目录

[OBD Server部署指南 1](#_Toc425799693)

[1.  概述 1](#_Toc425799694)

[1.1 目的 1](#_Toc425799695)

[1.2 适用范围 1](#_Toc425799696)

[2.  运行环境 1](#_Toc425799697)

[2.1安装JDK 1](#_Toc425799698)

[2.2检测Redis 1](#_Toc425799699)

[2.3检测Kafka 2](#_Toc425799700)

[3.  安装部署 2](#_Toc425799701)

[3.1目录结构 2](#_Toc425799702)

[3.2启动参数 2](#_Toc425799703)

[3.3配置参数 3](#_Toc425799704)

# OBD Server部署指南

## 1.  概述

### 1.1 目的

介绍OBD Server的安装部署流程，为项目实施人员提供部署指导。

### 1.2 适用范围

适用于公车管理通信系统项目实施人员。

## 2.  运行环境

### 2.1安装JDK

安装JDK1.7，并且设置环境变化JAVA\_HOME到JDK安装路径，并且添加JDK1.7的bin目录到环境变量。

### 2.2检测Redis

在命令行输入redis-cli –v

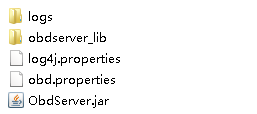


### 2.3检测Kafka

使用kafka-list-topic脚本查看当前topic状态。

## 3.  安装部署

### 3.1目录结构



logs：log文件目录

obdserver\_lib：jar包目录

log4j.properties：log配置文件

obd.properties：OBD Server配置文件

obdserver.jar：主程序

### 3.2启动参数

**命令行**

Java –jar obdserver.jar -[p] [t] [r]

-p：设置OBD Server监听端口

-t：设置Http代理端口

-r：设置OBD Server运行模式“app”：通信服务器模式，“login”：登录服务器模式

例如：启动登录服务器监听9088

java -jar obdserver.jar -p 9088 -r login

例如：启动通信服务器监听9087

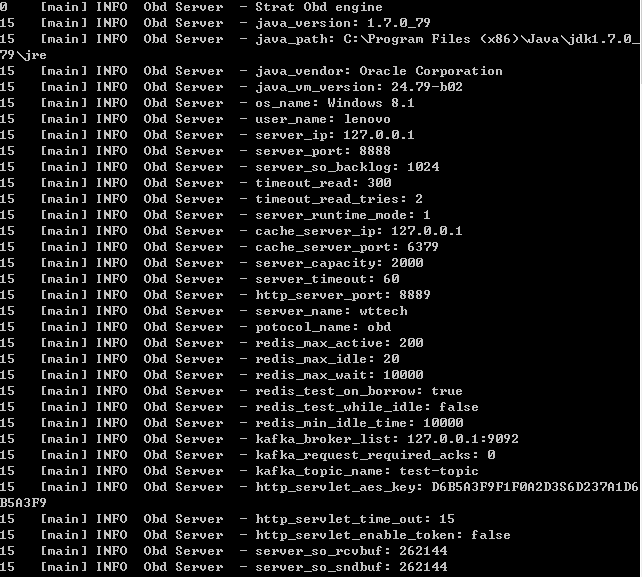
java -jar obdserver.jar -p 9087 -t 8999 -r app

注：LINUX下运行使用后台运行模式

nohup java -jar obdserver.jar -p 9087 -t 8999 -r app 1>/dev/null 2>&1 &

nohup java -jar obdserver.jar -p 9088 -r login 1>/dev/null 2>&1 &

**系统信息**



### 3.3配置参数

打开obd.properties文件配置运行参数

1. **server\_ip**

OBD Server服务地址

server\_ip=127.0.0.1

1. **server\_port**

OBD Server服务监听端口

server\_port=9087

1. **server\_so\_backlog**

OBD Server服务最大并发接入数量

server\_so\_backlog=1024

1. **timeout\_read**

OBD Server服务读取数据超时（秒）

timeout\_read=300

1. **timeout\_read\_tries**

OBD Server服务读取超时次数

timeout\_read\_tries=2

1. **server\_runtime\_mode**

OBD Server服务运行模式

server\_runtime\_mode=0

运行模式常量

0: login

1: app

1. **cache\_server\_ip**

Redis缓存地址

cache\_server\_ip=127.0.0.1

1. **cache\_server\_port**

Redis缓存监听端口

cache\_server\_port=6379

1. **cache\_server\_password**

Redis缓存口令

cache\_server\_password=

1. **server\_capacity**

OBD Server容量

server\_capacity=2000

1. **server\_timeout**

OBD Server更新超时（秒）

server\_timeout=60

1. **http\_server\_port**

嵌入http服务器监听端口

http\_server\_port=8999

1. **server\_name**

OBD Server名称

server\_name=wttech

1. **potocol\_name**

OBD通信协议名称

potocol\_name=obd

1. **redis\_max\_active**

Redis 服务器最大连接数

redis\_max\_active=200

1. **redis\_max\_idle**

Redis 服务器最大空闲连接数

redis\_max\_idle=20

1. **redis\_max\_wait**

最长等待时间（毫秒）

redis\_max\_wait=10000

1. **redis\_test\_on\_borrow**

在borrow一个jedis实例时，是否提前进行validate操作；如果为true，则得到的jedis实例均是可用的

redis\_test\_on\_borrow=true

1. **redis\_test\_while\_idle**

如果为true，表示有一个idle object evitor线程对idle object进行扫描，如果validate失败，此object会被从pool中drop掉；这一项只有在timeBetweenEvictionRunsMillis大于0时才有意义

redis\_test\_while\_idle=false

1. **redis\_min\_idle\_time**

最短空闲时间间隔（毫秒）

redis\_min\_idle\_time=10000

1. **kafka\_broker\_list**

Kafka 队列broker列表

kafka\_broker\_list=127.0.0.1:9092

1. **kafka\_request\_required\_acks**

kafaka请求认证应答机制

kafka\_request\_required\_acks=0

1. **kafka\_topic\_name**

Kafka的OBD Server保存上传数据的topic

kafka\_topic\_name=test-topic

1. **http\_servlet\_aes\_key**

webapi的AES的key

http\_servlet\_aes\_key=D6B5A3F9F1F0A2D3S6D237A1D6B5A3F9

1. **http\_servlet\_access\_list**

webapi运行访问列表

http\_servlet\_access\_list=cc88025e-227d-ab556c2a6665

1. **http\_servlet\_time\_out**

webapi超时时间间隔（秒）

http\_servlet\_time\_out=15

1. **http\_servlet\_enable\_token**

webapi是否开启token检测

http\_servlet\_enable\_token=false

1. **server\_so\_rcvbuf**

OBD Server接收缓存大小（字节）

server\_so\_rcvbuf=262144

1. **server\_so\_sndbuf**

OBD Server发送缓存大小（字节）

server\_so\_sndbuf=262144

1. **byte\_buffer\_max**

OBD命令包体最大值

byte\_buffer\_max=65535

1. **verify\_crc\_in\_each\_request**

OBD命令是否每次校验CRC值

verify\_crc\_in\_each\_request=true

1. **byteorder\_little\_endian**

OBD命令是否使用小端字节序

byteorder\_little\_endian=true