cili001.com:6666/announce <http://tracker.openbittorrent.com/announce>
udp://tracker.opentrackr.org:1337/announce udp://open.demonii.com:1337
udp://9.rarbg.com:2720/announce <http://tracker1.wasabii.com.tw:6969/announce>

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:30

- 1. Linux下安装GLIBC_2-15
 - 1.1.1. 1、查看系统glibc支持的版本
 - 1.1.2. 2、升级glibc支持的版本到GLIBC_2.15
 - 1.1.3. 3、升级glibc挂了后可以救命的命令

1. Linux下安装GLIBC_2-15

1.1.1. 1、查看系统glibc支持的版本

```
# strings /lib64/libc.so.6 |grep GLIBC
# rpm -qa | grep glibc
```

1.1.2. 2、升级glibc支持的版本到GLIBC_2.15

官网地址 <http://www.gnu.org/software/libc/> 官网所有安装包 <http://ftp.gnu.org/gnu/glibc/>

```
# cd /usr/local
# wget http://ftp.gnu.org/gnu/libc/glibc-2.15.tar.xz
# tar -xvf glibc-2.15.tar.xz
# mkdir -p /var/VMdisks
# mv glibc-2.15 /var/VMdisks/
# cd /var/VMdisks/glibc-2.15/
# mkdir build && cd build
--- 不要将 glibc 安装到默认的目录【/usr/local】或者 【任何自定义的目录】，请务必安装到【/usr】目录
# ./configure --prefix=/usr --disable-profile --enable-add-ons --with-headers=/usr/include --with-binutils=/usr/bin
# make -j4
# make install
--- 解决中文乱码问题
# make localesdata/install-locales
```

检查验证

```
# ll /lib64/libc*
--- 查看系统glibc支持的版本
# strings /lib64/libc.so.6 |grep GLIBC
# /lib64/libc.so.6
```

1.1.3. 3、升级glibc挂了后可以救命的命令

如果实在不幸，在升级glibc时挂掉了，执行各种命令都提示错误，比如：

- Segmentation fault
- error while loading shared libraries: libc.so.6: cannot open shared object file: No such file or directory

这类错误出现千万不要着急退出SSH，执行下面的命令是可以挽救的：

```
# cd /lib64
# LD_PRELOAD=/lib64/libc-2.12.so ln -sf /lib64/libc-2.12.so libc.so.6
```

Tips： `libc-2.12.so` 这个文件名根据你系统中的文件而定，如果有多个版本so文件可以逐个尝试

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-02 13:44:28

- 1. Linux安装python2.7.9
 - 1.1.1. 1、Linux下安装(其他系统请自行前往官网安装)
 - 1.1.2. 2、python解释器指向python2.7.9
 - 1.1.3. 3、解决python -V执行报错
 - 1.1.4. 4、由于yum使用了原来python2.6的一些脚本，需要将yum重新导向原来的python2.6解释器才可以使用

1. Linux安装python2.7.9

官网地址：<https://www.python.org/downloads/>

1.1.1. 1、Linux下安装(其他系统请自行前往官网安装)

```
--- 安装依赖包
# cd /usr/local
# yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gcc make -y
# wget https://www.python.org/ftp/python/2.7.9/Python-2.7.9.tgz
# tar -xzvf Python-2.7.9.tgz
# cd Python-2.7.9
# ./configure --prefix=/usr/local/Python2.7.9 --enable-shared -enable-unicode=ucs4
# make
# make install
# python -V
```

1.1.2. 2、python解释器指向python2.7.9

```
# which python
# cd /usr/bin
# rm python
# ln -s /usr/local/Python2.7.9/bin/python python
# python -V --执行报错
```

1.1.3. 3、解决python -V执行报错

```
# cd /etc/ld.so.conf.d/
# echo "/usr/local/Python2.7.9/lib" > python2.7.conf
# ldconfig
# python -V
```

1.1.4. 4、由于yum使用了原来python2.6的一些脚本，需要将yum重新导向原来的python2.6解释器才可以使用

```
# which yum
```

```
# vi /usr/bin/yum  
----修改第一行路径为: #!/usr/bin/python2.6
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:29

- 1. Linux下安装rabbitMq
 - 1.1.1. Linux下安装rabbitMq

1. Linux下安装rabbitMq

1.1.1. Linux下安装rabbitMq

必须有Erlang环境----->>>[Linux下安装Erlang](#)

一、安装rabbitMq

```
# cd /usr/local/  
# yum install -y xz  
# wget http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.2/rabbitmq-server-generic-unix-3.6.2.tar.xz  
# xz -d rabbitmq-server-generic-unix-3.6.2.tar.xz  
# tar -xvf rabbitmq-server-generic-unix-3.6.2.tar  
--配置rabbitMq环境变量  
# vim /etc/profile  
文件末尾添加 export PATH=$PATH:/usr/local/rabbitmq_server-3.6.2/sbin  
# source /etc/profile
```

二、配置网页插件

```
# rabbitmq-plugins enable rabbitmq_management  
--启动rabbitMq  
# rabbitmq-server -detached  
Tips:rabbitmq默认会创建guest账号，只能用于localhost登录页面管理员  
--添加admin用户并且密码为admin  
# rabbitmqctl add_user admin admin  
--授权admin用户为管理员  
# rabbitmqctl set_user_tags admin administrator  
--授权访问权限  
# rabbitmqctl set_permissions -p "/" admin ".*" ".*"
```

三、验证rabbitMq是否安装成功

- 端口：15672 网页管理， 5672 AMQP端口
 - 访问 <http://IP:15672>
-

四、rabbitMq常用命令：

- 启动： `rabbitmq-server -detached`
- 查看服务状态： `rabbitmqctl status`

- 关闭服务: `rabbitmqctl stop`
- 查看所有用户: `rabbitmqctl list_users`
- 添加用户: `rabbitmqctl add_user admin admin`
- 授权用户管理员: `rabbitmqctl set_user_tags admin administrator`
- 授权访问权限: `rabbitmqctl set_permissions -p "/" "admin" "." ".*" "./*"`
- 添加虚拟机: `rabbitmqctl add_vhost admin_vhost`
- 授权用户到虚拟机: `rabbitmqctl set_permissions -p admin_vhost admin ".*" ".*" "./*"`

Tips: RabbitMQ与Erlang的版本对应关系图

RabbitMQ and Erlang/OTP Compatibility Matrix

<u>RabbitMQ version</u>	Minimum required Erlang/OTP	Maximum supported Erlang/OTP	Notes
<ul style="list-style-type: none"> • 3.7.13 • 3.7.12 • 3.7.11 • 3.7.10 • 3.7.9 • 3.7.8 • 3.7.7 	• 20.3.x	• 21.3.x	<ul style="list-style-type: none"> Erlang/OTP 19.3.x support was discontinued as of Jan 1st, 2019 For the best TLS support, the latest version of Erlang/OTP 21.x is recommended On Windows, Erlang/OTP 20.2 changed default cookie file location
<ul style="list-style-type: none"> • 3.7.6 • 3.7.5 • 3.7.4 • 3.7.3 • 3.7.2 • 3.7.1 • 3.7.0 	• 19.3	• 20.3.x	<ul style="list-style-type: none"> For the best TLS support, the latest version of Erlang/OTP 20.3.x is recommended Erlang versions prior to 19.3.6.4 have known bugs (e.g. ERL-430, ERL-448) that can prevent RabbitMQ nodes from accepting connections (including from CLI tools) and stopping Versions prior to 19.3.6.4 are vulnerable to the ROBOT attack (CVE-2017-1000385) On Windows, Erlang/OTP 20.2 changed default cookie file location

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:29

- 1. Linux下安装Erlang
 - 1.1.1. Linux下安装Erlang

1. Linux下安装Erlang

1.1.1. Linux下安装Erlang

一、升级openssl

```
# cd /usr/local/
# wget http://www.openssl.org/source/openssl-1.0.1s.tar.gz
# tar -zxvf openssl-1.0.1s.tar.gz
# cd openssl-1.0.1s
# ./config --prefix=/usr/local/openssl
# vim Makefile
--修改CFLAG=-DOPENSSL_THREADS 修改成 CFLAG= -fPIC -DOPENSSL_THREADS
--也就是添加 -fPIC
# make && make install
```

CC= gcc
CFLAG= -fPIC -DOPENSSL_THREADS
OPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2M512
DEPENAG -DOPENSSL_NO_EC_NISTP_64BIT

二、安装Erlang

```
# cd /usr/local/
# wget http://erlang.org/download/otp_src_18.3.tar.gz
# tar -zxvf otp_src_18.3.tar.gz
# cd otp_src_18.3
--创建Erlang安装目录
# mkdir /usr/local/erlang
--Erlang安装依赖
# yum install -y gcc gcc-c++ unixODBC-devel openssl-devel ncurses-devel
# ./configure --prefix=/usr/local/erlang
# make && make install
--配置Erlang环境变量
# vim /etc/profile
文件末尾添加 export PATH=$PATH:/usr/local/erlang/bin
# source /etc/profile
```

三、验证erlang是否安装成功

```
# erl
# ssl:versions().
# halt().
```

```
[root@normal otp_src_18.3]# erl
Erlang/OTP 18 [erts-7.3] [source] [64-bit] [smp:2:2] [async-threads:10] [hipe] [kernel-poll:false]

Eshell V7.3  (abort with ^G)
1> ssl:versions().
[{ssl_app,"7.3"},{supported,['tlsV1.2','tlsV1.1',tlsV1]}, {available,['tlsV1.2','tlsV1.1',tlsV1,sslV3]}]
2> halt().
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:27

- 1. Linux常用命令

- 1.1.1. 目录处理命令
- 1.1.2. 文件处理命令
- 1.1.3. 权限管理命令
- 1.1.4. 其他权限命令
- 1.1.5. 文件搜索命令
- 1.1.6. 帮助命令
- 1.1.7. 用户管理命令
- 1.1.8. 压缩解压命令
- 1.1.9. 网络命令
- 1.1.10. 关机重启命令

1. Linux常用命令

命令格式与目录处理命令 `ls`

命令格式与目录处理命令 `ls`

命令格式: 命令 [-选项][参数]

例: `ls -la /etc`

说明:

- 1) 个别命令使用不遵循格式
- 2) 当有多个选项时, 可以写在一起
- 3) 简化选项与完整选项
`-a` 等于 `--all`

目录处理命令 `ls`

命令名称: `ls`

命令因为原意: list

命令所在路径: `/bin/ls`

执行权限: 所有用户

功能描述: 显示目录文件

语法: `ls 选项[-ald][文件或目录]`

- a 显示所有文件, 包括隐藏文件
- l 详细信息显示
- d 查看目录属性

`-rwxr--r-`

文件类型 (-文件 d目录 l软链接文件)

`rwx r--r-`

`u g o`

u所有者 g所属组 o其他人

r读 w写 x执行

1.1.1. 目录处理命令

命令名称: `mkdir`

命令因为原意: make directories

命令所在路径: /bin/mkdir

执行权限: 所有用户

执行权限: 所有用户

功能描述: 创建新目录

-p 递归创建

例: mkdir -p/tmp/japan/hanmeimei

mkdir /tmp/japan/lilei /tmp/japan/hanmeimei

命令名称: **cd**

命令因为原意: change directory

命令所在路径: shell内置命令

执行权限: 所有用户

语法: cd [目录名]

功能描述: 切换目录

例: cd /tmp/japan/hanmeimei 切换到指定目录

cd .. 回到上一级目录

命令名称: **pwd**

命令因为原意: print working directory

命令所在路径: /bin/pwd

执行权限: 所有用户

语法: pwd

功能描述: 显示当前目录

范例: \$ pwd

命令名称: **rmdir**

命令因为原意: remove empty directories

命令所在路径: /bin/rmdir

执行权限: 所有用户

语法: rmdir [目录名]

功能描述: 删除空目录

范例: \$ rmdir /tmp/japan/lilei

命令名称: **cp**

命令因为原意: copy

命令所在路径: /bin/cp

执行权限: 所有用户

语法: cp -rp [原文件或目录][目标目录]

-r 复制目录

-p 保存文件属性

功能描述：复制文件或目录

范例: \$ cp -r /tmp/japan/hanmeimei /root

将目录/tmp/japan/hanmeimei 复制到目录/root下

cp -rp /tmp/japan/hameimei /tmp/japan/lilei /root

将/tmp/japan目录下的hanmeimei和lilei目录复制到root下，并保持目录属性

命令名称: **mv**

命令因为原意: move

命令所在路径: /bin/mv

执行权限: 所有用户

语法:mv [原文件或目录] [目标目录]

功能描述: 剪切文件、改名

命令名称: **rm**

命令因为原意: remove

命令所在路径: /bin/rm

执行权限: 所有用户

语法:rm -rf[文件或目录]

-r 删除目录

-f 强制执行

功能描述: 删除文件

范例: \$ rm /tmp/yum.log

删除文件/tmp/yum.log

rm -rf /tmp/japan/lilei

删除目录/tmp/japan/lilei

1.1.2. 文件处理命令

命令名称: **touch**

命令所在路径: /bin/touch

执行权限: 所有用户

语法:touch [文件名]

功能描述: 创建空文件

范例: \$ touch log.log

命令名称: **cat**

命令所在路径: /bin/cat

执行权限: 所有用户

语法:cat [文件名]

功能描述: 显示文件内容

-n 显示行号

```
范例: $ cat /etc/issue  
cat -n /etc/services
```

命令名称: **tac**

命令所在路径: /usr/bin/tac
执行权限: 所有用户
语法:tac [文件名]
功能描述: 显示文件内容(反向列示)

```
范例: $ tac /etc/issue
```

命令名称: **more**

命令所在路径: /bin/more
执行权限: 所有用户
语法:more [文件名]
(空格)或f 翻页
(Enter) 换行
q或Q 退出
功能描述: 分页显示文件内容

```
范例: $ more /etc/services
```

命令名称: **less**

命令所在路径: /usr/bin/less
执行权限: 所有用户
语法:less [文件名]
功能描述: 分页显示文件内容(可向上翻页)

```
范例: $ less /etc/services
```

命令名称: **head**

命令所在路径: /usr/bin/head
执行权限: 所有用户
语法:head [文件名]
功能描述: 显示文件前面几行
-n 指定行数

```
范例: $ head -n 20 /etc/services
```

命令名称: **tail**

命令所在路径: /usr/bin/tail
执行权限: 所有用户
语法:tail [文件名]

功能描述：显示文件后面面几行

- n 指定行数
- f 动态显示文件末尾内容

范例：\$ tail -n 18 /etc/services

命令名称： **ln**

命令英文原意：link

命令所在路径：/bin/link

执行权限：所有用户

语法：ls -s [原文件] [目标文件]

- s 创建软链接

功能描述：生成链接文件

范例：\$ ln -s /etc/issue /tmp/issue.soft

创建文件/etc/issue的软链接/tmp/issue.soft

ln /etc/issue /tmp/issue.hard

创建文件/etc/issue的硬链接/tmp/issue.hard

软链接特征：类似Windows快捷方式

1. 1rwxrwxrwx 1软链接

软链接文件权限都为rwxrwxrwx

2、文件大小-只是符号链接

3、/tmp/issue.soft->/etc/issue

箭头指向原文件

硬链接特征：

1.拷贝cp -p + 同步更新

echo "this is a test" >> /etc/motd

2.可通过i节点识别

3.不能跨分区

4.不能针对目录使用

1.1.3. 权限管理命令

命令名称： **chmod**

命令英文原意：change the permissions mode of a file

命令所在路径：/bin/chmod

执行权限：所有用户

语法：chmod [{ugoa}{+-=}{rwx}] [文件或目录]

[mode=421] [文件或目录]

-R 递归修改

功能描述：改变文件或目录权限

权限的数字表示

权限的数字表示

r 4

w 2

x 1

```
范例: $ chmod g+w testfile  
7 6 4  
范例: $ chmod -R 777 testdir  
赋予文件testfile所属组写权限  
修改目录testdir及其目录下文件为所有用户具有全部权限
```

1.1.4. 其他权限命令

命令名称: **chown**

```
命令英文原意: change file ownership  
命令所在路径: /bin/chown  
执行权限: 所有用户  
语法: chown [用户] [文件或目录]  
功能描述: 改变文件或目录的所有者  
范例: $ chown hanmeimei beijian  
改变文件beijian的所有者为hanmeimei
```

命令名称: **chgrp**

```
命令英文原意: change file group ownership  
命令所在路径: /bin/chgrp  
执行权限: 所有用户  
语法: chgrp [用户组] [文件或目录]  
功能描述: 改变文件或目录的所属组  
范例: $ chgrp hanmeimei beijing  
改变文件beijing的所属组为hanmeimei
```

命令名称: **umask**

```
命令英文原意: the user file-creation mask  
命令所在路径: Shell内置命令  
执行权限: 所有用户  
语法: umask [-S]  
-S 以rwx形式显示新建文件缺省权限  
功能描述: 显示、设置文件的缺省权限  
范例: $ umask -S
```

1.1.5. 文件搜索命令

命令名称: **find**

```
命令所在路径: /bin/find  
执行权限: 所有用户  
语法: find [搜索范围] [匹配条件]  
功能描述: 文件搜索
```

```
范例: $ find /etc -name init
在目录/etc中查找文件init
-iname 不区分大小写
find / -size +204800
在根目录下查找大于100MB的文件
+n 大于 -n 小于 n 等于
find /home -user hanmeimei
在根目录下查找所有者为hanmeimei的文件
-group 根据所属组查找
find /etc -cmin -5
在/etc下查找5分钟内被修改过属性的文件和目录
-amin 访问时间 access
-cmin 文件属性 change
-mmin 文件内容 modify
find /etc -size +163840 -a -size -204800
在/etc下查找大于80MB小于100MB的文件
-a 两个条件同时满足
-o 两个条件满足任意一个即可
find /etc -name inittab -exec ls -l {} \;
在/etc下查找inittab文件并显示其详细信息
-exec/-ok 命令 {} \; 对搜索结果执行操作
-type 根据文件类型查找
f 文件 d 目录 l 软链接文件
-inum 根据i节点查找
```

命令名称: **locate**

命令所在路径: /usr/bin/locate
执行权限: 所有用户
语法: locate 文件名
功能描述: 在文件资料库中查找文件
范例: \$ locate inittab

命令名称: **which**

命令所在路径: /usr/bin/which
执行权限: 所有用户
语法: which 命令
功能描述: 搜索命令所在目录及别名信息
范例: \$ which ls

命令名称: **whereis**

命令所在路径: /usr/bin/whereis
执行权限: 所有用户
语法: whereis [命令名称]
功能描述: 搜索命令所在目录及帮助文档路径
范例: \$ whereis ls

命令名称: **grep**

命令所在路径: /bin/grep
执行权限: 所有用户
语法: grep -iv [指定字串] [文件]
功能描述: 在文件中搜寻字串匹配的行并输出
 -i 不区分大小写
 -v 排除指定字串

范例: # grep mysql /root/install.log

1.1.6. 帮助命令

命令名称: **man**

命令英文原意: manual
命令所在路径: /usr/bin/man
执行权限: 所有用户
语法: man [命令或配置文件]
功能描述: 获得帮助信息

范例: \$ man ls
查看ls命令的帮助信息
\$ man services
查看配置文件services的帮助信息

命令名称: **help**

命令所在路径: Shell内置命令
执行权限: 所有用户
语法: help 命令
功能描述: 获得Shell内置命令的帮助信息
范例: \$ help umask
查看umask命令的帮助信息

1.1.7. 用户管理命令

命令名称: **useradd**

命令所在路径: /usr/sbin/useradd
执行权限: root
语法: useradd 用户名
功能描述: 添加新用户
范例: \$ useradd lilei

命令名称: **passwd username**

命令所在路径: /usr/bin/passwd
执行权限: 所有用户

语法: `passwd 用户名`
功能描述: 设置用户密码
范例: `$ passwd lilei`

命令名称: **who**

命令所在路径: `/usr/bin/who`
执行权限: 所有用户
语法: `who`
功能描述: 查看登录用户信息
范例: `$ who`

命令名称: **w**

命令所在路径: `/usr/bin/w`
执行权限: 所有用户
语法: `w`
功能描述: 查看登录用户详细信息
范例: `$ w`

1.1.8. 压缩解压命令

命令名称: **gzip** (只能压缩文件, 不能压缩目录)

命令英文原意: GNU zip

命令所在路径: `/bin/gzip`

执行权限: 所有用户

语法: `gzip [文件]`

功能描述: 压缩文件

压缩后文件格式: `.gz`

命令名称: **gunzip** (解压)

命令英文原意: GNU unzip

命令所在路径: `/bin/gunzip`

执行权限: 所有用户

语法: `gunzip [压缩文件]`

功能描述: 解压缩`.gz`的压缩文件

范例: `$ gunzip boduo.gz`

命令名称: **tar**

命令所在路径: `/bin/tar`
执行权限: 所有用户
语法: `tar 选项[-zcf] [压缩后文件名] [目录]`
 -c 打包
 -v 显示详细信息

-f 指定文件名
-z 打包同时压缩

功能描述：打包目录

压缩后文件格式：.tar.gz

范例：\$ tar -zcf Japan.tar.gz Japan

将目录Japan打包并压缩为.tar.gz文件

tar命令解压缩语法：

-x 解包
-v 显示详细信息
-f 指定解压文件
-z 解压缩

范例：\$ tar -zxvf Japan.tar.gz

命令名称： **zip**

命令所在路径：/usr/bin/zip

执行权限：所有用户

语法：zip 选项[-r] [压缩后文件名] [文件或目录]

-r 压缩目录

功能描述：压缩文件或目录

压缩后文件格式：.zip

范例：\$ zip buduo.zip buduo

压缩文件

\$ zip -r Japan.zip Japan

压缩目录

命令名称： **unzip**

命令所在路径：/usr/bin/unzip

执行权限：所有用户

语法：unzip [压缩文件]

功能描述：解压.zip的压缩文件

范例：\$ unzip test.zip

命令名称： **bzip2**

命令所在路径：/usr/bin/bzip2

执行权限：所有用户

语法： bzip2 选项 [-k] [文件]

-k 产生压缩文件后保留原文件

功能描述：压缩文件

压缩后文件格式：.bz2

范例：\$ bzip2 -k buduo

\$ tar -cjf Japan.tar.bz2 Japan

命令名称： **bunzip2**

命令所在路径: /usr/bin/bunzip2
执行权限: 所有用户
语法: bunzip2 选项 [-k] [压缩文件]
-k 解压缩后保留原文件
功能描述: 解压缩
范例: \$ bunzip2 -k boduo.bz2
\$ tar -xjf Japan.tar.bz2

1.1.9. 网络命令

指令名称: **write**

指令所在路径: /usr/bin/write
执行权限: 所有用户
语法: write <用户名>
功能描述: 给用户发信息, 以Ctrl+D保存结束
范例: # write linzhiling

命令名称: **ping**

命令所在路径: /bin/ping
执行权限: 所有用户
语法: ping 选项 IP地址
-c 指定发送次数
功能描述: 测试网络连通性

范例: # ping 192.168.1.156

命令名称: **ifconfig**

命令英文原意: interface configure
命令所在路径: /sbin/ifconfig
执行权限: root
语法: ifconfig 网卡名称 IP地址
功能描述: 查看和设置网卡信息
范例: # ifconfig eth0 192.168.8.250

命令名称: **mail**

命令所在路径: /bin/mail
执行权限: 所有用户
语法: mail [用户名]
功能描述: 查看发送电子邮件

范例: # mail root

命令名称: **last**

命令所在路径: /usr/bin/last
执行权限: 所有用户
语法: last
功能描述: 列出目前与过去登入系统的用户信息
范例: # last

命令名称: **lastlog**

命令所在路径: /usr/bin/lastlog
执行权限: 所有用户
语法: lastlog
功能描述: 检查某特定用户上次登录的时间
范例: # lastlog
lastlog -u 502

命令名称: **traceroute**

命令所在路径: /bin/traceroute
执行权限: 所有用户
语法: traceroute
功能描述: 显示数据包到主机间的路径
范例: # traceroute www.baidu.com

命令名称: **netstat**

命令所在路径: /bin/netstat
执行权限: 所有用户
语法: netstat [选项]
功能描述: 显示网络相关信息
选项:
-t : TCP协议
-u : UDP协议
-l : 监听
-r : 路由
-n : 显示IP地址和端口号

范例:
netstat -tlun 查看本机监听的端口
netstat -an 查看本机所有的网络连接
netstat -rn 查看本机路由表

命令名称: **setup**

命令所在路径: /usr/bin/setup
执行权限: root
语法: setup
功能描述: 配置网络
范例: # setup

命令名称: **mount**

```
命令位置: /bin/mount  
执行权限: 所有用户  
命令语法: mount [-t 文件系统] 设备文件名 挂载点  
  
范例: # mount -t iso9660 /dev/sr0 /mnt/cdrom
```

1.1.10. 关机重启命令

shutdown [选项] 时间

```
选项:  
-c: 取消前一个关机命令  
-h: 关机  
-r: 重启
```

其他关机命令

```
# halt  
# poweroff  
# init 0
```

其他重启命令

```
# reboot  
# init 6
```

系统运行级别

```
0  关机  
1  单用户  
2  不完全多用户, 不含NFS服务  
3  完全多用户  
4  未分配  
5  图形界面  
6  重启
```

修改系统默认运行级别

```
# cat /etc/inittab  
id:3:initdefault:
```

查询系统运行级别

```
# runlevel
```

退出登录命令

```
# logout
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:27

- 1. Linux查看硬件信息

1. Linux查看硬件信息

系统

```
# uname -a          # 查看内核/操作系统/CPU信息
# head -n 1 /etc/issue  # 查看操作系统版本
# cat /proc/cpuinfo  # 查看CPU信息
# hostname         # 查看计算机名
# lspci -tv        # 列出所有PCI设备 (yum install pciutils)
# lsusb -tv        # 列出所有USB设备
# lsmod            # 列出加载的内核模块
# env              # 查看环境变量
```

资源

```
# free -m          # 查看内存使用量和交换区使用量
# df -h           # 查看各分区使用情况
# du -sh <目录名>  # 查看指定目录的大小
# grep MemTotal /proc/meminfo  # 查看内存总量
# grep MemFree /proc/meminfo  # 查看空闲内存量
# uptime          # 查看系统运行时间、用户数、负载
# cat /proc/loadavg  # 查看系统负载
```

磁盘和分区

```
# mount | column -t    # 查看挂接的分区状态
# fdisk -l             # 查看所有分区
# swapon -s            # 查看所有交换分区
# hdparm -i /dev/hda   # 查看磁盘参数(仅适用于IDE设备)
# dmesg | grep IDE     # 查看启动时IDE设备检测状况
```

网络

```
# ifconfig          # 查看所有网络接口的属性
# iptables -L       # 查看防火墙设置
# route -n          # 查看路由表
# netstat -lntp     # 查看所有监听端口
# netstat -antp     # 查看所有已经建立的连接
# netstat -s          # 查看网络统计信息
```

进程

```
# ps -ef          # 查看所有进程
# top            # 实时显示进程状态
```

用户

```
# w                      # 查看活动用户  
# id <用户名>          # 查看指定用户信息  
# last                   # 查看用户登录日志  
# cut -d: -f1 /etc/passwd # 查看系统所有用户  
# cut -d: -f1 /etc/group  # 查看系统所有组  
# crontab -l             # 查看当前用户的计划任务
```

服务

```
# chkconfig --list        # 列出所有系统服务  
# chkconfig --list | grep on # 列出所有启动的系统服务
```

程序

```
# rpm -qa                # 查看所有安装的软件包
```

查看网卡型号

```
# lspci | grep Ethernet
```

查看内存和cpu最直接最常用的命令

```
# free  
# cat /proc/cpuinfo
```

查看硬盘信息 (查看硬盘型号和硬盘序列号)

```
# df -h
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:30

- 1. Linux下同步网络时间

1. Linux下同步网络时间

一、安装ntp

```
# yum install -y ntpdate
```

二、同步时间

```
# 方式一、使用域名连接，要经过DNS解析，速度慢。  
# ntpdate pool.ntp.org  
# 方式二、使用IP连接，超级快。  
# ntpdate 120.24.81.91
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:29

- 1. Linux或Win下安装node和npm

1. Linux或Win下安装node和npm

一、离线安装Node

官网下载链接：<https://nodejs.org/zh-cn/download/> node和npm对应关系：<https://nodejs.org/zh-cn/download/releases/>

在你的平台上下载 Node.js 源码或预编译安装包，然后即可马上进行开发。



解压目录

```
# cd /usr/local
# yum install -y xz
# xz -d node-v8.11.1-linux-x64.tar.xz
# tar -xf node-v8.11.1-linux-x64.tar
```

这里想要全局使用 npm 和 node ,就需要做相关软链，如下！

```
# cd /usr/local/bin
# ln -s /usr/local/src/node-v8.10.0-linux-x64/bin/npm
# ln -s /usr/local/src/node-v8.10.0-linux-x64/bin/node
```

然后即可在任意位置执行 node -v npm -v 查看相应的版本，则安装完成！

二、yum安装node和npm (Linux快速安装)

```
# yum install epel-release -y
# yum install nodejs -y
```

三、使用淘宝镜像站 (建议跳过此步骤，不建议使用)

```
# npm config get registry // https://registry.npmjs.org/
# npm config set registry https://registry.npm.taobao.org
---- 验证是否配置成功
# npm config get registry
```

四、指定全局安装路径和缓存路径 (看自己选择是否使用)

设置全局模块的安装路径到 `node_global` 文件夹，设置缓存到 `node_cache` 文件夹

```
# npm config set prefix D:\nodejs\node_global
# npm config set cache D:\nodejs\node_cache
---- 验证是否配置成功
# npm config get cache
# npm ls -g
```

五、安装 cnpm (看自己选择是否使用)

```
# npm install cnpm -g --registry=https://registry.npm.taobao.org
# cnpm -v
```

Tips:

1. **npm**默认的全局目录可执行命令查看

```
# npm config ls
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:29

- 1. Linux下安装Git

1. Linux下安装Git

1、使用yum安装

```
# yum -y install git  
# yum remove git
```

2、源代码安装

```
---安装依赖的包  
# yum -y install curl-devel expat-devel gettext-devel openssl-devel zlib-devel gcc  
perl-ExtUtils-MakeMaker zlib  
# cd /usr/local  
----下载git源码  
# wget https://github.com/git/git/archive/v2.21.0.tar.gz  
----解压git源码  
# tar -zxvf v2.21.0.tar.gz  
# cd git-2.21.0/  
# mkdir /usr/local/git  
----编译安装到指定目录  
# make prefix=/usr/local/git all  
# sudo make prefix=/usr/local/git install  
# vim /etc/profile  
----添加在文件的最后一行  
export PATH=/usr/local/git/bin:$PATH  
# source /etc/profile  
# git --version
```

3、FAQ

1、如果解压时出现报错，建议手动下载git源码包然后上传到服务器

```
gzip: stdin: unexpected end of file  
tar: Unexpected EOF in archive  
tar: Unexpected EOF in archive  
tar: Error is not recoverable: exiting now
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:28

- 1. Linux下Tomcat指定JDK和设置内存大小

1. Linux下Tomcat指定JDK和设置内存大小

一、Linux下Tomcat指定JDK

```
# vim bin/setclasspath.sh
```

在脚本开头的地方指定 JAVA_HOME 和 JRE_HOME

```
export JAVA_HOME=/usr/local/jdk1.8.0_40
export JRE_HOME=/usr/local/jdk1.8.0_40/jre
```

二、Linux下Tomcat设置内存大小

```
# vim bin/catalina.sh
```

Tomcat设置内存为8G: JAVA_OPTS="-server -Xms8192M -Xmx8192M -XX:PermSize=256M -XX:MaxPermSize=256M"

Tomcat设置内存为4G: JAVA_OPTS="-server -Xms4096M -Xmx4096M -XX:PermSize=256M -XX:MaxPermSize=256M"

```
[root@webserver tomcat8]# vim bin/catalina.sh

#
#           LOGGING_CONFIG="-Djava.util.logging.config.file=$CATALINA_BASE/conf/logging.properties"
#
#           LOGGING_MANAGER (Optional) Override Tomcat's logging manager
#           Example (all one line)
#           LOGGING_MANAGER="-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager"
#
#           UMASK      (Optional) Override Tomcat's default UMASK of 0027
#
#           USE_NOHUP    (Optional) If set to the string true the start command will
#           use nohup so that the Tomcat process will ignore any hangup
#           signals. Default is "false" unless running on HP-UX in which
#           case the default is "true"
#
# -----
#
# OS specific support. $var _must_ be set to either true or false.
JAVA_OPTS="-server -Xms8192M -Xmx8192M -XX:PermSize=256M -XX:MaxPermSize=256M"
cygwin=false
darwin=false
os400=false
hpx=false
case "`uname`" in
CYGWIN*) cygwin=true;;
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:29

- 1. nohup 和 >/dev/null 2>&1

1. nohup 和 >/dev/null 2>&1

一、用途： nohup 表示永久运行 & 表示后台运行

在应用Unix/Linux时，我们一般想让某个程序在后台运行

```
# nohup ./start-mysql.sh &
```

该命令的一般形式为：

```
# nohup command &
```

在缺省情况下该作业的所有输出都被重定向到一个名为 nohup.out 的文件中,除非另外指定了输出文件:

```
# nohup command > myout.file 2>&1 &
```

在上面的例子中，输出被重定向到myout.file文件中。

二、>/dev/null 2>&1

/dev/null 代表空设备文件，也就是不输出任何信息到终端，说白了就是不显示任何信息。

```
# nohup command 1>/dev/null 2>&1 &
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:29

- 1. Linux临时和永久关闭selinux

1. Linux临时和永久关闭selinux

临时关闭：

```
# setenforce 0
```

永久关闭：

```
# vim /etc/sysconfig/selinux
SELINUX=enforcing 改为 SELINUX=disabled
-- 重启生效
# reboot
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-02 13:44:29

- 1. vi-vim-粘贴内容时,保留原文件内容格式

1. vi-vim-粘贴内容时,保留原文件内容格式

在vi/vim普通模式下执行 :set paste 命令, 再按 i 插入就会保留原文件格式

- :set paste 打开粘贴命令
- :set nopaste 关闭命令
- :h paste 帮助

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:30

- 1. CentOS6-5-离线安装MySQL5-6-26

1. CentOS6-5-离线安装MySQL5-6-26

1、卸载之前安装的MySQL

```
# rpm -qa | grep mysql    --查看系统之前是否已安装MySQL。
# rpm -e --nodeps mysql-libs-5.1.71-1.el6.x86_64   --执行删除语句
# rm -rf /var/lib/mysql*   --善后处理
# rm -rf /usr/share/mysql*
```

2、安装MySQL

2.1、下载并安装 libaio

```
# cd /usr/local
# wget http://mirror.centos.org/centos/6/os/x86_64/Packages/libaio-0.3.107-10.el6.x86_64.rpm
# rpm -ivh libaio-0.3.107-10.el6.x86_64.rpm
```

2.2、下载并安装MySQL

```
# cd /usr/local
# wget https://dev.mysql.com/get/archives/mysql-5.6/MySQL-5.6.26-1.linux_glibc2.5.x86_64.rpm-bundle.tar
# tar -xf MySQL-5.6.26-1.linux_glibc2.5.x86_64.rpm-bundle.tar
# rpm -ivh MySQL-server-5.6.26-1.linux_glibc2.5.x86_64.rpm
# rpm -ivh MySQL-client-5.6.26-1.linux_glibc2.5.x86_64.rpm
```

2.3、查看并修改MySQL root用户密码

```
# more /root/.mysql_secret
# service mysql restart
# mysqladmin -uroot -pFeyU2YjwMcAw9Nv0 password 'root'  --修改root密码为root
# mysql -uroot -proot  --测试登陆
```

```
[root@normal share]# ^C
[root@normal share]# more /root/.mysql_secret
# The random password set for the root user at Mon May 28 18:33:35 2018 (local time): FeyU2YjwMcAw9Nv0
[root@normal share]#
```

2.4、配置MySQL数据库编码

```
# cp -rp /usr/share/mysql/my-default.cnf /etc/my.cnf
# 在mysqld下面增加
init-connect='SET NAMES utf8'
character-set-server = utf8
# service mysql restart
```

```
# mysql -uroot -proot --验证是否成功
# show variables like 'character%';
```

The screenshot shows a terminal window with two main sections. The top section displays MySQL configuration parameters:

```
# *** upgrade to a newer version of MySQL!
[mysqld]
init-connect='SET NAMES utf8'
character-set-server = utf8
# Remove leading # and set to the amount
# cache in MySQL. Start at 70% of total memory
# innodb buffer pool size = 128M
```

The bottom section shows the output of the command `show variables like 'character%';`, which lists various character set variables and their values. A red box highlights the first eight rows of this table.

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/

8 rows in set (0.00 sec)

2.5、配置MySQL远程访问

```
# service iptables stop
# chkconfig iptables off
# mysql -uroot -proot
# grant all privileges on *.* to root@'%' identified by 'root' WITH GRANT OPTION;
# flush privileges;
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:29

- 1. 更改MySQL的存储目录

1. 更改MySQL的存储目录

重要步骤：一定要先停止Mysql

```
# service mysqld stop
```

1、创建目标文件夹

```
# mkdir -p /data/mysql  
# chown -R mysql.mysql /data/mysql/
```

2、迁移命令

```
# mysql_install_db --user=mysql --basedir=/usr --datadir=/data/mysql
```

3、为避免麻烦，删掉原文件夹/var/lib/mysql

```
# rm -rf /var/lib/mysql/*
```

4、修改/etc/my.cnf配置文件

```
[mysqld]  
datadir=/data/mysql
```

5、启动mysql

```
# chkconfig mysqld on  
# service mysqld restart
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:29

- 1. Linux开启MySQL日志

1. Linux开启MySQL日志

```
# vim /etc/my.cnf
```

如果里面存在 !includedir /etc/my.cnf.d

```
#  
# This group is read both by the client and the server  
# use it for options that affect everything  
#[client-server]  
  
#  
# include all files from the config directory  
#  
!includedir /etc/my.cnf.d ←
```

则

```
# vim /etc/my.cnf.d/server.cnf  
没有就是  
# vim /etc/my.cnf  
#在mysqld标签下添加  
#指定错误日志保存位置  
log-error=/test_log/mysql_log/error.log  
#开启通用查询日志  
general_log=ON  
#指定通用查询日志保存位置  
general_log_file=/test_log/mysql_log/mysql.log  
#开启慢查询日志  
slow_query_log=on  
#记录超过1秒的SQL执行语句  
long_query_time=1  
#指定慢查询日志的保存位置  
slow-query-log-file=/test_log/mysql_log/slowquery.log
```

```
# this is only for the mysqld standalone daemon
[mysqld]

#指定错误日志保存位置
log-error=/test_log/mysql_log/error.log
#开启通用查询日志
general_log=ON
#指定通用查询日志保存位置
general_log_file=/test_log/mysql_log/mysql.log
#开启慢查询日志
slow_query_log=on
#记录超过1秒的SQL执行语句
long_query_time=1
#指定慢查询日志的保存位置
slow-query-log-file=/test_log/mysql_log/slowquery.log
```

重启MySQL

```
service mysqld restart
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:28

- 1. 解决MariaDB中文乱码

1. 解决MariaDB中文乱码

1、检查自己数据库编码

```
# mysql -uroot -proot
# show VARIABLES like 'char%';
```

```
MariaDB [(none)]> show VARIABLES like 'char%';
+-----+-----+
| Variable_name      | Value
+-----+-----+
| character_set_client | utf8
| character_set_connection | utf8
| character_set_database | latin1
| character_set_filesystem | binary
| character_set_results | utf8
| character_set_server | latin1
| character_set_system | utf8
| character_sets_dir   | /usr/share/mysql/charsets/
+-----+-----+
8 rows in set (0.00 sec)
```

2、更改Client和Server编码都是UTF-8

```
# vim /etc/my.cnf.d/server.cnf
```

在 `server.cnf` 中 `[mysqld]` 标签下添加代码

```
init-connect='SET NAMES utf8'
character-set-server = utf8
```

```
# this is only for the mysqld standalone daemon
[mysqld]
#大小写不敏感
lower_case_table_names=1
#解决中文乱码
init-connect='SET NAMES utf8'
character-set-server = utf8
#
# * Galera-related settings
#
[galera]
# Mandatory settings
#wsrep_on=ON
#wsrep_provider=
#wsrep_cluster_address=
```

重启MariaDB即可

```
# systemctl restart mariadb  
--- 如果已经添加为服务  
# service mysqld restart
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:28

- 1. CentOS-6-X-安装VirtualBox-5.1

1. CentOS-6-X-安装VirtualBox-5.1

一、安装virtualbox的yum源

```
# vim /etc/yum.repos.d/virtualbox.repo
# 添加以下内容
[virtualbox]
name=Oracle Linux / RHEL / CentOS-$releasever / $basearch - VirtualBox
baseurl=http://download.virtualbox.org/virtualbox/rpm/el/$releasever/$basearch
enabled=1
gpgcheck=1
gpgkey=http://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc
```

二、添加EPEL源

```
# rpm -Uvh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
```

三、添加必要的依赖包

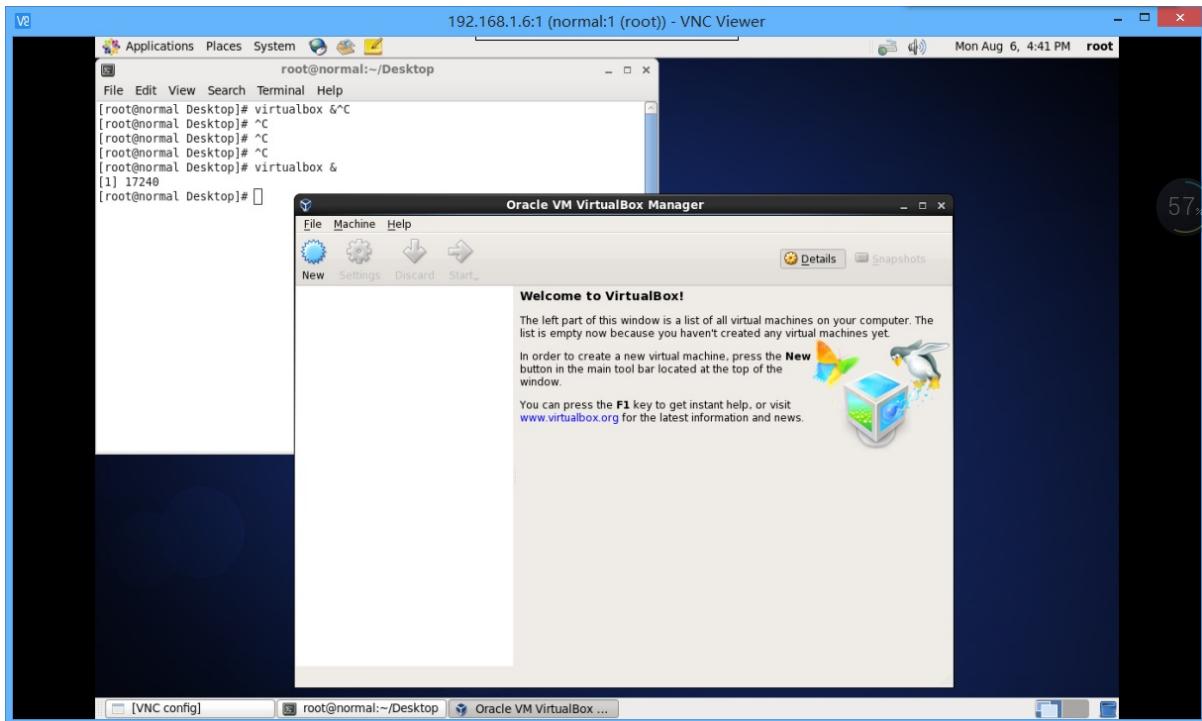
```
# yum install -y gcc make patch dkms qt libgomp
# yum install -y kernel-headers kernel-devel fontforge binutils glibc-headers glibc-devel
# yum install -y kernel kernel-headers kernel-devel gcc make
```

四、设置环境变量，主要是内核的路径

```
# vim /etc/profile
# 文件最后行添加
export KERN_DIR=/usr/src/kernels/你的最新内核版本xxx
# source /etc/profile
```

五、安装virtualbox

```
# yum search VirtualBox
# yum install -y VirtualBox-5.1
# reboot
# 重新编译与virtualbox相关的内核模块
# service vboxdrv setup
# 启动virtualbox
# virtualbox &
```



Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:30

- 1. CentOS-6.X-安装VNC-Server实现图形化访问

1. CentOS-6.X-安装VNC-Server实现图形化访问

一、安装gnome图形化桌面

```
# yum groupinstall -y "X Window System"
# yum groupinstall -y "Desktop"
# yum groupinstall -y "Chinese Support"
```

二、安装vncserver (vnc是一款优秀的远程控制软件)

```
# yum install -y tigervnc-server
```

三、配置vncserver

```
# ---配置为开机自启动
# chkconfig --level 345 vncserver on
# ---配置vnc密码
# vncserver
# ---配置为使用gnome桌面
# ---修改文件xstratup
# vim /root/.vnc/xstartup    把最后的 twm & 删掉 加上 gnome-session &
# ---配置vncserver启动后监听端口和环境参数
# vim /etc/sysconfig/vncservers
# ---在文件末添加以下内容
VNCSEVERS="1:root"
VNCSEVERARGS[1]="-geometry 1200x800"
# ---重启vncserver服务
# service vncserver restart
```

四、允许root访问图形界面和生成新的machine-id

```
# sed -i 's/.*= root.*#/&/' /etc/pam.d/gdm
# dbus-uuidgen > /var/lib/dbus/machine-id
```

五、关闭selinux和NetworkManager服务

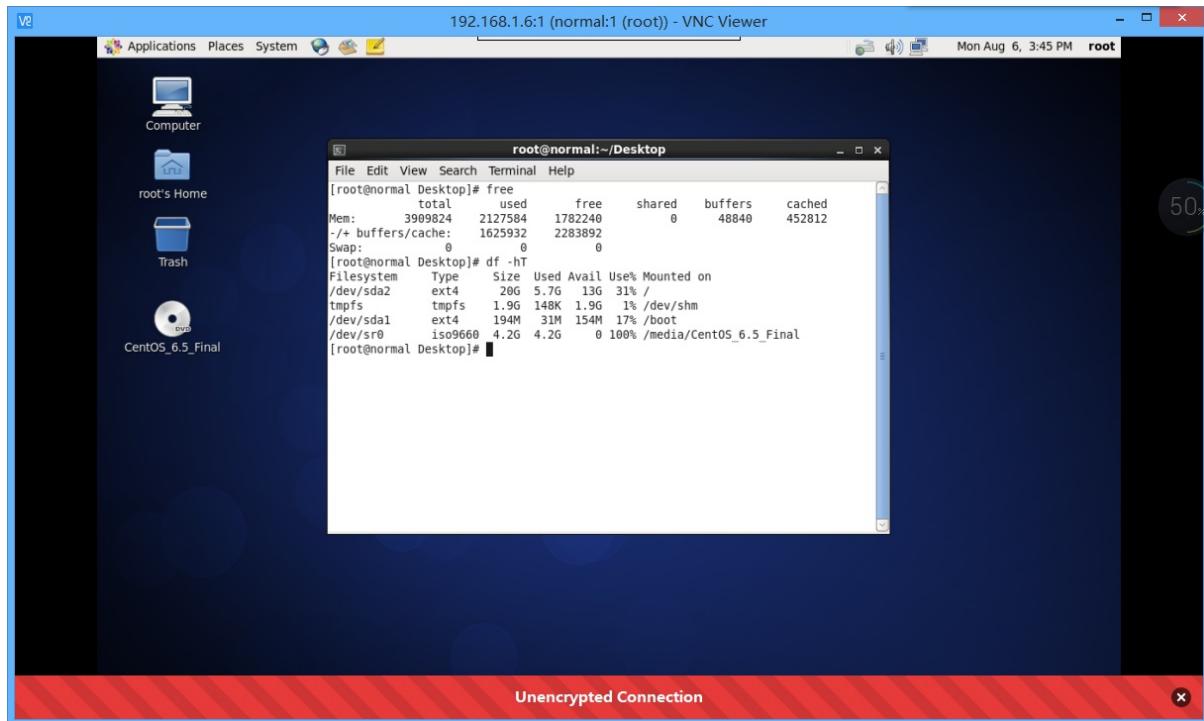
1.检查selinux服务并关闭(确认里面的SELINUX字段的值是disabled, 如果不是则改为disabled)

```
# vim /etc/selinux/config
```

2.关闭NetworkManager服务

```
# chkconfig --del NetworkManager
```

安装vncviewer客户端输入IP:1 输入密码访问即可



Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:30

- 1. CentOS-6.X-图形界面安装

1. CentOS-6.X-图形界面安装

一、安装图形界面

```
# yum groupinstall -y "Desktop" "Desktop Platform" "Fonts" "General Purpose Desktop"
" "Graphical Administration Tools" "Input Methods" "Legacy X Window System compatibility"
" "Remote Desktop Clients" "X Window System"
# vi /etc/inittab
# 修改最后一行为id:5:initdefault:
# startx
# reboot
```

二、安装中文输入法(可选)

```
# yum -y install "@Chinese Support"
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:30

- 1. Linux中Apache(httpd)安装、配置
 - 1.1.1. 1、yum安装
 - 1.1.2. 2、配置、启动Apache
 - 1.1.3. 3、Apache设置为Linux系统服务，并开机启动

1. Linux中Apache(httpd)安装、配置

1.1.1. 1、yum安装

```
--- 先yum list看看有没有你可以的包
# yum list httpd
--- 如果有的话，比如我用x86_64版，就可以安装
# yum install httpd.x86_64
```

1.1.2. 2、配置、启动Apache

2.1、`httpd.conf` 是主配置文件，你可以按需更改，一般来说可能需要自定义端口，那么就 `vim` 修改 `httpd.conf` 文件的监听端口，改为你希望的

```
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 8010
```

查找httpd.conf文件位置

```
# find / -name httpd.conf
```

2.2、启动、关闭、重启

```
# ./apachectl start
# ./apachectl stop
# ./apachectl restart
```

一般此文件在 /usr/sbin 目录下

1.1.3. 3、Apache设置为Linux系统服务，并开机启动

3.1、将链接 `apachectl` 复制到系统启动目录下并命名为 `httpd`

```
# cp /usr/local/httpd/bin/apachectl /etc/rc.d/init.d/httpd
```

至此，其实已经可以使用 `service httpd start` 等命令，但在linux服务列表还没有注册

3.2、使用 `chkconfig` 来注册 `apache` 服务，并其可以在linux的服务列表中看到（`chkconfig --list`）

在 `init.d` 的 `httpd` 链接文件中第2行添加如下注释，并 `wq` 保存退出

```
# vim /etc/rc.d/init.d/httpd
```

```
#!/bin/sh
# chkconfig:35 61 61
# description:Apache
```

将 `httpd` 配置自启并加入linux服务监控

```
# chkconfig --add httpd
```

全局可以使用命令：

```
# service httpd start
# service httpd stop
# service httpd restart
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-02 13:44:26

- 1. Linux-Apache服务不能访问文件解决方案

1. Linux-Apache服务不能访问文件解决方案

httpd服务url地址不能访问文件 请先关闭selinux再reboot重启

关闭selinux

```
# vim /etc/selinux/config  
SELINUX更改为  
SELINUX=disabled  
# reboot
```

重启后访问http地址ok

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:28

- 1. Linux安装OpenResty-api-网关-Orange

1. Linux安装OpenResty-api-网关-Orange

1. 安装openresty

```
# yum -y install libuuid-devel pcre-devel openssl-devel gcc-c++ wget
# mkdir /openresty
# cd /openresty
# wget https://openresty.org/download/openresty-1.9.15.1.tar.gz
# tar -zxf openresty-1.9.15.1.tar.gz
# cd openresty-1.9.15.1
# ./configure --with-http_stub_status_module --with-http_v2_module --with-http_ssl_module
# gmake && gmake install
# ln -s /usr/local/openresty/nginx/sbin/nginx /usr/sbin/nginx
# nginx -v
```

2. 创建MySQL数据库并导入数据(脚本在orange/install文件夹下)

```
# yum -y install mariadb-server
# mysql -u root
# CREATE DATABASE orange CHARACTER SET utf8 COLLATE utf8_general_ci;
# CREATE USER 'orange'@'%' IDENTIFIED BY 'orange';
# GRANT ALL PRIVILEGES ON orange.* TO 'orange'@'%';
# FLUSH PRIVILEGES;
# 最后一定要执行mysql的数据库导入。
# mysql -u orange -porange -h 10.0.2.15 orange < orange-v0.6.2.sql
```

3. 安装Orange

Tips: 安装之前需要 lor 框架, 否则启动有问题。

```
# yum install -y git
# git clone https://github.com/sumory/lor.git
# cd lor
# make install
```

启动并配置 orange 服务

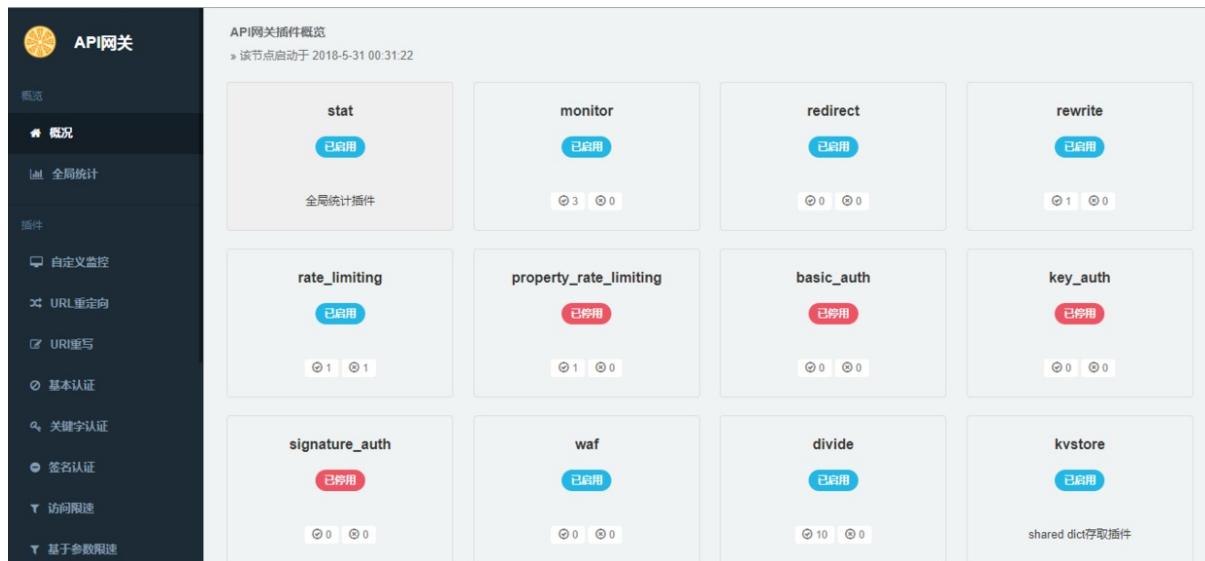
```
# service iptables stop
# chkconfig iptables off
# git clone https://github.com/sumory/orange.git
# cd orange
# vim conf/orange.conf
# sh start.sh
```

```

        "divide",
        "kvstore"
    ],
    "store": "mysql",
    "store_mysql": {
        "timeout": 5000,
        "connect_config": {
            "host": "192.168.1.201",
            "port": 3306,
            "database": "orangel",
            "user": "root",
            "password": "root",
            "max_packet_size": 1048576
        },
        "pool_config": {
            "max_idle_timeout": 10000,
            "pool_size": 3
        },
        "desc": "mysql configuration"
    }
}

```

访问地址: <http://IP:9999>



Tips:启动不起来查看端口占用情况杀掉其他进程

```
# netstat -tunlp |grep 80
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:29

- 1. Linux搭建GitLab并汉化
 - 1.1.1. 一、环境准备
 - 1.1.2. 二、获取GitLab汉化包（要部署非汉化版，可以跳过这一块内容）
 - 1.1.3. 三、部署社区版GitLab
 - 1.1.4. 四、覆盖汉化包

1. Linux搭建GitLab并汉化

1.1.1. 一、环境准备

1.1、下载wget

```
# yum install -y vim  
# yum install -y wget  
# cd /usr/local  
# wget https://mirrors.tuna.tsinghua.edu.cn/gitlab-ce/yum/el6/gitlab-ce-10.0.2-ce.0  
.el6.x86_64.rpm
```

1.2、关闭防火墙

```
# service iptables stop  
# chkconfig iptables off
```

1.1.2. 二、获取GitLab汉化包（要部署非汉化版，可以跳过这一块内容）

2.1、安装Git

```
# yum install -y git
```

2.2、克隆获取汉化版本库

```
# cd /usr/local  
# git clone https://gitlab.com/xhang/gitlab.git
```

1.1.3. 三、部署社区版GitLab

3.1、安装GitLab的依赖项

```
# yum -y install policycoreutils openssh-server openssh-clients postfix cronie poli  
cycoreutils-python
```

3.2、启动postfix，并设置为开机启动

```
# service postfix start
# chkconfig postfix on
```

3.3、安装rpm包

```
# cd /usr/local
# rpm -ivh gitlab-ce-10.0.2-ce.0.el6.x86_64.rpm
```

3.4、修改配置文件gitlab.rb

```
# vim /etc/gitlab/gitlab.rb
将external_url变量的地址修改为gitlab所在centos的ip地址
修改GitLab默认端口

unicorn['port'] = '9092'

##! https://gitlab.com/gitlab-org/omnibus-gitlab/raw/master/files/gitlab-confi

## GitLab URL
##! URL on which GitLab will be reachable.
##! For more details on configuring external_url see:
##! https://docs.gitlab.com/omnibus/settings/configuration.html#configuring-th
external_url 'http://192.168.1.8/'
unicorn['port'] = '9092'
## Legend
##! The following notations at the beginning of each line may be used to
##! differentiate between components of this file and to easily select them us
##! a regex.
##! ## Titles, subtitles etc
##! ##! More information - Description, Docs, Links, Issues etc.
##! Configuration settings have a single # followed by a single space at the
##! beginning; Remove them to enable the setting.

##! **Configuration settings below are optional.**
##! **The values currently assigned are only examples and ARE NOT the default
```

因为修改了配置文件，故需要重新加载配置内容。

```
# gitlab-ctl reconfigure
# gitlab-ctl restart
```

1.1.4. 四、覆盖汉化包

4.1、停止GitLab服务

```
# gitlab-ctl stop
```

4.2、比较汉化标签和原标签，导出 patch 用的 diff 文件到/root下

```
# cd /usr/local/gitlab
```

```
# git diff v10.0.2 v10.0.2-zh > ../10.0.2-zh.diff
```

4.3、将10.0.2-zh.diff作为补丁更新到GitLab中

```
# cd /usr/local  
# yum install patch -y  
# patch -d /opt/gitlab/embedded/service/gitlab-rails -p1 < 10.0.2-zh.diff  
Tips:重启GitLab  
# gitlab-ctl reconfigure  
# gitlab-ctl restart
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:28

- 1. Linux配置SSH-Key到GitHub

1. Linux配置SSH-Key到GitHub

1、生成SSH Key

首先检查下本机是否已经安装了SSH，在终端输入ssh即可：

```
[root@repo hexo]# ssh
usage: ssh [-1246AaCfgKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-e escape_char] [-F configfile]
           [-I pkcs11] [-i identity_file]
           [-L [bind_address:]port:host:hostport]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-R [bind_address:]port:host:hostport] [-S ctl_path]
           [-W host:port] [-w local_tun[:remote_tun]]
           [user@]hostname [command]
```

如果没有安装进行yum安装

```
# yum -y install openssh-clients
```

接下来就是生成 ssh key 了，输入 ssh-keygen -t rsa，然后连续按回车键三次（注意：千万不要输入密码！）。

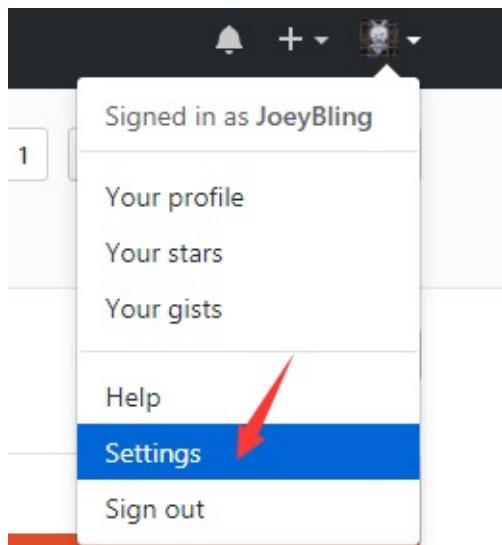
```
[root@repo ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
19:76:07:0c:56:4c:0d:56:5a:00:c1:71:42:cc:65:28 root@repo
The key's randomart image is:
+--[ RSA 2048]--+
| =0%@=0 |
| E.==0+. |
| .0 0 . |
| . + . |
| S |
| |
| |
| |
+-----+
```

`id_rsa.pub` 为公钥，`id_rsa` 为私钥，它们都是隐藏文件。那么如何查看它们的内容呢？只需要继续执行以下两条命令即可。

```
# cd .ssh/
# cat id_rsa.pub
```

```
[root@repo .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQEA074RKekj+lsIILxfynF43Bz70cpw/yuD9dc2PUiiX/vq8oRtJJJeMNd+hqaSlb8DBWcpcmhazMOF30XTHPzKCxpKMXJYIeWfUln/S4uf0mFwJNG+jbLxwAQjH_lj0Xn4
xu8gjtkWCGr2zM48Q9Tjg5b3bCsggtpEQAA4CsHUdYjmBsHpiQM04IaAQ0AjTfmfwgy/TNfNPdiXmNBBFZcHIPciLRVYdmBPVqn32tP9HmbenGnQs/QS45LxeCBejDaLtM9g/Xodtr29xFw1P5gdZpUmplkwkltN
Uc6eUKDD3EXXZX0imG9fGXKL/U9g/fLLKZizz56ES4yrTIQJ84e+Q== root@repo
```

2、添加SSH Key到GitHub上



接着执行下面操作：

The screenshot shows the GitHub settings interface. On the left, there's a sidebar with links: Personal settings, Profile, Account, Emails, Notifications, Billing, **SSH and GPG keys** (which has a red arrow pointing to it), Security, Blocked users, Repositories, and Organizations. The main area is titled "SSH keys". It contains a list of associated SSH keys. One key is shown: "Linux_Test" (Fingerprint: [redacted], Added on 17 Apr 2018, Last used within the last week — Read/write). There is a "Delete" button next to it. A red arrow points to the "New SSH key" button at the top right of the list. Below the SSH keys, there's a "GPG keys" section which says "There are no GPG keys associated with your account." and a "New GPG key" button.

点击New SSH Key按钮后进行Key的填写操作，完成SSH Key的添加。如下图：

Title
| 不用填写

Key
Begin with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key 点击完成添加操作

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

<http://blog.csdn.net/gulingfengze>

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Key2	Fingerprint: 13:ef:7c:... SSH	Added on 17 May 2016 Last used within the last 2 months	Delete
farsight@ubuntu	Fingerprint: f5:21:61:00:... SSH	Added on 8 Apr 2017 Never used	Delete

<http://blog.csdn.net/gulingfengze>

添加SSH Key成功之后，继续输入命令进行测试。

```
# ssh -T git@github.com
```

```
[root@repo .ssh]# ssh -T git@github.com
The authenticity of host 'github.com (52.74.223.119)' can't be established.
RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,52.74.223.119' (RSA) to the list of known hosts.
Hi JoeyBling! You've successfully authenticated, but GitHub does not provide shell access.
```

出现上图结果则说明添加SSH Key成功。

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-02 13:44:29

- 1. Linux下fsck-ext4-Unable-to-resolve问题记录

1. Linux下fsck-ext4-Unable-to-resolve问题记录

linux启动报错:

```
Welcome to CentOS
Starting udev: [ OK ]
Setting hostname localhost.localdomain: [ OK ]
Checking filesystems
/dev/sda3: clean, 21146/630784 files, 270602/2518784 blocks
fsck.ext4: Unable to resolve 'UUID=e29667b7-202a-470c-b578-3fab9fbdc08b'
/dev/sda1: clean, 38/51200 files, 33261/204800 blocks [ FAILED ]

**** An error occurred during the file system check.
**** Dropping you to a shell; the system will reboot
**** when you leave the shell.
**** Warning -- SELinux is active
**** Disabling security enforcement for system recovery.
**** Run 'setenforce 1' to reenable.
Give root password for maintenance
(or type Control-D to continue): _
```

由于其他原因将分区UUID为e29667b7-.... 删除了，重启系统发现fsck 文件系统时failed

问题解决:

在(or type control -D to continue):后面输入root密码后回车

```
# vi /etc/fstab
```

在编辑该文件的时候，系统提示该文件为只读，无法编辑，原因是在进入单用户模式后，由于根分区是只读的，需要通过重新挂载

```

#
# /etc/fstab
# Created by anaconda on Wed Mar 14 13:16:30 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=3d5ade3d-45cb-402a-bc39-5d06222f3711 / ext4 defaul
ts 1 1
UUID=7455e2b7-ecb0-459a-8901-48e89f321e0f /boot ext4 defaul
ts 1 2
UUID=e29667b7-202a-470c-b578-3fab9fbdc08b /home ext4 defaul
ts 0 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
/dev/sda5 /data ext4 defaults 0 0
/dev/sda6 /data2 ext4 defaults 0 0
-
-
-
-
E303: Unable to open swap file for "/etc/fstab", recovery impossible
Press ENTER or type command to continue_

```

```

# mount -o remount,rw /
# vi /etc/fstab

```

把无法挂载的分区那一行行首用 # 注释掉或将那一行直接删除掉，重新启动就OK了。

```

#
# /etc/fstab
# Created by anaconda on Wed Mar 14 13:16:30 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=3d5ade3d-45cb-402a-bc39-5d06222f3711 / ext4 defaul
ts 1 1
UUID=7455e2b7-ecb0-459a-8901-48e89f321e0f /boot ext4 defaul
ts 1 2
#UUID=e29667b7-202a-470c-b578-3fab9fbdc08b /home ext4 defaul
lts 0 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
/dev/sda5 /data ext4 defaults 0 0
/dev/sda6 /data2 ext4 defaults 0 0
-
-
-
-
-- INSERT --

```

```

# reboot

```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:28

- 1. Linux-卸载分区

1. Linux-卸载分区

注意，卸载分区会格式化分区内所有的数据，请谨慎操作或进行数据备份

```
# df -hT
```

```
[root@localhost ~]# df -hT
Filesystem      Type  Size  Used  Avail Use% Mounted on
/dev/sda3        ext4  9.5G  905M  8.1G  10% /
tmpfs           tmpfs 491M   0  491M   0% /dev/shm
/dev/sda1        ext4  194M  27M  158M  15% /boot
/dev/sda2        ext4  194M  5.6M  179M  4% /home
/dev/sda5        ext4  9.9G  151M  9.2G  2% /data
/dev/sda6        ext4  9.9G  151M  9.2G  2% /data2
[root@localhost ~]#
```

准备卸载/dev/sda6这个分区

```
# umount /data2
```

重新检查一下

```
# df -hT
```

```
[root@localhost ~]# df -hT
Filesystem      Type  Size  Used  Avail Use% Mounted on
/dev/sda3        ext4  9.5G  905M  8.1G  10% /
tmpfs           tmpfs 491M   0  491M   0% /dev/shm
/dev/sda1        ext4  194M  27M  158M  15% /boot
/dev/sda2        ext4  194M  5.6M  179M  4% /home
/dev/sda5        ext4  9.9G  151M  9.2G  2% /data
```

```
# fdisk -l
```

发现待分区的磁盘 /dev/sda

```
[root@localhost ~]# fdisk -l
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End    Blocks  Id  System
/dev/sda1  *          1         26    204800  83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2          26         52    204800  83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3          52        1306   10075136  83  Linux
/dev/sda4          1306       3916   20969510   5  Extended
/dev/sda5          1306       2611   10487066  83  Linux
/dev/sda6          2612       3916   10482381  83  Linux
```

对该磁盘进行卸载分区，输入m并回车

```
# fdisk /dev/sda
# m 输入帮助
# p 打印分区表
```

```
[root@localhost ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): m
Command action
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)
```

```
Command (m for help): p

Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End      Blocks   Id  System
/dev/sda1  *           1         26     204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            26         52     204800   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3            52        1306    10075136   83  Linux
/dev/sda4          1306        3916    20969510    5  Extended
/dev/sda5          1306        2611    10487066   83  Linux
/dev/sda6          2612        3916    10482381   83  Linux
```

卸载/dev/sda6分区 此时注意/dev/sda6是刚刚卸载了/data2的分区名

```
[root@localhost ~]# df -hT
Filesystem  Type  Size  Used  Avail Use% Mounted on
/dev/sda3    ext4  9.5G  905M  8.1G  10% /
tmpfs       tmpfs  491M   0  491M  0% /dev/shm
/dev/sda1    ext4  194M  27M  158M  15% /boot
/dev/sda2    ext4  194M  5.6M  179M  4% /home
/dev/sda5    ext4  9.9G  151M  9.2G  2% /data
/dev/sda6    ext4  9.9G  151M  9.2G  2% /data2
```

```
# d 删除一个分区
```

6 删除第六个分区

```
Command (m for help): p
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End      Blocks   Id  System
/dev/sdal    *          1         26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            26         52      204800   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3            52        1306     10075136   83  Linux
/dev/sda4        1306        3916     20969510   5   Extended
/dev/sda5        1306        2611     10487066   83  Linux
/dev/sda6    <--           2612        3916     10482381   83  Linux

Command (m for help): d
Partition number (1-6): 6 <--
```

Command (m for help): p

重新打印分区表发现已卸载/dev/sda6分区

p

```
Command (m for help): p
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End      Blocks   Id  System
/dev/sdal    *          1         26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            26         52      204800   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3            52        1306     10075136   83  Linux
/dev/sda4        1306        3916     20969510   5   Extended
/dev/sda5        1306        2611     10487066   83  Linux
```

w 将表写入磁盘并退出

再检查一遍

fdisk /dev/sda

p 打印分区表

```

Command (m for help): p

Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End      Blocks   Id  System
/dev/sda1  *           1         26     204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            26         52     204800   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3            52        1306    10075136   83  Linux
/dev/sda4          1306        3916    20969510     5  Extended
/dev/sda5          1306       2611    10487066   83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

```

```

Syncing disks.
[root@localhost ~]# fdisk /dev/sda
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): p
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End      Blocks   Id  System
/dev/sda1  *           1         26     204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            26         52     204800   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3            52        1306    10075136   83  Linux
/dev/sda4          1306        3916    20969510     5  Extended
/dev/sda5          1306       2611    10487066   83  Linux

```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:29

- 1. Linux-新建扩展分区

- 1.1.1. Tips:

1. Linux-新建扩展分区

```
# fdisk -l
```

```
[root@localhost ~]# fdisk -l
Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

Device Boot Start End Blocks Id System
/dev/sdal * 1 26 204800 83 Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2 26 52 204800 83 Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3 52 1306 10075136 83 Linux
```

要进行新建扩展分区的磁盘是 /dev/sda

```
# fdisk /dev/sda
# p 打印分区表
```

```
[root@localhost ~]# fdisk /dev/sda
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

Device Boot Start End Blocks Id System
/dev/sdal * 1 26 204800 83 Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2 26 52 204800 83 Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3 52 1306 10075136 83 Linux

Command (m for help):
```

此时磁盘已经有了3个分区，如果要再弄2个分区就不行，这个时候我们进行新建扩展分区

```
# n 添加一个新的分区
```

```
# e 添加一个扩展分区
# p 打印分区表
```

Command (m for help): n ←
 Command action
 e extended
 p primary partition (1-4)
 e ←
 Selected partition 4
 First cylinder (1306-3916, default 1306):
 Using default value 1306
 Last cylinder, +cylinders or +size{K,M,G} (1306-3916, default 3916):
 Using default value 3916
 Command (m for help): p ←
 Disk /dev/sda: 32.2 GB, 32212254720 bytes
 255 heads, 63 sectors/track, 3916 cylinders
 Units = cylinders of 16065 * 512 = 8225280 bytes
 Sector size (logical/physical): 512 bytes / 512 bytes
 I/O size (minimum/optimal): 512 bytes / 512 bytes
 Disk identifier: 0x000f15b2
 Device Boot Start End Blocks Id System
 /dev/sda1 * 1 26 204800 83 Linux
 Partition 1 does not end on cylinder boundary.
 /dev/sda2 26 52 204800 83 Linux
 Partition 2 does not end on cylinder boundary.
 /dev/sda3 52 1306 10075136 83 Linux
 /dev/sda4 1306 3916 20969510 5 Extended →

此时已经看见扩展分区创建完成，在扩展分区下面新建2个分区 因为现在扩展分区是20G 所以下面的2个分区都给10G

```
# n
# +10G
# p
```

Command (m for help): n ←
 First cylinder (1306-3916, default 1306): → 右车 ←
 Using default value 1306
 Last cylinder, +cylinders or +size{K,M,G} (1306-3916, default 3916): +10G

```
Command (m for help): p

Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End      Blocks   Id  System
/dev/sdal    *          1         26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            26         52      204800   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3            52        1306     10075136   83  Linux
/dev/sda4        1306        3916     20969510     5  Extended
/dev/sda5        1306        2611     10487066   83  Linux
```

重复上面的操作

```
# n
# 回车
# 回车    此时回车等于把磁盘剩下的所有空间给了此分区
# p
```

```
Command (m for help): n
First cylinder (2612-3916, default 2612):
Using default value 2612
Last cylinder, +cylinders or +size{K,M,G} (2612-3916, default 3916):
Using default value 3916
```

```
Command (m for help): p

Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End      Blocks   Id  System
/dev/sdal    *          1         26      204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            26         52      204800   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3            52        1306     10075136   83  Linux
/dev/sda4        1306        3916     20969510     5  Extended
/dev/sda5    ←           1306        2611     10487066   83  Linux
/dev/sda6    ←           2612        3916     10482381   83  Linux
```

创建完毕 保存并重启

```
# w
# reboot
```

```
Command (m for help): w ←
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: Device or resource busy
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
[root@localhost ~]# reboot ←
```

重新登入系统

```
# fdisk -l
```

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f15b2

      Device Boot      Start        End    Blocks   Id  System
/dev/sda1  *          1         26     204800   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2          26         52     204800   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sda3          52        1306    10075136   83  Linux
/dev/sda4        1306        3916    20969510     5  Extended
/dev/sda5 ←        1306        2611    10487066   83  Linux
/dev/sda6 ←        2612        3916    10482381   83  Linux
```

需要格式化 /dev/sda5 /dev/sda6 分区

```
--- ext4是磁盘文件系统的格式，也可以是ext3
# mkfs -t ext4 /dev/sda5
# mkfs -t ext4 /dev/sda6
```

```
[root@localhost ~]# mkfs -t ext4 /dev/sda5
mke2fs 1.41.12 (17-May-2010)
warning: 326 blocks unused.

Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
656640 inodes, 2621440 blocks
131088 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
8208 inodes per group
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 37 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

```
[root@localhost ~]# mkfs -t ext4 /dev/sda6
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2620595 blocks
131029 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 36 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

```
# mount /dev/sda5 /data
报错:mount: mount point /data does not exist
原因是根目录没有此文件夹
创建文件夹
# mkdir /data
# mkdir /data2
# mount /dev/sda5 /data    把分区 /dev/sda5 挂载到 /data 目录下
# mount /dev/sda6 /data2  把分区 /dev/sda6 挂载到 /data2 目录下
# df -hT  查看是否挂载成功
```

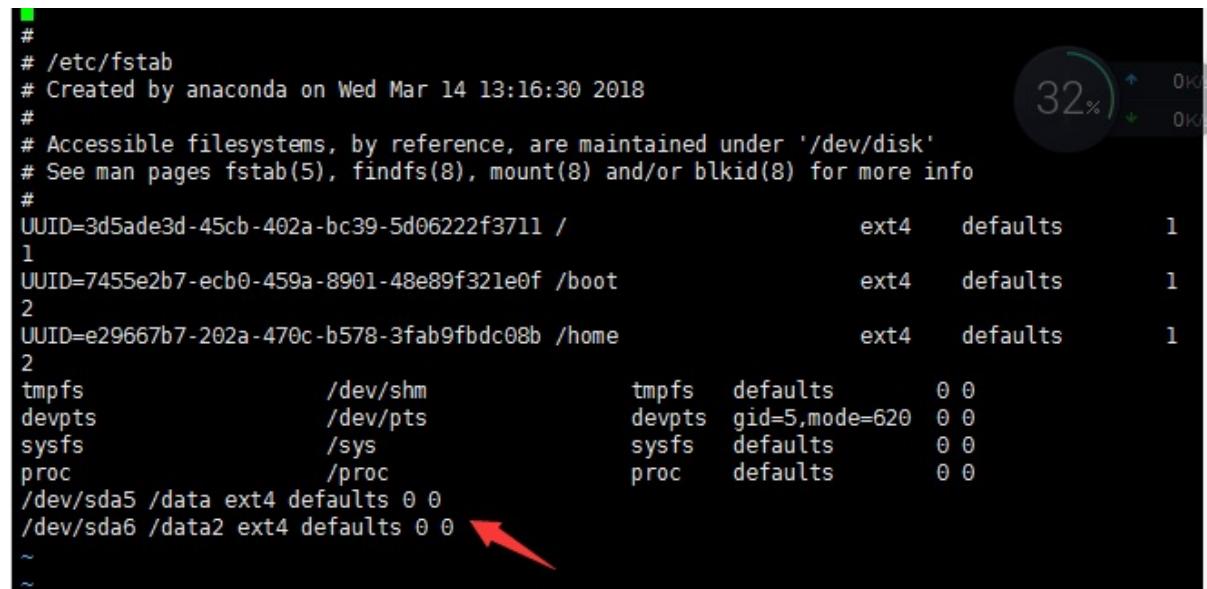
```
[root@localhost ~]# df -hT
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/sda3        ext4  9.5G  902M  8.1G  10% /
tmpfs           tmpfs 491M   0  491M  0% /dev/shm
/dev/sda1        ext4  194M  27M  158M  15% /boot
/dev/sda2        ext4  194M  5.6M  179M  4% /home
/dev/sda5        ext4  9.9G  151M  9.2G  2% /data
/dev/sda6        ext4  9.9G  151M  9.2G  2% /data2
```

分区格式类型是ext4 分区大小也是10G

此时注意，这样机器重启后就重置了，如果需要永久生效需要设置开机自动挂载（磁盘）

开机自动挂载（磁盘）

```
# vi /etc/fstab
--- 最后面添加
/dev/sda5 /data ext4 defaults 0 0
/dev/sda6 /data2 ext4 defaults 0 0
```



```
# /etc/fstab
# Created by anaconda on Wed Mar 14 13:16:30 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=3d5ade3d-45cb-402a-bc39-5d06222f3711 /          ext4    defaults      1
1
UUID=7455e2b7-ecb0-459a-8901-48e89f321e0f /boot       ext4    defaults      1
2
UUID=e29667b7-202a-470c-b578-3fab9fbdc08b /home       ext4    defaults      1
2
tmpfs          /dev/shm      tmpfs   defaults      0 0
devpts         /dev/pts      devpts  gid=5,mode=620  0 0
sysfs          /sys          sysfs   defaults      0 0
proc            /proc          proc    defaults      0 0
/dev/sda5 /data ext4 defaults 0 0
/dev/sda6 /data2 ext4 defaults 0 0
```

1.1.1. Tips:

第一列：分区的UUID或分区名 例如:/dev/sda5

第二列：Mount point：设备的挂载点，就是你要挂载到哪个目录下。

第三列：filesystem：磁盘文件系统的格式，包括ext2、ext3、reiserfs、nfs、vfat等

第四列：parameters：文件系统的参数

- Async/sync 设置是否为同步方式运行，默认为async
- auto/noauto 当下载mount -a 的命令时，此文件系统是否被主动挂载。默认为auto
- rw/ro 是否以只读或者读写模式挂载
- exec/noexec 限制此文件系统内是否能够进行"执行"的操作

- `user/nouser` 是否允许用户使用mount命令挂载
- `suid/nosuid` 是否允许SUID的存在
- `Usrquota` 启动文件系统支持磁盘配额模式
- `Grpquota` 启动文件系统对群组磁盘配额模式的支持
- `Defaults` 同事具有`rw,suid,dev,exec,auto,nouser,async`等默认参数的设置

第五列：能否被dump备份命作用：dump是一个用来作为备份的命令。通常这个参数的值为0或者1

- 0 代表不要做dump备份
- 1 代表要每天进行dump的操作
- 2 代表不定日期的进行dump操作

第六列：是否检验扇区：开机的过程中，系统默认会以fsck检验我们系统是否为完整（clean）。

- 0 不要检验
- 1 最早检验（一般根目录会选择）
- 2 在级别检验完成之后进行检验

Tips:

查看各个分区的UUID和磁盘文件系统的格式

```
# blkid
```

```
[root@localhost ~]# blkid
/dev/sda1: UUID="7455e2b7-ecb0-459a-8901-48e89f321e0f" TYPE="ext4"
/dev/sda2: UUID="e29667b7-202a-470c-b578-3fab9fbdc08b" TYPE="ext4"
/dev/sda3: UUID="3d5ade3d-45cb-402a-bc39-5d06222f3711" TYPE="ext4"
/dev/sda5: UUID="1ca0a8ee-65dd-43f2-83a8-8d843760a306" TYPE="ext4"
/dev/sda6: UUID="f6717548-2bea-4cd7-9bf6-f6b035c74ad3" TYPE="ext4"
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:29

- 1. Centos用parted分区超过2TB硬盘-分区格式化
 - 1.1.1. 1、问题描述
 - 1.1.2. 2、原因分析
 - 1.1.3. 3、解决方案
 - 1.1.4. 分区格式化

1. Centos用parted分区超过2TB硬盘-分区格式化

1.1.1. 1、问题描述

1、问题一

CentOS 6.x 在格式化大于16TB的ext4分区时，会提示如下错误：

```
mke2fs 1.41.12 (17-May-2010)
mkfs.ext4: Size of device /dev/sda1 too big to be expressed
in 32 bits using a blocksize of 4096.
```

2、问题二

CentOS 6.x 无法使用fdisk分区大于2TB的磁盘。

1.1.2. 2、原因分析

1、问题一：

原因在于CentOS 自带的 e2fsprogs 版本为1.41.12较低，无法创建16TB以上的文件系统。

2、问题二：

原因在于MBR分区表只支持2T磁盘。

1.1.3. 3、解决方案

1、问题一：

解决方案为手工下载最新的e2fsprogs源码包，进行编译安装，然后使用 `mkfs.ext4` 命令使用64位的方式进行磁盘格式化。

2、问题二：

使用parted命令将MBR分区表，改为GPT分区表。

通过输入 print 设备名 进入分区命令行模式如下图

```
[root@localhost ~]# parted /dev/sdb
GNU Parted 2.1
使用 /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) █
```

通过 partprint 打印列出当前分区设备的磁盘容量大小，如下图12.9tb

```
(parted) print
Model: LSI MRSASRoMB-8i (scsi)
Disk /dev/sdb: 12.9TB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End    Size   File system  Name  标志

```

设置磁盘分区为gpt模式， mklabel gpt 然后点击YES继续(提示磁盘的数据可能会丢失是否继续)

```
(parted) mklabel gpt
警告: The existing disk label on /dev/sdb will be destroyed and all data on
this disk will be lost. Do you want to continue?
是/Yes/否/No? yes
(parted) █
```

提示下面警告，再 mkpart primary 0 12.9TB，下一条警告忽略继续即可输入 i 或者 ignore (primary 为分区的名字，如图中所示)

```
(parted) mkpart primary 0 12.9TB
警告: The resulting partition is not properly aligned for best performance.
忽略/Ignore/放弃/Cancel? i
(parted) print
Model: LSI MRSASRoMB-8i (scsi)
Disk /dev/sdb: 12.9TB
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End    Size   File system  Name  标志
 1      17.4kB  12.9TB  12.9TB          primary
```

退出parted，设置好之后输入 quit 退出 (本经验以最简单的一个盘分一个区为例)

```
(parted)
(parted) quit
信息: You may need to update /etc/fstab.

[root@localhost ~]# █
```

1.1.4. 分区格式化

把创建好的大分区进行格式化，通过 `fdisk -l` 再次查看确认

```
[root@localhost ~]# fdisk -l

Disk /dev/sda: 1099.5 GB, 1099511103488 bytes
255 heads, 63 sectors/track, 133674 cylinders
Units = cylinders of 10065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x44274426

Device Boot Start End Blocks Id System
/dev/sda1 * 1 64 512000 83 Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2 64 133675 1073227776 8e Linux LVM

WARNING: GPT (GUID Partition Table) detected on '/dev/sdb'! The util fdisk doesn't support GPT. Use GNU Parted.

Disk /dev/sdb: 12899.4 GB, 12899397926912 bytes
255 heads, 63 sectors/track, 1560262 cylinders
Units = cylinders of 10065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x44364435

Device Boot Start End Blocks Id System
/dev/sdb1 1 267350 2147483647+ ee GPT
```



直接用 `mkfs.ext4 /dev/sdb1` 格式化会报错，在文章开头已经介绍了，这里我们先下载最新的 e2fsprogs 源码包，下载地址如下：<ftp://ftp.ntu.edu.tw/linux/kernel/people/ttyso/e2fsprogs/>

下载的版本为1.43.8，进行编码安装：

```
tar -zxvf e2fsprogs-1.43.8.tar.gz
cd e2fsprogs-1.43.8
mkdir build
cd build/
./configure (如果linux系统是最小化安装，会缺少gcc相关依赖包。./configure之前要先安装gcc，命令yum install gcc，否则configure失败下一步make操作无法进行)
make
make install
```

编译完成后使用如下命令格式化：

```
mke2fs -O 64bit, \
has_journal,extents, \
huge_file,flex_bg,uninit_bg, \
dir_nlink, \
extra_isize \
-i 4194304 \
/dev/sdb1
或者：
mkfs.ext4 -O 64bit /dev/sdb1 (建议用这个)
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-02 13:44:29

- 1. fdisk分区不超过2T的磁盘

- 1.1.1. 1、首先用fdisk -l发现待分区的磁盘 /dev/sdb
- 1.1.2. 2、fdisk /dev/sdb对该磁盘进行分区，输入m并回车
- 1.1.3. 3、输入n并回车，n是“new”新建分区的意思
- 1.1.4. 4、输入p并回车
- 1.1.5. 5、输入数字1并回车
- 1.1.6. 6、采用默认值一直回车会将整个20G都分给/dev/sdb1
- 1.1.7. 7、输入w "write"并回车，意思是对刚才的结果进行保存
- 1.1.8. 8、再次使用fdisk -l查看分区的结果

1. fdisk分区不超过2T的磁盘

不超过2T的磁盘都可以使用fdisk命令进行分区，如果超过2T，用fdisk每次最多也只能分2T的空间

1.1.1. 1、首先用fdisk -l发现待分区的磁盘 /dev/sdb

```
# fdisk -l
```

```
[root@redis-150-15 ~]# fdisk -l

Disk /dev/sda: 53.7 GB, 53687091200 bytes
255 heads, 63 sectors/track, 6527 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00075bbf

      Device Boot      Start        End      Blocks   Id  System
/dev/sda1  *           1         64     512000   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2          64        6528    51915776   8e  Linux LVM

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/volGroup-lv_root: 51.1 GB, 51078234112 bytes
255 heads, 63 sectors/track, 6209 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/volGroup-lv_swap: 2080 MB, 2080374784 bytes
255 heads, 63 sectors/track, 252 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

1.1.2. 2、fdisk /dev/sdb对该磁盘进行分区，输入m并回车

```
# fdisk /dev/sdb
```

```
[root@redis-150-15 ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x316f6bf1.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): n
```

1.1.3. 3、输入n并回车， n是“new”新建分区的意思

```
Command (m for help): m
Command action
  a    toggle a bootable flag
  b    edit bsd disklabel
  c    toggle the dos compatibility flag
  d    delete a partition
  l    list known partition types
  m    print this menu
  n    add a new partition
  o    create a new empty DOS partition table
  p    print the partition table
  q    quit without saving changes
  s    create a new empty Sun disklabel
  t    change a partition's system id
  u    change display/entry units
  v    verify the partition table
  w    write table to disk and exit
  x    extra functionality (experts)

Command (m for help): n
```

Baidu 经验
jingyan.baidu.com

1.1.4. 4、输入p并回车

```
Command (m for help): n
Command action
  e    extended
  p    primary partition (1-4)
p
```

1.1.5. 5、输入数字1并回车

```
p
Partition number (1-4): 1
```

1.1.6. 6、采用默认值一直回车会将整个20G都分给/dev/sdb1

```
First cylinder (1-2610, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2610, default 2610):
Using default value 2610
```

1.1.7. 7、输入w "write"并回车，意思是将刚才的结果进行保存

```
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2610, default 2610):
Using default value 2610

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@redis-150-15 ~]#
```

1.1.8. 8、再次使用fdisk -l查看分区的结果

如图分的新区为 /dev/sdb1 , 创建的新区格式化后就可以挂载使用了

```
[root@redis-150-15 ~]# fdisk -l

Disk /dev/sda: 53.7 GB, 53687091200 bytes
255 heads, 63 sectors/track, 6527 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00075bbf

      Device Boot      Start        End      Blocks   Id  System
/dev/sda1  *           1         64     512000   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            64        6528    51915776   8e  Linux LVM

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x316f6bf1

      Device Boot      Start        End      Blocks   Id  System
/dev/sdb1            1        2610    20964793+ jingyan.baidu.com
```

```
# reboot
--- 重启一下，防止报错：Could not stat /dev/sda4 --- No such file or directory
```

```
[root@redis-150-15 ~]# mkfs -t ext3 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1310720 inodes, 5241198 blocks
262059 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
160 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
      4096000

writing inode tables: done
creating journal (32768 blocks): done
writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 39 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@redis-150-15 ~]# mount /dev/sdb1 /data
[root@redis-150-15 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/volGroup-lv_root  47G  5.1G  40G  12% /
tmpfs          238M  72K  238M  1% /dev/shm
/dev/sda1       485M  35M  426M  8% /boot
/dev/sdb1       20G  173M  19G  1% /data
[root@redis-150-15 ~]#
```

```
# mkfs -t ext3 /dev/sdb1
```

```
# mount /dev/sdb1 /data  
# df -hT
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:28

- 1. Linux中bond网卡绑定方法

1. Linux中bond网卡绑定方法

首先在虚拟机上添加俩块虚拟网卡（具体做法不在赘述）

然后为网卡添加配置文件。文件内容如下：

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
# vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
DEVICE=eth0
HWADDR=00:0C:29:F7:23:19
TYPE=Ethernet
UUID=10323900-281a-49c9-a231-b8958f4fe8ea
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
NETMASK=255.255.255.0
SLAVE=yes
MASTER=bond0
~
~
~
```

```
DEVICE=eth1
HWADDR=00:0C:29:F7:23:23
TYPE=Ethernet
UUID=567b705b-e86c-48bb-acfb-a53748ab448e
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
NETMASK=255.255.255.0
SLAVE=yes
MASTER=bond0
~
~
```

HWADDR 对应mac地址 ifconfig -a 可以查看 UUID 对应网卡UUID [Linux 查看网卡UUID](#)

两块网卡的配置文件差不多，就是mac地址和设备名称还有UUID需要更改。

接下来就是要添加bond网卡的配置文件

```
# vi /etc/sysconfig/network-scripts/ifcfg-bond0
```

```
IPADDR=192.168.2.184
NETMASK=255.255.255.0
GATEWAY=192.168.2.2
TYPE=Ethernet
BOOTPROTO=none
DEVICE=bond0
ONBOOT=yes
BONDING_OPTS="miimo=10 mode=1"
~
```

- Tips: BONDING_OPTS="miimo=10 mode=1" 网卡配置模式以及通讯时间 (单位为毫秒)
- 在linux中实现多网卡的绑定 介绍常见的7种Bond模式

最后重启网络服务。 (在7.0以上的版本有 NetworkManager , 需要关闭此项服务, bond网卡才能生效)

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:26

- 1. Linux-查看网卡UUID
 - 1.1.1. Linux 查看网卡UUID

1. Linux-查看网卡UUID

1.1.1. Linux 查看网卡UUID

1、首先我们查看一下 `nmcli` 是哪个软件包提供的

```
# yum provides "*/nmcli"
```

```
[root@localhost ~]# yum provides "*/nmcli"
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.163.com
 * extras: mirrors.163.com
 * updates: mirrors.cn99.com
1:NetworkManager-0.8.1-113.el6.x86_64 : Network connection manager and user applications
Repo        : base
Matched from:
Filename    : /usr/bin/nmcli
```

2、安装NetworkManager服务

```
# yum -y install NetworkManager
```

3、启动NetworkManager服务

```
# service NetworkManager start
```

```
[root@localhost ~]# service NetworkManager start
Setting network parameters... [ OK ]
Starting NetworkManager daemon: [ OK ]
```

4、查看网卡UUID

```
# nmcli con
```

Tips:如果有发现有报错

```
-- 查看/var/log/messages日志
# vi /var/log/messages
tips:按两下大写的G跳到最后一页
报错:
```
Apr 20 14:53:05 localhost NetworkManager[2013]: <info> NetworkManager (version 0.8.1-113.el6) is starting...
```

```

Apr 20 14:53:05 localhost NetworkManager[2013]: <info> Read config file /etc/NetworkManager.conf
Apr 20 14:53:05 localhost NetworkManager[2013]: <error> [1492671185.606620] [nm-dbus-manager.c:278] nm_dbus_manager_init_bus(): Could not get the system bus. Make sure the message bus daemon is running! Message: Failed to connect to socket /var/run/dbus/system_bus_socket: Connection refused
...
-- 需要先启动messagebus, 再启动NetworkManager
/etc/init.d/messagebus start
service NetworkManager start

```

```

[root@localhost ~]# /etc/init.d/messagebus start
Starting system message bus: [OK]
[root@localhost ~]#
[root@localhost ~]# service NetworkManager start
Setting network parameters... [OK]
Starting NetworkManager daemon: [OK]

```

## 5、运行 nmcli

```
nmcli con
```

```
[root@localhost ~]# nmcli con
Error: Can't obtain connections: settings services are not running.
```

报错:

```
vi /var/log/messages
```

## 发现有报错

```

Mar 15 10:59:00 localhost NetworkManager[1878]: <info> NetworkManager (version 0.8.1-113.el6) is starting...
Mar 15 10:59:00 localhost NetworkManager[1878]: <info> Read config file /etc/NetworkManager/NetworkManager.conf
Mar 15 10:59:00 localhost NetworkManager[1878]: <info> modem-manager is now available
Mar 15 10:59:00 localhost NetworkManager[1878]: <info> monitoring kernel firmware directory '/lib/firmware'.
Mar 15 10:59:00 localhost NetworkManager[1878]: <info> Bonding/Bridging/VLAN support disabled
Mar 15 10:59:00 localhost NetworkManager[1878]: ifcfg-rh: Acquired D-Bus service com.redhat.ifcfggrh1
Mar 15 10:59:00 localhost NetworkManager[1878]: <info> Loaded plugin ifcfg-rh: (c) 2007 - 2008 Red Hat, Inc. To report bugs please use the NetworkManager mailing list.
Mar 15 10:59:00 localhost NetworkManager[1878]: <info> Loaded plugin keyfile: (c) 2007 - 2008 Red Hat, Inc. To report bugs please use the NetworkManager mailing list.
Mar 15 10:59:00 localhost NetworkManager[1878]: <error> [1521125940.826929] [main.c:708] main(): failed to initialize the network manager: Could not load plugin 'ibft': /usr/lib64/NetworkManager/libnm-settings-plugin-ibft.so: undefined symbol: g_slist_free_full
Mar 15 10:59:00 localhost NetworkManager[1878]: <info> exiting (error)

```

`g_slist_free_full` 属于 `glist` 的一个方法, 缺少 `glist`

## 解决方法:

```
yum -y install glib2-devel
```

## 6、此时再运行nmcli即可查看网卡UUID

```
service NetworkManager start
nmcli con
```

```
[root@localhost ~]# nmcli con
NAME UUID
TIMESTAMP-REAL
Auto eth1 567b705b-e86c-48bb-acfb-a53748ab448e 802-3-ethernet system
Thu 15 Mar 2018 11:01:25 AM EDT
System eth0 10323900-281a-49c9-a231-b8958f4fe8ea 802-3-ethernet system
Thu 15 Mar 2018 11:01:23 AM EDT
NAME UUID
TIMESTAMP-REAL
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:30

- 1. Linux-搭建-ElasticSearch5-x
  - 1.1. 一、ElasticSearch5.2.2 安装

## 1. Linux-搭建-ElasticSearch5-x

### 1.1. 一、ElasticSearch5.2.2 安装

#### 1、检查 JDK 版本（必须使用 jdk1.8 版本）

```
java -version
```

```
[root@localhost ~]# java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

#### 2、初始化环境

```
service iptables status
service iptables stop
chkconfig iptables off
yum install -y wget vim
vim /etc/selinux/config //SELINUX=enforcing 改为 SELINUX=disabled
reboot
```

#### 3、下载安装ElasticSearch5.2.2

```
cd /usr/local/
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.5.2.tar.gz
tar -zvxf elasticsearch-5.5.2.tar.gz
cd elasticsearch-5.5.2/
vim config/elasticsearch.yml //添加如下配置
```

```
配置es的集群名称，es会自动发现在同一网段下的es，如果在同一网段下有多个集群，就可以用这个属性来区分不同的集群
cluster.name: elasticsearch
节点名称
node.name: "es_client"
bootstrap.memory_lock: false
bootstrap.system_call_filter: false
network.host: 0.0.0.0
```

```
[elasticsearch@localhost elasticsearch-5.5.2]$ vim ./config/elasticsearch.yml

配置es的集群名称，es会自动发现在同一网段下的es,如果在同一网段下有多个集群,就可以用这个属性来区分不同的集群
cluster.name: elasticsearch
节点名称
node.name: "es_client"
bootstrap.memory_lock: false
bootstrap.system_call_filter: false
network.host: 0.0.0.0
===== Elasticsearch Configuration =====
#
NOTE: Elasticsearch comes with reasonable defaults for most settings.
Before you set out to tweak and tune the configuration, make sure you
understand what are you trying to accomplish and the consequences.
#
The primary way of configuring a node is via this file. This template lists
the most important settings you may want to configure for a production cluster.
#
Please consult the documentation for further information on configuration options:
http://www.elasticsearch.org/guide/en/elasticsearch/reference/index.html
```

#### 4、因为安全问题 elasticsearch 不让用root用户直接运行，所以要创建新用户并运行

```
groupadd elasticsearch
useradd elasticsearch -g elasticsearch -p elasticsearch
chown -R elasticsearch:elasticsearch /usr/local/elasticsearch-5.5.2 //更改用户权限
su elasticsearch
./bin/elasticsearch
```

#### Tips：启动发现报错

```
led=false, -Dlog4j.disable.jmx=true, -Dlog4j.skipJansi=true, -XX:+HeapDumpOnOutOfMemoryError, -Des.path.home=/usr/local/elasticsearch-5
[2018-08-21T11:23:55.687][INFO][o.e.p.PluginsService] [es_client] loaded module [aggs-matrix-stats]
[2018-08-21T11:23:55.687][INFO][o.e.p.PluginsService] [es_client] loaded module [ingest-common]
[2018-08-21T11:23:55.687][INFO][o.e.p.PluginsService] [es_client] loaded module [lang-expression]
[2018-08-21T11:23:55.687][INFO][o.e.p.PluginsService] [es_client] loaded module [lang-groovy]
[2018-08-21T11:23:55.687][INFO][o.e.p.PluginsService] [es_client] loaded module [lang-mustache]
[2018-08-21T11:23:55.687][INFO][o.e.p.PluginsService] [es_client] loaded module [lang-painless]
[2018-08-21T11:23:55.692][INFO][o.e.p.PluginsService] [es_client] loaded module [parent-join]
[2018-08-21T11:23:55.692][INFO][o.e.p.PluginsService] [es_client] loaded module [percolator]
[2018-08-21T11:23:55.692][INFO][o.e.p.PluginsService] [es_client] loaded module [reindex]
[2018-08-21T11:23:55.692][INFO][o.e.p.PluginsService] [es_client] loaded module [transport-netty3]
[2018-08-21T11:23:55.693][INFO][o.e.p.PluginsService] [es_client] loaded module [transport-netty4]
[2018-08-21T11:23:55.693][INFO][o.e.p.PluginsService] [es_client] no plugins loaded
[2018-08-21T11:23:57.950][INFO][o.e.d.DiscoveryModule] [es_client] using discovery type [zen]
[2018-08-21T11:23:58.809][INFO][o.e.n.Node] [es_client] initialized
[2018-08-21T11:23:58.809][INFO][o.e.n.Node] [es_client] starting ...
[2018-08-21T11:23:59.026][INFO][o.e.t.TransportService] [es_client] publish_address {192.168.1.29:9300}, bound_addresses {[::]:9300}
[2018-08-21T11:23:59.041][INFO][o.e.b.BootstrapChecks] [es_client] bound or publishing to a non-loopback or non-link-local address,
hecks
ERROR: [3] bootstrap checks failed
[1]: max file descriptors [4096] for elasticsearch process is too low, increase to at least [65536]
[2]: max number of threads [1024] for user [elasticsearch] is too low, increase to at least [2048]
[3]: max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]
[2018-08-21T11:23:59.056][INFO][o.e.n.Node] [es_client] stopping ...
[2018-08-21T11:23:59.060][INFO][o.e.n.Node] [es_client] stopped
[2018-08-21T11:23:59.089][INFO][o.e.n.Node] [es_client] closing ...
[2018-08-21T11:23:59.106][INFO][o.e.n.Node] [es_client] closed
[elasticsearch@localhost elasticsearch-5.5.2]$
```

切换成root用户执行以下命令

```
su root
vim /etc/security/limits.conf
//添加如下内容:
* soft nofile 65536
* hard nofile 131072
* soft nproc 2048
* hard nproc 4096
```

```
End of file
* soft nofile 65536
* hard nofile 131072
* soft nproc 2048
* hard nproc 4096
```

```
vim /etc/security/limits.d/90-nproc.conf
//修改如下内容:
* soft nproc 1024
#修改为
* soft nproc 2048
```

```
See rhbz #432903 for reasoning.

* soft nproc 2048
root soft nproc unlimited
~
~
```

```
vim /etc/sysctl.conf
添加下面配置:
vm.max_map_count=655360
并执行命令:
sysctl -p
```

## 重新启动ElasticSearch

```
su elasticsearch
./bin/elasticsearch
./bin/elasticsearch -d //后台运行
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:25

- 1. 搭建-ElasticSearch-6-1-3分布式集群
  - 1.1.1. 单机多节点集群配置

## 1. 搭建-ElasticSearch-6-1-3分布式集群

ELasticsearch 6.1.3要求JDK版本最低为 1.8

mac上的本机ip为192.168.1.111,设为 master 节点, 配置如下:

```
cluster.name: my application
node.name: node-111

network.host: 192.168.1.111
http.port: 9200

http.cors.enabled: true
http.cors.allow-origin: "*"

node.master: true
node.data: true
discovery.zen.ping.unicast.hosts: ["192.168.1.111"]
```

Ubuntu机器的ip位192.168.1.102, 配置如下:

```
cluster.name: my application
node.name: node-102

network.host: 192.168.1.102
http.port: 9200

http.cors.enabled: true
http.cors.allow-origin: "*"

node.master: false
node.data: true
discovery.zen.ping.unicast.hosts: ["192.168.1.111"]
```

先启动mac上的master, 再启动Ubuntu上的slave节点, 观察输出, 会有一个node-102探测到master节点的提示:

```
2017-06-07T11:33:39,369][INFO][o.e.c.s.ClusterService] [node-102] detected_master {node-111}{3dQd1RRVTMiKdTckM68nPQ}{H6Zu7PAQRWewUBc1lsQWTQ}{192.168.1.111}{192.168.1.111:9300}, added {{node-111}{3dQd1RRVTMiKdTckM68nPQ}{H6Zu7PAQRWewUBc1lsQWTQ}{192.168.1.111}{192.168.1.111:9300}}, reason: zen-disco-receive(from master [master {node-111}{3dQd1RRVTMiKdTckM68nPQ}{H6Zu7PAQRWewUBc1lsQWTQ}{192.168.1.111}{192.168.1.111:9300} committed version [8]])'
```

访问 head , 一个master一个slave组成集群, 界面如下:



<http://blog.csdn.net/napoay>

### 1.1.1. 单机多节点集群配置

如果想要在一台机器上启动多个节点，步骤如下：

- 1、复制一份Elasticsearch的安装包
- 2、修改端口，比如一个是9200，一个是9205
- 3、删除data目录下的数据(如果是新解压的安装包就不必了)

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-02 13:44:25

- 1. Elasticsearch-5-5-0安装head插件
  - 1.1. 1、npm安装grunt
  - 1.2. 2、下载官方源码
  - 1.3. 3、修改head源码
  - 1.4. 4、运行Elasticsearch-head

## 1. Elasticsearch-5-5-0安装head插件

Tips:环境

- 楼主博客: <https://zhousiwei.gitee.io/>
- Windows7 , JDK-1.8 , ElasticSearch-5.5.0 , node-v8.1.2 , git

windows 用户建议安装个 **TortoiseGit**客户端

下载 Elasticsearch 5.5.0 <https://www.elastic.co/downloads/elasticsearch>

- zip 和 tar 格式是各种系统都通用的，解压之后启动 Elasticsearch 即可。

### 1.1. 1、npm安装grunt

```
$ npm install -g grunt-cli
$ grunt -version
```

```
D:\nodejs>npm install grunt-cli
grunt-cli@1.2.0 node_modules\grunt-cli
├── grunt-known-options@1.1.0
├── nopt@3.0.6 (abbrev@1.0.9)
├── resolve@1.1.7
└── findup-sync@0.3.0 (glob@5.0.15)

D:\nodejs>grunt -version
grunt-cli v1.2.0
grunt v1.0.1

D:\nodejs>
```

注意这里安装的时候路径一定要切换到 **grunt-cli** 文件夹下面

这些是 **Grunt file.js** 中引用的，分别下载

```
$ npm install grunt-contrib-copy
$ npm install grunt-contrib-concat
$ npm install grunt-contrib-uglify
$ npm install grunt-css
```

### 1.2. 2、下载官方源码

官方源码 <https://github.com/mobz/elasticsearch-head>

```
$ git clone git://github.com/mobz/elasticsearch-head.git
```

### 1.3. 3、修改head源码

由于 head 的代码还是2.6版本的，直接执行有很多限制，比如无法跨机器访问。因此需要用户修改两个地方：

目录： head/Gruntfile.js : 增加 hostname 属性，设置为 \*

```
connect: {
 server: {
 options: {
 port: 9100,
 hostname: '*',
 base: '/',
 keepalive: true
 }
 }
}
```

#### 3.1、修改连接地址：

目录： head/\_site/app.js

修改 head 的连接地址: this.base\_uri = this.config.base\_uri || this.prefs.get("app-base\_uri") || "http://localhost:9200";

把 localhost 修改成你es的服务器地址，如： this.base\_uri = this.config.base\_uri || this.prefs.get("app-base\_uri") || "http://10.10.10.10:9200";

### 1.4. 4、运行Elasticsearch-head

修改 elasticsearch 的参数

修改一下es使用的参数。编辑 config/elasticsearch.yml

注意，设置参数的时候:后面要有空格！

```
换个集群的名字，免得跟别人的集群混在一起
cluster.name: es 5.0 test

换个节点名字
node.name: node-101

修改一下ES的监听地址，这样别的机器也可以访问
network.host: 0.0.0.0

默认的就好
```

```
http.port: 9200

增加新的参数，这样head插件可以访问es
http.cors.enabled: true
http.cors.allow-origin: "*"
```

开启 ES-5.5.0 : ElasticSearch-5.0.0\bin\elasticsearch.bat

然后在head源码目录中，执行 npm install 进行安装：

```
$ npm install
```

- 初次运行安装可能会报警告或错误。可以重新运行一次 npm install

最后，使用 grunt 命令在head源码目录下启动：

```
$ grunt server
```

```
E:\projects\elasticsearch-header\elasticsearch-head>grunt server
Running "connect:server" (connect) task
Waiting forever http://blog.csdn.net/qq_30930805
Started connect web server on localhost:9100.
```

效果如图：

访问 <http://localhost:9100>

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间： 2019-07-02 13:44:25

- 1. ES插件BigDesk安装

## 1. ES插件BigDesk安装

bigdesk Github地址: <https://github.com/lukas-vlcek/bigdesk>

bigdesk支持的最新版本的ES是1.3.0 ... 1.3.x

然而想要在新版本ES上安装 ( 2.x.x ) , 项目中遇到过一些BUG, 在ES在最新版本中有修复, 所以采用了 2.3.0 的版本

head 插件查看数据和集群其他一些挺方便, 但是对集群状况监控不够细致, 而且使用中有一些BUG问题, 一般会和 bigdesk 、 kopf 结合使用。

下载安装的时候出现问题:

```
Installing lukas-vlcek/bigdesk ...
Trying https://github.com/lukas-vlcek/bigdesk/archive/master.zip ...
DownloadingDONE
Verifying https://github.com/lukas-vlcek/bigdesk/archive/master.zip checksums if available ...
NOTE: Unable to verify checksum for downloaded plugin (unable to find .sha1 or .md5 file to verify)
ERROR: Could not find plugin descriptor 'plugin-descriptor.properties' in plugin zip
```

解决办法:

1. 下载bigdesk的master的zip包
2. 解压到安装路径下 `elasticsearch-2.3.0/plugins/` 目录下, 会有一个 `bigdesk` 或者 `bigdesk-master` (可以改成 `bigdesk`) 目录
3. 提示没有找到 `plugin-descriptor.properties` 文件, 那就创建这个文件, 并写入以下内容

```
description=bigdesk
version=bigdesk
name=bigdesk
site=true
```

保存OK, 大功告成。

此时目录树为 `plugins/bigdesk/_site/*`

`plugin-descriptor.properties` 文件放置在 `plugins/bigdesk/` 下解压的文件放在 `/_site/` 目录下

然后在本地浏览器中输入 [http://localhost:9200/\\_plugin/bigdesk/](http://localhost:9200/_plugin/bigdesk/)

注：\_plugin后面的bigdesk就是plugins下目录名字，若是bigdesk-master，则需要写这个名字

本以为会正常运行，然而出现了一个js弹框，有这样的内容

```
Bigdesk may not work correctly!
Found ES node version:2.3.0
Require ES node version:>= 1.0.0.RC1
```

解决办法：

定位到 `plugins/bigdesk/_site/js/store` 目录，打开 `BigdeskStore.js`，定位到 142 行

```
return (major == 1 && minor >= 0 && maintenance >= 0 && (build != 'Beta1' || build
!= 'Beta2'));
```

看到这么一行，把 `major ==1` 改成 `major >=1`，然后保存。

浏览器重新打开 [http://localhost:9200/\\_plugin/bigdesk/](http://localhost:9200/_plugin/bigdesk/)

OK.可以看到正常运行了。

分享一下楼主打包的文件。插件包含了head插件和bigdesk插件 head插件是单独运行的

[http://download.csdn.net/download/qq\\_30930805/10233395](http://download.csdn.net/download/qq_30930805/10233395) 其中head插件基于nodejs运行，具体参考

[http://blog.csdn.net/qq\\_30930805/article/details/74990324](http://blog.csdn.net/qq_30930805/article/details/74990324)

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间： 2019-07-02 13:44:25

- 1. logstash-收集tomcat-log

## 1. logstash-收集tomcat-log

```

input {
 file {
 path => "/usr/Elastic/test_tomcat_logs/localhost_access_log*.txt"
 start_position => "beginning" #从文件开始处读取
 type => "tomcat_access_log"
 #since_db_path => "/dev/null"
 }
}

filter {
 if [type] == "tomcat_access_log" {
 grok {
 match => { "message" => "%{COMMONAPACHELOG}" }
 }

 kv {
 source => "request"
 field_split => "&?"
 value_split => "="
 }

 urldecode {
 all_fields => true
 }
 }
}

#定义时间戳的格式
date {
 match => ["timestamp", "yyyy-MM-dd-HH:mm:ss"]
 locale => "cn"
}
}

output {
 if [type] == "tomcat_access_log" {
 elasticsearch {
 action => "index" #es要执行的动作
 hosts => ["192.168.1.114:9201", "192.168.1.114:9202"]
 #事件要被写进的索引，可是动态的用%{foo}语句
 index => "logstash-tomcat_access_logs-%{+YYYY.MM.dd}"
 #protocol => "http" #这个打开会报错
 #index => "system-message-%{+YYYY.MM.dd}"
 #一个默认的es mapping 模板将启用 (除非设置为false 用自己的template)
 #manage_template => true
 #有效的filepath 设置自己的template文件路径, 不设置就用已有的
 #template => "/data1/cloud/logstash-5.5.1/filebeat-template.json"
 #template_name => "my_index" #在es内部模板的名字
 }
 }
}

```

```
#template_overwrite => true
#document_type => "log"
user => ""
password => ""
}
}

stdout {
 codec => rubydebug
}
}
```

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟修订时间: 2019-07-02 13:44:25

- 1. Kibana使用高德地图
  - 1.1.1. 说明
  - 1.1.2. 一、修改配置

## 1. Kibana使用高德地图

### 1.1.1. 说明

- 目前 Kibana 默认自带的地图全部是英文，更换高德地图对地图汉化

### 1.1.2. 一、修改配置

1、编辑 kibana 配置文件 `kibana.yml`，最后面添加

```
tilemap.url: 'http://webrd02.is.autonavi.com/appmaptile?lang=zh_cn&size=1&scale=1&tyle=7&x={x}&y={y}&z={z}'
```

删除 `kibana` 目录下的 `optimize/bundles` 文件夹后，重启，如果 `bundles` 目录没有新文件生成，注意权限哈！

2、在 logstash 服务器下载 IP 地址归类查询库

```
wget http://geolite.maxmind.com/download/geoip/database/GeoLite2-City.mmdb.gz
gunzip GeoLite2-City.mmdb.gz
```

3、编辑 logstash 配置文件

```
input {
 stdin {}
}

filter {
 geoip {
 source => "message"
 target => "geoip"
 database => "/usr/local/logstash-5.1.1/config/GeoLite2-City.mmdb"
 add_field => "[geoip][coordinates]", "%{[geoip][longitude]}"
 add_field => "[geoip][coordinates]", "%{[geoip][latitude]}"
 }
}

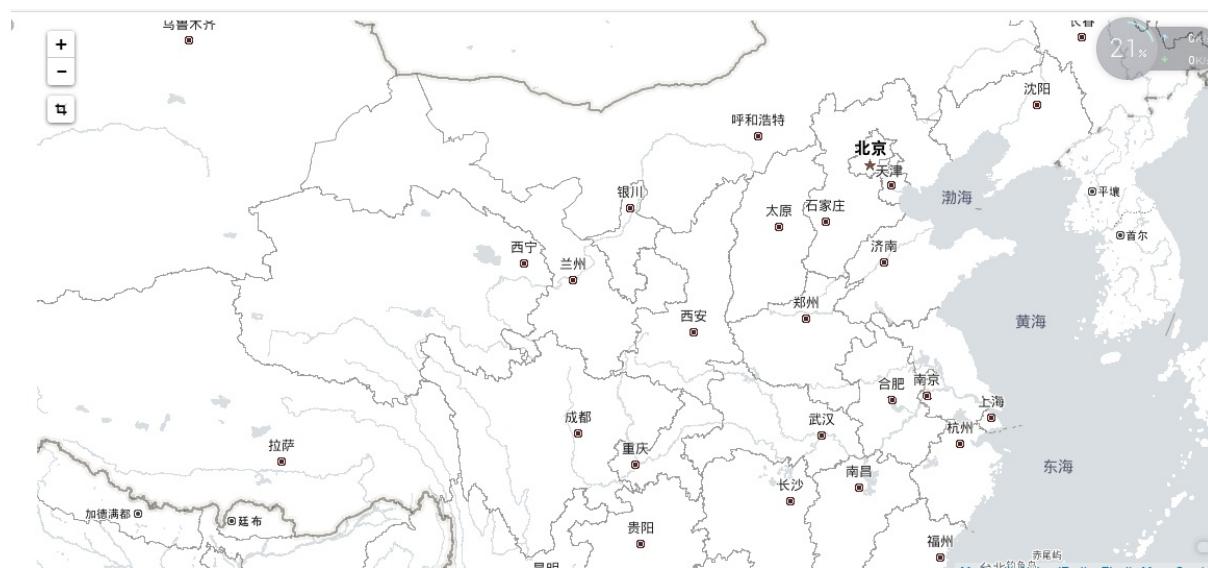
output {
 elasticsearch {
 hosts => "172.16.11.199"
 index => "logstash-map-%{+YYYY.MM.dd}"
 }
}
```

## 配置解释

这里为了测试，数据通过终端手动输入

- geoip : IP查询插件
- source : 需要通过geoip插件处理的field，一般为ip，这里因为通过控制台手动输入的是ip所以直接填message，生成环境中如果查询nginx访问用户，需先将客户端ip过滤出来，然后这里填clientip即可
- target : 解析后的Geoip地址数据，应该存放在哪一个字段中，默认是geoip这个字段
- database : 指定下载的数据库文件
- add\_field : 这里两行是添加经纬度，地图中地区显示是根据经纬度来识别

## 效果图



Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:25

- 1. ELK之本地Grok-Debugger配置
  - 1.1. 1、Ruby环境安装
  - 1.2. 2、GrokDebug的安装

## 1. ELK之本地Grok-Debugger配置

### 1.1. 1、Ruby环境安装

#### 1.1、清理已安装过的

```
yum erase ruby ruby-libs ruby-mode ruby-rdoc ruby-irb ruby-ri ruby-docs
yum remove ruby
```

#### 1.2、Ruby的安装( 别使用ruby最新的2.2或者2.3的版本, 可能出现部分组件无法安装 )

```
yum install -y wget unzip
cd /usr/local
yum -y install openssl-devel gcc
wget https://ruby.taobao.org/mirrors/ruby/2.1/ruby-2.1.7.tar.gz
tar zxf ruby-2.1.7.tar.gz
cd ruby-2.1.7
./configure --prefix=/usr/local/ruby2.1.7
make && make install
echo 'export PATH=/usr/local/ruby2.1.7/bin:$PATH'>>/etc/profile
source /etc/profile
```

#### 1.3、RubyGems工具安装

```
cd /usr/local
wget http://rubygems.global.ssl.fastly.net/rubygems/rubygems-2.6.2.tgz
tar zxf rubygems-2.6.2.tgz
cd rubygems-2.6.2
ruby setup.rb
```

#### 1.4、替换gem源,由于网络环境的问题, 访问官方源非常慢, 使用淘宝的gem源

```
gem sources --add https://ruby.taobao.org/ --remove https://rubygems.org/
gem sources -l
```

### 1.2. 2、GrokDebug的安装

#### 2.1、下载源码包

```
mkdir /usr/local/grokgdb
cd /usr/local/grokgdb
```

```
wget https://code.load.github.com/nickethier/grokdebug/zip/master
unzip master
mv grokdebug-master/* .
rm -rf grokdebug-master/
```

## 2.2、Ruby组件安装(以下组件都对版本有相应的要求)

```
--- 查看缺少的组件
ruby config.ru
--- 就会提示组件及对应的版本
gem install bundler
gem install cabin -v=0.5.0
gem install haml -v=3.1.7
gem install jls-grok -v=0.10.10
gem install json -v=1.7.5
gem install kgio -v=2.8.0
gem install rack -v=1.4.1
gem install rack-protection -v=1.2.0
gem install raindrops -v=0.11.0
gem install shotgun -v=0.9
gem install tilt -v=1.3.3
gem install sinatra -v=1.3.3
gem install unicorn -v=4.6.3
```

## 2.3、启动服务

```
cd /usr/local/grokbug
nohup bundle exec unicorn -p 8080 -c ./unicorn &
```

## 2.4、关闭防火墙

```
service iptables stop
chkconfig iptables off
```

## 2.5、替换Google的jquery源

```
cd /usr/local/grokbug
cd views
```

## 执行下面5条语句

```
sed -i 's#/ajax.googleapis.com/ajax/libs/jquery/1.8.1/jquery.min.js##/lib.sinaap
p.com/js/jquery/1.8.1/jquery.min.js#g' index.haml

sed -i 's#/ajax.googleapis.com/ajax/libs/jqueryui/1.9.2/jquery-ui.min.js##/lib.s
inaapp.com/js/jquery-ui/1.9.2/jquery-ui.min.js#g' index.haml

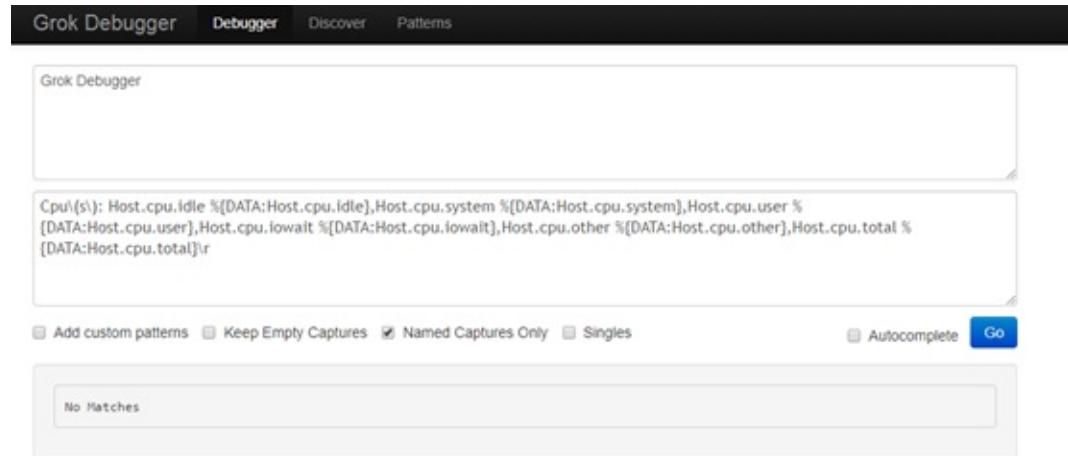
sed -i 's#/ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js##/lib.sinaap
```

```
p.com/js/jquery/1.7.2/jquery.min.js#g' patterns.haml

sed -i 's#/ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/themes/ui-lightness/jquery-ui.css#/lib.sinaapp.com/js/jquery-ui/1.9.0/themes/ui-lightness/jquery-ui.css#g' layout.haml

sed -i 's#/ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js#/lib.sinaap.p.com/js/jquery/1.7.2/jquery.min.js#g' discover.haml
```

测试，通过web路径访问测试了 <http://xxx:8080>



"I grok in fullness." Robert A. Heinlein, Stranger in a Strange Land

Copyright © 試毅-思伟 2019 all right reserved, powered by 試毅-思伟 修订时间: 2019-07-02 13:44:25