

JWS 配置标准化

修订纪录

文档版本	修订时间	内容摘要	作者
v1.0.0	2013-03-15	初稿	章卫华
V1.1.0	2013-11-27	修订配置项命名	梁瑞麟
V1.2.0	2014-04-03	修改配置文件规范	方荣耀

目录

一、	背景	3
二、	项目名称规范	3
三、	配置文件功能划分	4
3.1	配置文件功能划分	4
3.2	标准目录结构示例	5
四、	配置文件规范	6
4.1	目录规范	6
4.2	编码规范	7
4.3	命名规范	7
五、	配置项规范	7
5.1	常规配置项命名规范	7
5.2	变量配置项命名规范	8
5.3	实例案例演示	8
5.4	常规配置项其他规范	9
六、	日志规范	10
七、	项目启动规范	11

一、背景

目前游戏中心各业务在使用 JWS 框架过程中，没有统一的配置文件规范，各团队根据自己的喜好定义配置文件，对于各团队都有一定的学习成本，维护成本。

JWS 配置标准化主要是将配置文件按照不同的功能划分配置到对应的目录，并通过 JWS 框架 (jws1.4.2) 变量注入的方式统一管理配置文件。

采用统一 JWS 配置文件规范，便于自动化部署及维护，减少团队学习成本，提高生产环境稳定性及团队工作效率。

二、项目名称规范

项目名称规范：业务英文名称缩写+服务名称，使用下划线连接；

比如：九游发号，项目名称为：ka_web，ka_server，ka_management

命名不规范的业务需要按照规范更正，比如基础组件：

业务大类	业务英文名称	修改前 app 名称	修改后 app 名称
基础组件	gdc	management SimpleFileServer interface	gdc_management gdc_sfs gdc_interface

三、 配置文件功能划分

3.1 配置文件功能划分

功能模块		配置文件名称	备注
主控配置		application.conf	包含项目名称，include 配置信息， 不包含端口，pool 信息。
项目 jar 包依赖配置		dependencies.yml	
信息提示配置		messages	
路由配置		routes	
业务类配置 biz		business.conf caches.xml 等	由研发管理，无需运维关注
环境类配置 env	账号安全配置	account-sdk.conf	
	白名单配置	acl.conf	
	缓存类配置	cache-servers.xml	多个缓存实例，序号则依次增加
	计划任务配置	cron.conf	
	数据库配置	database-servers.conf	多个数据库实例，序号则依次增加
	http 配置	http.conf	端口，模板、压缩、解密文件配置
	日志配置	log4j.properties	
	调用外部系统配置	out-system.conf	
	秘钥配置	secret.properties	后续考虑将此配置存数据库，通过后

			台管理，目前不是 jws 框架管理。
	基础支撑服务配置	fdfs.conf ucmq.conf sphinx.conf	
	邮件配置	smtp.conf	
	动态配置	synctimer.properties	需要将此配置存数据库，通过后台管理
生产环境变量配置		define.conf	生产环境变量配置，存放在 /home/jws/app/conf 目录， 运维关注

3.2 标准目录结构示例

以下以消息组件为例：

```
[jws@game3 ~]$ tree -L 1 /home/jws/app/message/
/home/jws/app/message/
|-- conf
|-- lib
|-- modules
|-- precompiled
`-- public
```

```
[jws@game3 ~]$ tree /home/jws/app/conf/
/home/jws/app/conf/
|-- defined-message-sh.conf
`-- defined-message-st.conf
```

```
[jws@game3 ~]$ tree /home/jws/app/message/conf/
/home/jws/app/message/conf/
|-- application.conf
|-- biz
|   |-- business.conf
|   |-- cache
|   |   |-- caches.xml
|   |   |-- cluster-configs.xml
|   |-- database
|   |   |-- classes.xml
|   |   |-- cluster-configs.xml
|   |-- sphinx
|   |   |-- coreseek
|-- dependencies.yml
|-- env
|   |-- account-sdk.conf
|   |-- acl.conf
|   |-- cache-servers.xml
|   |-- cron.conf
|   |-- database-servers.conf
|   |-- fdfs.conf
|   |-- http.conf
|   |-- log4j.properties
|   |-- outer-system.conf
|   |-- secret.properties
|   |-- smtp.conf
|   |-- sphinx.conf
|   |-- ssl.jks
|   |-- ucmq.conf
|-- messages
|-- routes
```

四、 配置文件规范

4.1 目录规范

1. 业务类的配置统一存放在 biz 目录下，由研发管理，运维无需关注；
2. 环境类的配置统一存放在 env 目录下，配置项 value 配置为变量，JWS 框架通过查找 defined.conf 中对应的 value，然后通过注入的方式写到对应的配置中；
3. 开发环境，测试环境，生产环境不同的配置全部通过变量存放 defined.conf 中，运维仅关注

defined.conf 配置文件即可；

4. defined.conf 建议存放目录为：/home/jws/app/conf；

5. pid 文件建议存放目录为：/home/jws/app/pid；

4.2 编码规范

1. 所有配置文件编码都使用 utf-8；

2. message 等文件不需要用 native2ascii 转码，直接用中文；

4.3 命名规范

1. 配置文件命名规范：使用中划线连接，后缀名一般是 conf，xml，properties。如：

```
cache-servers.xml , database-servers.conf
```

2. defined.conf 命名规范：defined+项目名称+集群；如：

```
defined-message-sh.conf , defined-message-st.conf
```

错误示范：

```
account_sdk.conf , portal_interface.conf
```

正确示范：

```
account-sdk.conf , portal-interface.conf
```

五、 配置项规范

5.1 常规配置项命名规范

1. 常规配置项命名规范：项目前缀.配置项，使用小数点连接；

错误示范：

```
message_type=4
```

正确示范：

```
message.type=4
```

5.2 变量配置项命名规范

1. 变量配置项语法：常规配置项=\${变量名称}；
2. 变量名称语法：配置文件名称前缀+变量，使用小数点连接，不允许有特殊字符，此变量由研发定义；
3. defined.conf 配置语法：变量名称=value，变量名称只要在 defined.conf 配置文件中是唯一即可；
4. 存在的问题：
 - 1) defined.conf 文件行数可能会达到百位数以上，不便于管理，后续考虑通过 web 方式管理；
 - 2) defined.conf 文件变量行数可以使用 hg 包与生产包配置对比得出开发环境与生产环境配置的区别，即得出 defined.conf 文件行数，后续考虑采用工具获取；

5.3 实例案例演示

1. account-sdk.conf

```
#开放平台访问地址
openpf.baseUrl=${account-sdk.openurl}
openpf.caller=${account-sdk.opencaller}
openpf.secret=${account-sdk.opensecret}
openpf.calltimeout=${account-sdk.opentimeout}
```

openpf.baseUrl 为常规配置项，\${account-sdk.openurl}为变量配置项。

account-sdk.openurl：account-sdk 为 account-sdk.conf 的前缀，openurl 为开放平台接口地址缩写

2. defined-message-sh.conf

账号安全配置

account-sdk.openurl=http://[REDACTED]

account-sdk.opencaller=[REDACTED]

account-sdk.opensecret=[REDACTED]

account-sdk.opentimeout=15

account-sdk.openurl 为变量名称，http://api.open.uc.cn 为 value

3. 其他配置文件参考

配置文件名称	配置文件范例
account-sdk.conf	<pre> openpf.baseUrl=\${account-sdk.openurl} openpf.caller=\${account-sdk.opencaller} openpf.secret=\${account-sdk.opensecret} openpf.calltimeout=\${account-sdk.opentimeout} </pre>
acl.conf	<pre> acl.allow=\${acl.allowip} acl.deny=\${acl.denyip} </pre>
cache-servers.xml	<pre> <servers>\${cache-servers.cache01},\${cache-servers.cache02}</servers> </pre>
cron.conf	<pre> cron.loadAllPkgNameJob=\${cron.task01} </pre>

5.4 常规配置项其他规范

1. key value 类型的配置信息，如非必要，配置项前后不留空格（代码是否可以兼容末尾空格）
2. 接口地址配置项使用 http 全地址，末尾不带斜杠，不带接口地址；

错误示范：

account.access.url=[REDACTED] account.9game.cn [REDACTED]

account.access.url=http://account.9game.cn/

csbd.questionlist.url=http://csws.st.uae.uc.cn/csws/cpOrders

正确示范：

```
account.access.url=http://account.9game.cn
```

```
csbd.questionlist.url=http://csws.st.uae.uc.cn
```

3. 建议采用固定的前缀约束，如：

1) http 相关。前缀：http

2) 日志相关。前缀：log

3) url 相关。前缀：url

六、 日志规范

1. JWS 框架日志存放在 \$HOME/logs/ application.name/根目录下;

jws.log , access.log , stat.log , error.log , threadState.log , threadStack.log ;

2. 程序日志存放在\$HOME/logs/ application.name/app/目录下；

3. 不允许将产品统计需求日志写入到 JWS 框架日志，如：jws.log;

4. 日志级别默认打印 INFO 日志，不允许打印 DEBUG 日志；

5. 日志内容不要打印中文，默认按天打印；

6. 日志格式

日期 错误级别 接口名称 错误码 `错误描述`

7. 线上默认关闭 console 输出，相关配置如下：

1) application.conf 修改如下：system.out 配置 off。

jws1.4.2 之前版本，需要配置成 on，对 jws1.4.2 版本，通过 jws start 方式启动的，必须重定向系

统流（out/err）至文件，即忽略该配置项

```
application.log=INFO
```

```
application.log.path=${application.path}/conf/log4j.properties
```

```
application.log.system.out=on
```

2) log4j.properties 修改如下：

线上部署时候，控制台不打印日志。如下所示：

```
log4j.rootLogger=INFO,errorRolling,console
```

以下需要删除

```
log4j.additivity.console=false
```

```
log4j.appender.console=org.apache.log4j.ConsoleAppender
```

```
log4j.appender.console.layout=org.apache.log4j.PatternLayout
```

```
log4j.appender.console.layout.ConversionPattern=[%d{yyyy-MM-dd HH:mm:ss}] %p --- %m%n
```

七、 项目启动规范

在 start.sh 脚本后面增加 “-F /home/jws/app/conf/defined-message-sh.conf”，比如：

```
jws start -Dprecompiled=true -server -Xmx2048m -XX:MaxPermSize=128m -Ddefine.conf
```

```
"/home/jws/app/conf/defined-message-sh.conf"
```