



APUSIC  
固若长城  
睿比世界

# 用户手册

金蝶Apusic应用服务器 V10.0.11标准版

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

## 版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

## 免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

## 商标声明

 Apusic 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

# 目录

- 1 前言
  - 1.1 面向对象
- 2 版本变更说明
- 3 产品简介
  - 3.1 概述
  - 3.2 平台环境
  - 3.3 系统环境
  - 3.4 JavaEE规范
  - 3.5 术语
- 4 产品清单
- 5 许可证授权
  - 5.1 标准版获取特征码
  - 5.2 嵌入式部署获取特征码
- 6 安装和运行
  - 6.1 安装
  - 6.2 卸载
  - 6.3 运行
    - 6.3.1 在Linux环境下
    - 6.3.2 在 Windows环境下
  - 6.4 停止
    - 6.4.1 在Linux环境下
    - 6.4.2 在 Windows环境下
- 7 应用管理
  - 7.1 部署应用
  - 7.2 取消部署应用
  - 7.3 访问应用
- 8 JVM管理
  - 8.1 设置 JVM 内存
- 9 配置文件说明
  - 9.1 apusic.conf 文件
    - 9.1.1 listeners 配置 (监听器)
    - 9.1.2 resources 配置 (用户权限)

- 9.1.3 executors 配置 (线程池)
- 9.1.4 endpoints 配置 (网络 IO 和协议)
  - 9.1.4.1 SSLHostConfig 元素
  - 9.1.4.2 Certificate 元素
- 9.1.5 services 配置 (虚拟主机和应用)
  - 9.1.5.1 Realm
  - 9.1.5.2 host
  - 9.1.5.3 application
- 9.1.6 servers 配置 (引用其他功能)
- 9.1.7 containers 配置 (监听器等)
- 9.2 其他配置
  - 9.2.1 共享库配置
  - 9.2.2 手动部署多个应用
  - 9.2.3 端口的修改
  - 9.2.4 开启 JMX
- 10 配置数据源
- 11 配置 HTTPS
  - 11.1 单向认证 -- 证书格式为 JKS
  - 11.2 双向认证 -- 证书格式为 JKS
  - 11.3 单向认证 -- 证书格式为 PEM
  - 11.4 双向认证 -- 证书格式为 PEM
- 12 国密配置说明
  - 12.1 环境说明
  - 12.2 应用服务器证书配置
    - 12.2.1 单向认证
    - 12.2.2 双向认证
    - 12.2.3 增加 endpoint 到 server
  - 12.3 启动应用服务器
- 13 日志配置
  - 13.1 系统日志配置
- 14 访问日志记录
  - 14.1 配置访问日志
  - 14.2 访问控制
    - 14.2.1 远程地址

- 14.2.2 远程主机名
- 14.2.3 远程CIDR
- 15 附录
  - 15.1 附录A listeners配置说明
  - 15.2 附录B resources配置说明
  - 15.3 附录C executors 配置说明
  - 15.4 附录D endpoints配置说明
    - 15.4.1 SSLHostConfig配置参数
    - 15.4.2 Certificate配置参数
  - 15.5 附录E servers 配置说明
    - 15.5.1 Realm
    - 15.5.2 host
    - 15.5.3 application
    - 15.5.4 servers
  - 15.6 附录F containers 配置说明
  - 15.7 附录G Resource数据源配置说明

# 1 前言

本文档是金蝶Apusic应用服务器V10.0.11标准版的用户手册，详细介绍金蝶Apusic应用服务器V10.0.11标准版的基本安装以及配置，提供基本操作方法以及各功能的具体步骤。

## 1.1 面向对象

本手册主要面向对象为使用金蝶Apusic应用服务器V10.0.11 进行应用开发的开发人员，生产环境的系统管理员，应用发布人员，技术运维人员等。具备以下技能可能会更好理解和使用金蝶Apusic应用服务器V10.0.11用户手册内容：

- 熟悉Linux常用命令
- 基本的系统管理任务
- 安装和管理软件

## 2 版本变更说明

本手册根据产品实际更新情况同步更新，最新版本将会包括历史版本内容或作出对应的修改说明。

日期	手册版本	适用产品	更新说明
2026年3月	V1011A01F01	AAS标准版 V10.0.11	更新用户手册

## 3 产品简介

### 3.1 概述

金蝶 Apusic 应用服务器V10.0.11标准版是一款支持JavaEE规范的web应用服务器，支持全新Servlet和JSP规范。金蝶 Apusic 应用服务器V10.0.11标准版技术先进、性能稳定、轻量简便，快速和支持随需而变的业务需求，是应用系统高效开发的首选产品。

图1：金蝶Apusic应用服务器V10.0.11标准版架构图



### 3.2 平台环境

金蝶Apusic应用服务器V10.0.11标准版平台环境。

表1：金蝶Apusic应用服务器V10.0.11标准版平台环境

操作系统	说明
Windows	Windows7、Windows10、Windows Server
Linux	CentOS、Ubuntu、SUSE、Redhat
国产平台	银河麒麟系列、中标麒麟系列、普华、中科红旗、深度等

### 3.3 系统环境

金蝶Apusic应用服务器V10.0.11标准版系统环境。

表2：金蝶Apusic应用服务器V10.0.11标准版系统环境

系统组件	系统要求
Java环境	JDK17及以上
内存	至少需要 512MB 的内存，建议1024MB以上
磁盘空间	至少需要 1024MB 的磁盘空间，建议10G以上

### 3.4 JavaEE规范

金蝶Apusic应用服务器V10.0.11标准版产品规范。

表3：金蝶Apusic应用服务器V10.0.11标准版产品规范

规范/API	版本
Servlet	6.0
JSP	4.0
Expression Language	6.0
WebSocket	2.1
Authentication	2.0

### 3.5 术语

AAS：金蝶Apusic应用服务器

AE：金蝶Apusic应用服务器标准版产品标识

APUSIC\_HOME: 金蝶Apusic应用服务器安装目录

## 4 产品清单

金蝶Apusic应用服务器V10.0.11标准版产品包说明，通常情况下打包成 `AAS-[aas_version]-[sp_version]-AE-[date].zip` 或 `AAS-[aas_version]-[sp_version]-AE-[date].tar.gz`。详看产品清单说明。

表4：金蝶Apusic应用服务器V10.0.11标准版产品包

产品包	说明
AAS-V10.0.11-SP10-AE-20251128.zip/AAS-V10.0.11-SP10-AE-20251128.tar.gz	金蝶Apusic应用服务器V10.0.11标准版安装包

### 产品包说明:

金蝶Apusic应用服务器V10.0.11标准版产品包说明。

表5：金蝶Apusic应用服务器V10.0.11标准版产品包说明

文件	说明
bin	应用服务器启动、停止等脚本所在的目录
conf	应用服务器配置文件所在的目录
lib	应用服务器JAR文件所在的目录
logs	应用服务器日志文件默认目录
webapps	部署应用程序包的目录
webapps-javaee	部署支持JakartaEE9.1及以下版本的应用程序包的目录
temp	应用服务器临时文件存放的目录
work	存放JSP编译后产生的class文件
license.xml	产品包的授权文件

## 5 许可证授权

金蝶Apusic应用服务器需要有对应的许可证才能正常使用，通常情况下，金蝶天燕会根据用户购买的产品版本配套对应的许可证。如果产品将要过期或者已经过期，可向金蝶天燕对接人员申请许可证，与此同时将产品的特征码发给对接人员。

许可证位置为 `${APUSIC_HOME}/license.xml`。

### 5.1 标准版获取特征码

1. 可以运行bin目录下命令 `apusic` 获取，类似格式:

```
apusic -ac [ethname or ip]
```

其中 `[ethname or ip]` 表示可以输入ip地址或者mac地址的名称，如可以输入常用ip地址172.24.2.42作为参数获取软件特征码(适用windows、linux):

```
apusic -ac 172.24.2.42
```

输出格式如下，Auth Code=软件特征码:

```
Auth Code=SZTY740240846
```

```
[root@linux-2-40 bin]# ./apusic.sh -ac 172.24.2.40
Using APUSIC_BASE: /opt/ams11/0325/AAMS-V10.0.11
Using APUSIC_HOME: /opt/ams11/0325/AAMS-V10.0.11
Using APUSIC_TMPDIR: /opt/ams11/0325/AAMS-V10.0.11/temp
Using JRE_HOME: /opt/java/jdk-17.0.2
Using CLASSPATH: /opt/ams11/0325/AAMS-V10.0.11/bin/bootstrap.jar:/opt/ams11/0325/AAMS-V10.0.11/bin/aas-juli.jar
Using APUSIC_OPTS: -Djava.security.egd=file:/dev/urandom
Using APUSIC_OPTS: -Djava.security.egd=file:/dev/urandom
Auth Code=SZTY740240846
[root@linux-2-40 bin]#
```

2. 也可以输入网卡名称获取特征码(适用linux，可以通过 `ifconfig` 获取网卡名称)

```
apusic -ac ens192
```

输出格式如下，Auth Code=软件特征码:

```
Auth Code=SZTY740240846
```

### 5.2 嵌入式部署获取特征码

在与springboot整合的方式下，没有获取特征码的 `apusic` 命令，需要依赖用户的应用进行获取。

如果已经有临时的授权文件可以启动应用，则先把授权文件移除，然后在启动命令中增加JVM参数 `apusic-ac`，取值为ip地址或者网卡名称，类似如下,eth3为测试机器的网卡名称，需要更加实际的网络信息修改：

```
java -jar -Dapusic-ac=eth3 boot-quickstart-1.0.jar
```

由于没有授权文件，应用启动会停止，在停止前会打印特征码信息，类似如下，Auth Code=特征码内容：

```
Auth Code=SZTY263537626  
license is invalid.
```

## 6 安装和运行

下面介绍金蝶 Apusic 应用服务器V10.0.11标准版作为一个完整的产品进行使用，`${APUSIC_HOME}` 表示产品安装的根目录。

### 6.1 安装

准备安装包 `AAS-V10.0.11-AE-20260330.zip` 或 `AAS-V10.0.11-AE-20260330.tar.gz` 。解压产品包到 `${APUSIC_HOME}` 目录下，如下图所示：

```
[root@linux-2-42 aams]# cd AAMS-10.0.10/
[root@linux-2-42 AAMS-10.0.10]# ls
bin  conf  lib  license.xml  logs  temp  webapps  webapps-javaee  work
```

### 6.2 卸载

将 `${APUSIC_HOME}` 目录下所有文件删除即可完成卸载操作。

### 6.3 运行

#### 6.3.1 在Linux环境下

将应用程序打包文件，如 `examples.war` ，放置 `${APUSIC_HOME}/webapps/` 目录下；

1、进入安装目录 `${APUSIC_HOME}/bin` ，执行 `apusic.sh run` 。

```
[root@linux-2-40 bin]# ./apusic.sh run
Using APUSIC_BASE: /opt/ams11/0325/AAMS-V10.0.11
Using APUSIC_HOME: /opt/ams11/0325/AAMS-V10.0.11
Using APUSIC_TMPDIR: /opt/ams11/0325/AAMS-V10.0.11/temp
Using JRE_HOME: /opt/java/jdk-17.0.2
Using CLASSPATH: /opt/ams11/0325/AAMS-V10.0.11/bin/bootstrap.jar:/opt/ams11/0325/AAMS-V10.0.11/bin/aas-juli.jar
Using APUSIC_OPTS: -Djava.security.egd=file:/dev/urandom
Using APUSIC_OPTS: -Djava.security.egd=file:/dev/urandom
26-Mar-2026 10:33:07.773 INFO [ApusicConsoleHandler] com.apusic.ams.core.AprLifecycleListener.lifecycleEvent The Apusic AAS Native library which
allows using OpenSSL was not found on the java.library.path: [/usr/java/packages/lib:/usr/lib64:/lib64:/lib:/usr/lib]
26-Mar-2026 10:33:07.884 INFO [ApusicConsoleHandler] com.apusic.juli.logging.LogPrintStream.println Apusic Application Server Enterprise Edition
9.0
26-Mar-2026 10:33:07.886 INFO [ApusicConsoleHandler] com.apusic.juli.logging.LogPrintStream.println Licensed to 金蝶天燕2026临时授权
26-Mar-2026 10:33:07.888 INFO [ApusicConsoleHandler] com.apusic.juli.logging.LogPrintStream.println Connctions limited to 10
26-Mar-2026 10:33:07.889 INFO [ApusicConsoleHandler] com.apusic.juli.logging.LogPrintStream.println Licensed from 2025-12-10 to 2026-12-31
26-Mar-2026 10:33:07.889 INFO [ApusicConsoleHandler] com.apusic.juli.logging.LogPrintStream.println com.apusic.juli.logging.LogPrintStream.println CPUs limited to 1
26-Mar-2026 10:33:07.889 INFO [ApusicConsoleHandler] com.apusic.juli.logging.LogPrintStream.println
26-Mar-2026 10:33:08.182 INFO [ApusicConsoleHandler] com.apusic.connector.AbstractProtocol.init Initializing ProtocolHandler ["http-nio-6888"]
26-Mar-2026 10:33:08.205 INFO [ApusicConsoleHandler] com.apusic.ams.startup.Apusic.load Server initialization in [751] milliseconds
26-Mar-2026 10:33:08.245 INFO [ApusicConsoleHandler] com.apusic.ams.core.StandardService.startInternal Starting service [ams-server]
26-Mar-2026 10:33:08.249 INFO [ApusicConsoleHandler] com.apusic.ams.core.StandardEngine.startInternal Starting Servlet engine: [Apusic AAS/10.0.11]
26-Mar-2026 10:33:08.271 INFO [ApusicConsoleHandler] com.apusic.ams.startup.HostConfig.deployWAR Deploying web application archive [/opt/ams11/0325/AAMS-V10.0.11/webapps/examples.war]
```

2、进入安装目录 `${APUSIC_HOME}/bin` ，执行 `startapusic.sh` 。

```
[root@linux-2-40 bin]# ./startapusic.sh
Using APUSIC_BASE: /opt/ams11/0325/AAMS-V10.0.11
Using APUSIC_HOME: /opt/ams11/0325/AAMS-V10.0.11
Using APUSIC_TMPDIR: /opt/ams11/0325/AAMS-V10.0.11/temp
Using JRE_HOME: /opt/java/jdk-17.0.2
Using CLASSPATH: /opt/ams11/0325/AAMS-V10.0.11/bin/bootstrap.jar:/opt/ams11/0325/AAMS-V10.0.11/bin/aas-juli.jar
Using APUSIC_OPTS: -Djava.security.egd=file:/dev/urandom
Using APUSIC_OPTS: -Djava.security.egd=file:/dev/urandom
AAS started.
```

3、如果需要使用调试模式 (debug) , 可在启动脚本, `apusic.sh` 中设置 `JPDA_OPTS` , 如:

```
JPDA_OPTS="--
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000"
```

执行 `apusic.sh jpda start` 。

属性说明:

配置参数	说明	默认值
<code>-agentlib:jdwp</code>	启用 JPDA 调试协议, JVM内置调试库, 必须添加	
<code>transport=dt_socket</code>	调试传输方式: Socket 网络传输, 用于远程调试	
<code>server=y</code>	当前 JVM 作为 调试服务端, 等待 IDE (IDEA/Eclipse) 连接	
<code>suspend=n</code>	为"n"时, 表示启动不暂停, 直接运行。为"y"时, 启动会卡住, 直到 IDE 连接才继续, 用于启动调试启动类	
<code>address=8000</code>	调试端口为8000;如果要指定IP, 写法"address=192.120.1.1:8000"	

### 6.3.2 在 Windows环境下

1、将应用程序打包文件, 如examples.war, 放置 `${APUSIC_HOME}\webapps\` 目录下; 进入安装目录 `${APUSIC_HOME}\bin` , 执行 `apusic.bat run` 。

2、进入安装目录 `${APUSIC_HOME}\bin` , 执行 `startapusic.bat` 。

3、如果需要使用调试模式 (debug) , 可在启动脚本, `apusic.bat` 中设置 `JPDA_OPTS` , 如:

```
JPDA_OPTS=-
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8000
```

执行 `apusic.bat jpda start` 。

如果 Windows 启动出现乱码，则可通过修改 `conf/logging.properties` 文件，把控制台日志编码设置为 GBK。

```

7 # Handler specific properties.
8 # Describes specific configuration info for Handlers.
9 #####
10
11 1apusic.com.apusic.juli.AsyncFileHandler.level = FINE
12 1apusic.com.apusic.juli.AsyncFileHandler.directory = ${apusic.base}/logs
13 1apusic.com.apusic.juli.AsyncFileHandler.prefix = apusic.
14 1apusic.com.apusic.juli.AsyncFileHandler.maxDays = 90
15 1apusic.com.apusic.juli.AsyncFileHandler.encoding = UTF-8
16
17 2localhost.com.apusic.juli.AsyncFileHandler.level = FINE
18 2localhost.com.apusic.juli.AsyncFileHandler.directory = ${apusic.base}/logs
19 2localhost.com.apusic.juli.AsyncFileHandler.prefix = localhost.
20 2localhost.com.apusic.juli.AsyncFileHandler.maxDays = 90
21 2localhost.com.apusic.juli.AsyncFileHandler.encoding = UTF-8
22
23 3manager.com.apusic.juli.AsyncFileHandler.level = FINE
24 3manager.com.apusic.juli.AsyncFileHandler.directory = ${apusic.base}/logs
25 3manager.com.apusic.juli.AsyncFileHandler.prefix = manager.
26 3manager.com.apusic.juli.AsyncFileHandler.maxDays = 90
27 3manager.com.apusic.juli.AsyncFileHandler.encoding = UTF-8
28
29 4host-manager.com.apusic.juli.AsyncFileHandler.level = FINE
30 4host-manager.com.apusic.juli.AsyncFileHandler.directory = ${apusic.base}/logs
31 4host-manager.com.apusic.juli.AsyncFileHandler.prefix = host-manager.
32 4host-manager.com.apusic.juli.AsyncFileHandler.maxDays = 90
33 4host-manager.com.apusic.juli.AsyncFileHandler.encoding = UTF-8
34
35 java.util.logging.ConsoleHandler.level = FINE
36 java.util.logging.ConsoleHandler.formatter = com.apusic.juli.OneLineFormatter
37 java.util.logging.ConsoleHandler.encoding = GBK
38
39
40 #####
41 # Facility specific properties.
42 # Provides extra control for each logger.
43 #####
44
45 com.apusic.ams.core.ContainerBase.[Apusic].[localhost].level = INFO
46 com.apusic.ams.core.ContainerBase.[Apusic].[localhost].handlers = 2localhost.com.apusic.juli.AsyncFileHandler
47
48 com.apusic.ams.core.ContainerBase.[Apusic].[localhost].[/manager].level = INFO
49 com.apusic.ams.core.ContainerBase.[Apusic].[localhost].[/manager].handlers = 3manager.com.apusic.juli.AsyncFileHandler
50
51 com.apusic.ams.core.ContainerBase.[Apusic].[localhost].[/host-manager].level = INFO
52 com.apusic.ams.core.ContainerBase.[Apusic].[localhost].[/host-manager].handlers = 4host-manager.com.apusic.juli.AsyncFileHandler
53
54 ..

```

启动完成后，在浏览器中输入 `http://<IP>:6888` 进行访问，如果是本机，则输入 `<http://localhost:6888>`，则会出现如下的界面表示成功：



## 6.4 停止

### 6.4.1 在Linux环境下

#### 1、Ctrl + C

前端启动状态下，可以直接Ctrl + C退出，停止AAS。

#### 2、stopapusic.sh

进入安装目录 `${APUSIC_HOME}/bin`，执行 `stopapusic.sh`。

#### 3、kill进程

可以通过kill [进程号] 方式停止。

### 6.4.2 在 Windows环境下

#### 1、Ctrl + C

前端启动状态下，可以直接Ctrl + C退出，停止AAS。

#### 2、stopapusic.bat

进入安装目录 `${APUSIC_HOME}\bin`，执行 `stopapusic.bat`。

#### 3、停止进程

可以打开任务管理器，找到对应进程，停止运行。

## 7 应用管理

### 7.1 部署应用

应用服务器支持部署JakartaEE11及以下版本相关Web规范的应用，根据不同JakartaEE版本将应用文件拷贝至对应的目录。

部署JakartaEE11相关Web规范的应用，只需要把应用拷贝到 `webapps` 目录下即可自动部署。

```
[root@linux-2-40 AAMS-V10.0.11]# cd webapps
[root@linux-2-40 webapps]# pwd
/opt/ams11/0325/AAMS-V10.0.11/webapps
[root@linux-2-40 webapps]# ls
examples  examples.war  ROOT
[root@linux-2-40 webapps]#
```

部署JakartaEE9.1及以下版本相关Web规范的应用，只需要把应用拷贝到 `webapps-javaee` 目录下即可自动部署。

```
[root@linux-2-40 webapps-javaee]# pwd
/opt/ams11/0325/AAMS-V10.0.11/webapps-javaee
[root@linux-2-40 webapps-javaee]# ls
hello.war
[root@linux-2-40 webapps-javaee]#
```

### 7.2 取消部署应用

将应用程序文件从 `webapps` 或 `webapps-javaee` 目录下全部删除即可取消部署应用程序。

如有临时文件，可将 `temp/` 目录下对应的文件删除。

### 7.3 访问应用

应用部署完成后，客户端访问应用即可。如浏览器访问应用，格式为 `http://[应用服务器地址]:[端口]/[应用上下文路径]`，如 `http://127.0.0.1:6888/examples`。

如若使用SSL访问，需要先配置SSL证书，参考【配置HTTPS】部分。访问格式为 `https://[应用服务器地址]:[端口]/[应用上下文路径]`，如 `https://127.0.0.1:6888/examples`

## 8 JVM管理

### 8.1 设置 JVM 内存

- Windows

在 `${APUSIC_HOME}/bin` 目录下创建 `setenv.bat` 文件, 添加 `MEMORY_JVMOPTS_DOMAIN` 进行设置。

```
set "MEMORY_JVMOPTS_DOMAIN=-Xmx2g -Xms2g -XX:NewRatio=2 -  
XX:MetaspaceSize=256m -XX:MaxMetaspaceSize=256m"
```

- Linux

在 `${APUSIC_HOME}/bin` 目录下创建 `setenv.sh` 文件, 并添加 `**可运行权限**`, 在文件中添加

`MEMORY_JVMOPTS_DOMAIN` 进行设置。

```
MEMORY_JVMOPTS_DOMAIN="-Xmx2g -Xms2g -XX:NewRatio=2 -  
XX:MetaspaceSize=256m -XX:MaxMetaspaceSize=256m"
```

## 9 配置文件说明

### 9.1 apusic.conf 文件

apusic.conf 是金蝶 Apusic 应用服务器的重要配置文件，可通过该文件设置端口、线程池大小、设置部署应用路径、性能调优等等。

默认配置如：

```
<?xml version='1.0' encoding='utf-8'?>
<config>

    <listeners>
        <listener name="aprLifecycleListener"
className="com.apusic.ams.core.AprLifecycleListener" />
        <listener name="jreMemoryLeakPreventionListener"
className="com.apusic.ams.core.JreMemoryLeakPreventionListener"/>
        <listener name="globalResourcesLifecycleListener"
className="com.apusic.ams.mbeans.GlobalResourcesLifecycleListener"/>
        <listener name="threadLocalLeakPreventionListener"
className="com.apusic.ams.core.ThreadLocalLeakPreventionListener"/>
    </listeners>

    <resources>
        <resource name="userDatabase" auth="Container"
type="com.apusic.ams.UserDatabase"
description="User database that can be updated and
saved"
factory="com.apusic.ams.users.MemoryUserDatabaseFactory"
pathname="conf/aas-users.xml" />
    </resources>

    <executors>
        <executor name="http-thread-pool" namePrefix="http-exec-"
maxThreads="200" minSpareThreads="25" maxQueueSize="5000"
```

```
        threadStepReclaim="false" threadStepSize="20"
threadStepTime="10000"/>
</executors>

<endpoints>
  <endpoint name="ams-http"
    port="6888"
    protocol="HTTP/1.1"
    connectionTimeout="3000"
    redirectPort="6443"
    executor="http-thread-pool"
    maxConnections="10000"
    enableLookups="false"
    acceptCount="100"
    maxPostSize="104857600"
    maxHttpHeaderSize="8192"
    compression="on"
    disableUploadTimeout="true"
    compressionMinSize="2048"

compressibleMimeType="text/html,text/plain,text/css,application/javascript,
font-ttf,application/x-font-
otf,image/svg+xml,image/jpeg,image/png,image/gif,audio/mpeg,video/mp4"
    URIEncoding="utf-8"
    processorCache="20000"
    tcpNoDelay="true"
    keepAliveTimeout="15000"
    maxKeepAliveRequests="100"
    longThreadThreshold="60000"
    longThreadCycleTime="3000"
    longThreadRollingInterval="300000"
    enabledLongThread="false"
    sdosBlacklistEnable="false"
    sdosThreshold="3"
    sdosTimeWindowMinutes="30"
    sdosBlacklistTtlMinutes="30">
```

```

    </endpoint>
    <endpoint name="ams-jmx"
      port="6868"
      auth="false"
      protocol="JMXRMI/2.0">
</endpoint>
</endpoints>

<services>
  <service name="ams-service" defaultHost="localhost">
    <realm className="com.apusic.ams.realm.LockOutRealm">
      <realm resourceName="userDatabase"
className="com.apusic.ams.realm.UserDatabaseRealm"/>
    </realm>
    <host name="localhost" appBase="webapps"
unpackWARs="true" autoDeploy="true" appBaseReDeploy="true"
deployOnStartup="true">
      <valve
className="com.apusic.ams.valves.ErrorReportValve"
showReport="false" showServerInfo="false" />
    </host>
  </service>
</services>

<servers>
  <server name="ams-server" services="ams-service"
endpoints="ams-http" executors="http-thread-pool"/>
</servers>

<containers>
  <container name="container" port="-1" shutdown="shutdown"
servers="ams-server" resources="userDatabase"
listeners="aprLifecycleListener,jreMemoryLeakPreventionListener,globalF
  </container>
</containers>
</config>

```

### 9.1.1 listeners 配置 (监听器)

应用服务器的 `listener` (监听器) 是用于监听Web应用中某些对象的状态变化的组件, 当监听范围的对象状态发生变化时, 服务器会自动调用监听器对象中的方法作出相应的响应处理。

表6: 金蝶Apusic应用服务器V10.0.11标准版listeners配置说明

配置参数	说明	类名
<code>aprLifecycleListener</code>	使用 APR 进行接收客户端请求, 在应用服务器初始化之前、初始化之前尝试初始化 APR 库, 成功则使用 APR 接受处理客户端请求, 应用服务器销毁之后, 该监听器会做 APR 的清理工作	<code>com.apusic.ams.core.AprLifecycleListener</code>
<code>jreMemoryLeakPreventionListener</code>	防止 JRE 内存泄露	<code>com.apusic.ams.core.JreMemoryLeakPreventionListener</code>
<code>globalResourcesLifecycleListener</code>	初始化 JNDI 资源的 MBean	<code>com.apusic.ams.mbeans.GlobalResourcesLifecycleListener</code>
<code>threadLocalLeakPreventionListener</code>	防止 ThreadLocal 对象带来的内存泄漏	<code>com.apusic.ams.core.ThreadLocalLeakPreventionListener</code>

### 9.1.2 resources 配置 (用户权限)

应用服务器的用户权限配置主要涉及 `resources` 和用户信息的配置。可在 `apusic.conf` 配置 `resources`, 指定资源的名称、认证方式、类型等属性; 在 `aas-users.xml` 中配置用户信息, 如用户名、密码、角色信息等; 如果应用程序需要使用用户权限, 通过 `java:comp/env` 前缀加上资源名称来引用配置的资源。

userDatabase：用户权限配置。

表7：金蝶Apusic应用服务器V10.0.11标准版resources配置说明

配置参数	说明	值类型	默认值
name	资源名称, 必须唯一。 如"userDatabase"表示配置用户权限	字符	userDatabase
auth	授权范围	字符	Container
type	类型	字符	com.apusic.ams.UserDatabase
description	描述	字符	User database that can be updated and saved
factory	资源工厂	字符	com.apusic.ams.users.MemoryUserDatabaseFactory
pathname	路径名称	字符	conf/aas-users.xml

### 9.1.3 executors 配置 (线程池)

应用服务器的 `executors` 一个用于定义和管理线程池的配置元素。线程池是执行任务的线程集合，它允许Web应用以并发方式处理多个请求，从而提高性能和响应速度。

```
<executor name="http-thread-pool" namePrefix="http-exec-"
maxThreads="200" minSpareThreads="25" maxQueueSize="5000" />
```

表8：金蝶Apusic应用服务器V10.0.11标准版executors配置说明

配置参数	说明	值类型	默认值
name	线程池名称, 供配置文件其他地方引用, 必须唯一	字符	http-thread-pool
namePrefix	线程池里面的线程的前缀	字符	http-exec-

maxThreads	最大线程数	整型	200
minSpareThreads	最小空闲线程数	整型	25
maxIdleTime	当线程数大于 minSpareThreads 时，空闲线程销毁前最大空闲时间，单位毫秒	整型	60000
maxQueueSize	线程池队列最大任务数, 超过后将被拒绝	整型	Integer.MAX_VALUE

#### 9.1.4 endpoints 配置 (网络 IO 和协议)

应用服务器的 endpoints 是连接器组件中的关键部分，主要负责对外实现TCP/IP协议，提供基础的网络I/O服务。

```
<endpoint name="ams-http" port="6888" protocol="HTTP/1.1"
connectionTimeout="20000" redirectPort="6443"
    executor="http-thread-pool" maxConnections="10000"
enableLookups="false" acceptCount="100" maxPostSize="104857600"
    maxHttpRequestSize="8192" compression="on"
disableUploadTimeout="true" compressionMinSize="2048"

compressableMimeType="text/html,text/plain,text/css,application/javascript
font-ttf,application/x-font-otf"
    URIEncoding="utf-8" processorCache="20000" tcpNoDelay="true"
keepAliveTimeout="15000" maxKeepAliveRequests="100">
    <SSLHostConfig truststoreType="JKS"
truststorePassword="changeit"
    truststoreFile="conf/cert/truststore.jks"
certificateVerification="required">
        <Certificate certificateKeystoreFile="keystore.jks"
certificateKeystoreType="JKS"
            certificateKeystorePassword="changeit" />
    </SSLHostConfig>
</endpoint>
```

endpoint 代表一个端口，接听服务器上一特定 TCP 端口，接收请求。

表9：金蝶Apusic应用服务器V10.0.11标准版endpoints配置说明

配置参数	说明	值类型	默认值
port	监听端口	整型	6888
protocol	处理协议, 可选的有 HTTP/1.1、AJP/1.3	字符	HTTP1.1
connectionTimeout	连接超时时间, 读取 HTTP 请求行超时时间。单位毫秒	数字	3000
redirectPort	如果是 https, 重定向端口	整型	6443
executor	使用的线程池	字符	http-thread-pool
maxConnections	最大并发数	整型	10000
enableLookups	是否启用 DNS 查询, 如果设置 true, 则 request.getRemoteHost() 时, 将调用 DNS 查询, 否则只返回 IP 值	布尔值	false
acceptCount	backlog 大小, 达到 maxConnections 后, 最大队列数, 超出后, 请求将被拒绝	整型	100
maxPostSize	FORM URL 方式的最大提交数据。单位字节	数字	104857600
maxHttpHeaderSize	请求或响应的最大 http header 大小。单位字节	数字	8192 (8KB)
compression	是否启用压缩, 可选值为: off, on, force	字符	on
compressionMinSize	压缩文件值, 小于该值不会进行压缩	整型	2048
compressableMimeType	逗号隔开的可压缩的 MIME 类型	字符	text/html,text/plain,text/css,application/javascript,application/json,application/x-font-ttf,application/x-font-otf,image/svg+xml,image/jpeg,image/png,image/gif,audio/mpeg,video/mp4
connectionUploadTimeout	数据上传超时时间。仅当 disableUploadTimeout=false 时生效	数字	

	效。单位毫秒		
disableUploadTimeout	是否禁用上传超时	布尔值	true
URIEncoding	配置 url 中参数编码	字符	UTF-8
processorCache	缓存 Processor 的最大数量	整型	20000
tcpNoDelay	是否禁用了 Nagle 算法, 允许小包的发送	布尔值	true
keepAliveTimeout	keepalive 的超时时间。单位毫秒	数字	15000
maxKeepAliveRequests	保持 keepalive 的连接的个数, 超过就会关闭连接。-1 不限制	整型	100
address	服务器监听地址	IP地址	0.0.0.0 或 ::
maxCookieCount	最大 Cookie 数量	整型	200
relaxedPathChars	允许请求行中的特殊字符	字符	
relaxedQueryChars	允许请求行中参数的特殊字符	字符	
useAsyncIO	是否使用异步 IO API	布尔值	true
useSendfile	是否使用 sendfile 功能。使用 sendfile 后将不压缩响应	布尔值	true
SSLEnabled	是否允许 SSL 通信	布尔值	false
longThreadThreshold	执行时间超过该时间的线程, 判定为“长线程 / 卡死线程”, 单位毫秒	整型	60000
longThreadCycleTime	设置长线程检测间隔, 单位毫秒	整型	3000
longThreadRollingInterval	长线程日志滚动时间间隔, 单位毫秒	整型	300000
enabledLongThread	是否开启长线程检测功能, 值	布尔	false

	为true或false	值	
longThreadLogPath	长线程日志存放目录	字符	\${APUSIC_HOME}/logs/longThread
sdosBlacklistEnable	是否开启慢攻击检测，值为true或false	布尔值	false
sdosThreshold	阈值，超过指定值加入黑名单，如果小于0，则直接加入黑名单	整型	3
sdosTimeWindowMinutes	在30分钟内有多次被判定为慢攻击，并且次数超过sdosThreshold，就会被加入黑名单，如果小于0则直接加入黑名单。单位分钟	整型	30
sdosBlacklistTtlMinutes	黑名单过期时间，如果小于0，则不过期	整型	30

#### 9.1.4.1 SSLHostConfig 元素

当 Endpoint 允许 SSL 通信时，需要添加 `SSLHostConfig` 子元素，设置 SSL 通信相关的证书等属性。

表10：金蝶Apusic应用服务器V10.0.11标准版 SSLHostConfig配置说明

配置参数	说明	值类型	默认值
protocols	允许与客户端通信的协议名字。可以是以下的组合:SSLv2Hello/SSLv3/TLSv1/TLSv1.1/TLSv1.2/TLSv1.3/SMv1.1/all	字符	all
sslProtocol	服务器使用的 SSL 协议，用于创建 SSLContext	字符	TLS
truststoreFile	信任库文件，用于验证客户端证书	字符	
truststoreType	信任库类型	字符	JKS
truststorePassword	信任库密码	字符	
certificateVerification	证书验证，如果设置为： required: 客户端必须返回有效证书。 optional: 希望客户端返回证书，若客户端不发证书，也不报错。 none 不需要客户端证书，除非客户端请求 "CLIENT-CERT" 限制的资源	字符	none

ciphers	允许使用的密码套件	字符	HIGH:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!kRSA
---------	-----------	----	---

### 9.1.4.2 Certificate 元素

服务器端证书设置元素。

表11: 金蝶Apusic应用服务器V10.0.11标准版 Certificate配置说明

配置参数	说明	值类型	默认值
certificateFile	PEM 编码的证书文件	字符	
certificateChainFile	PEM 编码的证书链文件	字符	
certificateKeyFile	PEM 编码的密钥文件	字符	
certificateKeyPassword	密钥文件密码	字符	
certificateKeyAlias	服务器使用的密码和证书在密钥库的别名	字符	
certificateKeystoreFile	密钥库文件	字符	
certificateKeystorePassword	密钥库文件密码	字符	
certificateKeystoreType	密钥库文件类型	字符	JKS
type	证书类型，用于确认密码套件和证书是匹配的，必须是为 UNDEFINED,RSA,DSS 和 EC 之一。如果 SSLHostConfig 只有一个 Certificate 元素,type 属性不是必须的。若有多个 Certificate 元素，则必须指定 type 属性	字符	UNDEFINED

### 9.1.5 services 配置 (虚拟主机和应用)

应用服务器的 `service` 组件是 `server` 组件的子元素，代表一组共享Engine的Connector，负责管理它们的配置和生命周期，并接收和处理客户端请求。

```
<service name="ams-service" defaultHost="localhost">
  <realm className="com.apusic.ams.realm.LockOutRealm">
    <realm resourceName="userDatabase"
className="com.apusic.ams.realm.UserDatabaseRealm" />
  </realm>
</service>
```

```

</realm>
<host name="localhost" appBase="webapps" unpackWARs="true"
autoDeploy="true">
  <application path="/test-webapp"
docBase="D:\\work_test\\test-webapp" useHttpOnly="true" />
</host>
</service>

```

表12: 金蝶Apusic应用服务器V10.0.11标准版 services配置说明

配置参数	说明	值类型	默认值
name	服务名称, 必须全局唯一	字符	ams-service
defaultHost	默认虚拟主机名称	字符	localhost
jvmRoute	在负载均衡使用场景中, Session 粘滞时使用的标签, 在集群节点中必须唯一。此标签将后缀在 Session ID 中, 此后同一会话的请求将被同一 AAS 实例处理。	字符	

### 9.1.5.1 Realm

`Realm` 元素代表这一个存储了用户名, 密码及角色的库, 用于认证及授权处理。

表13: 金蝶Apusic应用服务器V10.0.11标准版realm配置说明

配置参数	说明	值类型	默认值
className	Java 类名, 此类必须实现 <code>com.apusic.ams.Realm</code>	字符	

AAS 提供的默认实现有:

1.DataSource Database Realm: `com.apusic.ams.realm.DataSourceRealm`

DataSource Database Realm 通过 JNDI 命名的 JDBC 数据源连接到关系型数据库中, 获取用户名, 密码及对应的角色。

```

<Realm className="com.apusic.ams.realm.DataSourceRealm"
dataSourceName="jdbc/authority"

```

```
userTable="users" userNameCol="user_name" userCredCol="user_pass"
userRoleTable="user_roles" roleNameCol="role_name"/>
```

表14: 金蝶Apusic应用服务器V10.0.11标准版DataSource Database Realm配置说明

配置参数	说明	值类型	默认值
dataSourceName	JDBC 数据源的 JNDI 名字	字符	
userTable	用户表名, 表中必须有 userNameCol 和 userCredCol 属性指定的列	字符	
userNameCol	用户名列	字符	
userCredCol	用户凭据列 (例: 密码)	字符	
userRoleTable	用户角色表, 表中必须有 userNameCol 和 roleNameCol 属性指定的列	字符	
roleNameCol	角色列	字符	

2.UserDatabase Realm: `com.apusic.ams.realm.UserDatabaseRealm`

UserDatabase Realm 是可通过 JNDI 获取的实现了 UserDatabase 的认证域。

表15: 金蝶Apusic应用服务器V10.0.11标准版UserDatabase Realm配置说明

配置参数	说明	值类型	默认值
resourceName	UserDatabase 资源的名字	字符	

\*\*3.Combined Realm\*\*: `com.apusic.ams.realm.CombinedRealm``

Combined Realm 组合域, 可以合并多个域, 通过一个或多个子域认证用户。

Combined Realm 子元素定义了它的子域, 按子域定义的顺序认证用户, 用户的角色将有第一个认证通过的域指定。

4.LockOut Realm: `com.apusic.ams.realm.LockOutRealm`

`LockOut Realm` 继承了组合域, 并提供了锁定机制 - 当在某一时段内, 用户验证失败多次后, 将用户锁定。

表16: 金蝶Apusic应用服务器V10.0.11标准版LockOut Realm配置说明

配置参数	说明	值类型	默认值
cacheSize	缓存验证失败用户数	整型	1000

failureCount	锁定前验证失败次数	整型	5
lockOutTime	用户锁死时间，在锁死时间内，验证失败将重置计时器，验证成功不起效。单位为秒	时间	300

### 9.1.5.2 host

`host` 元素代表一个虚拟主机，虚拟主机可以指定网络主机名（例如：[www.mysite.com](http://www.mysite.com)），[客户端可通过主机名访问服务器](#)。多个 `host` 元素可以嵌套在 `service` 元素中，必须有唯一一个 `host` 的 `name` 属性与 `service` 的 `defaultHost` 属性一致。

表17：金蝶Apusic应用服务器V10.0.11标准版host配置说明

配置参数	说明	值类型	默认值
name	虚拟名称，必须全局唯一，通常为在 DNS 注册的网络名	字符	
appBase	应用部署基路径	字符	webapps
unpackWARs	是否解压 war 进行部署	布尔值	true
autoDeploy	是否自动部署，如果为 true，AAS 将周期性的检查 appBase 目录，部署新发现的应用，重新部署更新过的应用	布尔值	true

### 9.1.5.3 application

应用服务器的 `application` 元素代表这一个应用，每个应用被打包成一个 Web Application Archive(WAR) 包或者解压缩的一个目录。`host` 元素下可以添加多个 `application` 元素，每个 `application` 的名字在 `host` 下必须唯一。

```
<application path="/test-webapp"
docBase="D:\\work_test\\test-webapp" useHttpOnly="true" />
```

表18：金蝶Apusic应用服务器V10.0.11标准版application配置说明

配置参数	说明	值类型	默认值
path	应用上下文，访问路径。同一虚拟主机里，必须唯一。如果 docBase 在 host 的 appBase 内，则此属性必须不能设置，不然同一应用可能会被部署两次。	字符	

docBase	应用的根目录（应用以目录形式打包），或者应用（WAR 包）的路径，可以是绝对路径或者相对与 host 元素的 appBase 属性的相对路径	字符	
useHttpOnly	设置为 true 表示防止客户端脚本读取 JSESSIONID，防止 CSRF/XSS 恶意攻击	布尔值	true
cookie	是否允许 cookies 传递 Session ID	布尔值	true
crossContext	若设置为 true, 则 ServletContext.getContext() 能获取同一虚拟主机下其他应用的 ServletContext. 若设置为 false, 则 ServletContext.getContext() 返回 null.	布尔值	false
logEffectiveWebXml	若为 true, 服务器在应用启动时输出生效的 web.xml 日志（INFO）等级，生效的 web.xml 包括默认 web.xml，应用提供的 web.xml、web-fragment.xml 和 annotations 解析结果	布尔值	false
override	若设置为 true，则忽略所有全局和 host 的默认 context 的设置	布尔值	true
reloadable	如果设置为 true，则监控 /WEB-INF/classes/ 和 /WEB-INF/lib/, 若文件有变化，则重新部署应用	布尔值	true
sessionCookieDomain	设置此应用中 Session cookies 的 domain 值	字符	
sessionCookieName	设置 Session cookies 的名字	字符	JSESSIONID
swallowOutput	若设置为 true，应用中输出到 System.out 和 System.err 的内容将被重定向到应用的 logger 中	布尔值	false

### 9.1.6 servers 配置 (引用其他功能)

应用服务器的 `server` 组件负责定义全局服务器配置、管理服务生命周期以及监听服务器关闭命令

```
<servers>
  <server name="ams-server" services="ams-service" endpoints="ams-
http" executors="http-thread-pool" />
</servers>
```

表19: 金蝶Apusic应用服务器V10.0.11标准版servers配置说明

配置参数	说明	值类型	默认值
name	定义的server名称, 全局唯一	字符	
services	引用的服务, 仅可添加一个服务	字符	
endpoints	引用的网络和协议配置, 多个用逗号隔开	字符	
executors	引用的线程池, 多个用逗号隔开, 注意: 该 server 里面的 endpoints 里面引用的线程池也必须在这里引用, 如上面的 http-thread-pool	字符	

### 9.1.7 containers 配置 (监听器等)

应用服务器的 `containers` 用于接收客户端请求并返回响应。

```
<container name="container" port="-1" shutdown="shutdown"
servers="ams-server" resources="userDatabase"
listeners="versionLoggerListener,aprLifecycleListener,jreMemoryLeakPreventor" />
```

表20: 金蝶Apusic应用服务器V10.0.11标准版containers配置说明

配置参数	说明	值类型	默认值
name	container名称, 全局唯一	字符	
port	侦听端口, 用于关闭应用服务器, 不使用可设置为 -1, 建议不使用	整型	
servers	该应用服务器实例引用的服务器, 多个可用逗号隔开, 建议使用单个	字符	
resources	该应用服务器实例引用的资源, 比如用户名密码等, 多个可用逗号隔开	字符	
listeners	该应用服务器实例引用的监听器, 多个可用逗号隔开。引用的是文件开头配置的监听器	字符	

## 9.2 其他配置

## 9.2.1 共享库配置

共享库可以在多个应用中共享相同的类库，降低内存的使用。配置的步骤如下：

1. 可以在 `${APUSIC_HOME}/` 目录下创建一个目录，如 `sharelib`；
2. 目录中放置需要在多个应用中共享的 jar 包；
3. 配置 `${APUSIC_HOME}/conf` 下的 `apusic.properties`，修改 `shared.loader` 的配置为：

```
shared.loader="${apusic.home}/sharelib/*.jar"
```

经过上面的步骤即可完成共享库配置，应用则可以直接使用该 jar 的资源，共享库的类库优先级低于应用中相同的类库优先级。

## 9.2.2 手动部署多个应用

在普通方式部署时，应用除了直接拷贝到 `webapps` 目录自动进行部署外，也可以通过修改配置文件手动部署：修改 `apusic.conf` 文件中的 `host` 标签，增加如下的类似配置，其中部署了 2 个 `application`，属性 `path` 为指定的上下文路径，`docBase` 是应用的目录：

```
<host name="localhost" appBase="webapps" unpackWARs="true"
autoDeploy="true">
  <application path="/test1" docBase="/opt/apps/test1.war"
useHttpOnly="true" />
  <application path="/test2" docBase="/opt/apps/test2.war"
useHttpOnly="true" />
</host>
```

## 9.2.3 端口的修改

修改 `apusic.conf` 文件中 `endpointname="ams-http"` 的 `port` 属性，默认为 `port="6888"`。

## 9.2.4 开启 JMX

JMX 服务默认为不开启，如果需要开启 JMX 服务，首先需要检查配置文件中是否存在 `ams-jmx` 服务配置，如下：

```
<endpoint name="ams-jmx" port="6868" auth="false"
protocol="JMXRMI/2.0">
</endpoint>
```

如果没有则需要添加。再需要在 `apusic.conf` 中找到如下的配置，把 `ams-jmx` 增加到 `endpoints` 属性中，如下配置：

```
<server name="ams-server" services="ams-service" endpoints="ams-
http,ams-jmx" executors="http-thread-pool" />
```

JMX 默认访问不进行授权验证，如果需要授权认证，则需要修改相关配置并制定授权文件：

修改 `ams-jmx` 服务属性 `auth` 设置为 `true`，配置密码和访问授权文件属性 `passFile` 和 `accessFile`，默认为 `conf` 目录下的 `jmxremote.password` 和 `jmxremote.access` 文件

`jmxremote.password` 文件内容类似：

```
controlRole changeit
```

其中 `controlRole` 为用户名，`changeit` 为密码

`jmxremote.access` 文件内容类似：

```
controlRole readonly

controlRole readwrite \
    create
javax.management.monitor.* , javax.management.timer.* \
    unregister
```

## 10 配置数据源

应用服务器提供配置数据源功能。

1.数据库连接池配置（在 `context.xml` 文件中新增内容，如下图）：

```
<Resource name="jdbc/test" auth="Container"
type="javax.sql.DataSource" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://127.0.0.1:3306/test_platform?
useUnicode=true&characterEncoding=utf-8" username="root"
password="root" maxTotal="100" maxIdle="30"
maxWaitMillis="10000" />
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- The contents of this file will be loaded for each web application -->
3 <Context>
4
5 <!-- Default set of monitored resources. If one of these changes, the -->
6 <!-- web application will be reloaded. -->
7 <WatchedResource>WEB-INF/web.xml</WatchedResource>
8 <WatchedResource>WEB-INF/aas-web.xml</WatchedResource>
9 <WatchedResource>${apusic.base}/conf/web.xml</WatchedResource>
10
11 <!-- Uncomment this to disable session persistence across Aas restarts -->
12 <!--
13 <Manager pathname="" />
14 -->
15
16 <Resource name="jdbc/test"
17 auth="Container"
18 type="javax.sql.DataSource"
19 driverClassName="com.mysql.jdbc.Driver"
20 url="jdbc:mysql://127.0.0.1:3306/test_platform?useUnicode=true&characterEncoding=utf-8"
21 username="root"
22 password="root"
23 maxTotal="100"
24 maxIdle="30"
25 maxWaitMillis="10000"/>
26
27 </Context>
28
```

表21: 金蝶Apusic应用服务器V10.0.11标准版数据源配置说明

配置参数	说明	值类型	默认值
name	数据源 jndi 名称	字符	
auth	容器负责资源的连接	字符	
type	绑定类型	字符	

url	数据库连接 url	字符	
username	用户名	字符	
password	密码	字符	
maxTotal	最大连接数	整型	
maxIdle	最大空闲连接数	整型	
maxWaitMillis	初始化连接大小	整型	0
minIdle	最小空闲连接数	整型	
validationQuery	连接有效性检测语句	字符	
validationQueryTimeout	检测语句超时时间, -1表示不超时	字符	-1
testOnBorrow	true 或 false, 从池中拿连接的时, 是否检测连接有效性	布尔值	true
testOnReturn	true 或 false, 还回连接到池里面时, 是否检测连接有效性	布尔值	false
removeAbandonedOnBorrow	true 或 false, 是否移除失效的连接	布尔值	false
logAbandoned	true 或 false, 是否打印移除失效连接	布尔值	false
testWhileIdle	true 或 false, 连接空闲时, 是否检测连接有效性	布尔值	false
minEvictableIdleTimeMillis	连接空闲时间, 超过该时间会从池中移除 (单位为毫秒)	整型	
timeBetweenEvictionRunsMillis	每隔多少时间去检测一次空闲连接是否超时, 默认不检测 (单位为毫秒)	整型	

2.将数据源对应版本的驱动放入到 `${APUSIC_HOME}/lib/` 目录下。

**\*\*注意:\*\***在代码中要引用 jndi 需要加上前缀: `java:comp/env`, 如上面的数据源的:

```
java:comp/env/jdbc/test
```

# 11 配置 HTTPS

## 11.1 单向认证 -- 证书格式为 JKS

注意：配置了新的 `endpoint` 需要在 `server` 中进行引用，配置了线程池也需要在 `server` 中进行引用。

```
<endpoint name="ams-https" port="8443" protocol="HTTP/1.1"
SSLEnabled="true" executor="http-thread-pool">
  <SSLHostConfig>
    <Certificate
certificateKeystoreFile="conf/cert/arsp_server_keystore.jks"
certificateKeystoreType="JKS"
      certificateKeystorePassword="changeit" />
    </SSLHostConfig>
  </endpoint>

<servers>
  <server name="ams-server" services="ams-service"
endpoints="ams-http,ams-https" executors="http-thread-pool"/>
</servers>
```

表22：金蝶Apusic应用服务器V10.0.11标准版证书格式为JKS的单向认证方式配置说明

配置参数	说明	值类型	默认值
SSLEnabled	是否是 ssl 通讯	布尔值	
executor	使用的线程池	字符	
certificateKeystoreType	keystore 格式，JKS 或者 PKCS12	字符	
certificateKeystorePassword	keystore 密码	字符	

certificateKeystoreFile

keystore 文件

字符

## 11.2 双向认证 -- 证书格式为 JKS

注意：配置了新的 `endpoint` 需要在 `server` 中进行引用，配置了线程池也需要在 `server` 中进行引用。

```
<endpoint name="ams-https" port="8443" protocol="HTTP/1.1"
SSLEnabled="true" executor="http-thread-pool">
  <SSLHostConfig truststoreType="JKS"
truststorePassword="changeit"
      truststoreFile="conf/cert/arsp_server_truststore.jks"
certificateVerification="required">
    <Certificate
certificateKeystoreFile="conf/cert/arsp_server_keystore.jks"
certificateKeystoreType="JKS"
      certificateKeystorePassword="changeit" />
  </SSLHostConfig>
</endpoint>

<servers>
  <server name="ams-server" services="ams-service"
endpoints="ams-http,ams-https" executors="http-thread-pool"/>
</servers>
```

表23：金蝶Apusic应用服务器V10.0.11标准版证书格式为JKS的双向认证方式配置说明

配置参数	说明	值类型	默认值
SSLEnabled	是否是 ssl 通讯	布尔值	
executor	使用的线程池	字符	
truststoreType	信任库格式，JKS 或者 PKCS12	字符	

truststorePassword	信任库密码	字符	
truststoreFile	信任库文件	字符	
certificateVerification	是否需要验证客户端证书，即是否双向认证，默认不验证	字符	
certificateKeystoreType	keystore 格式，JKS 或者 PKCS12	字符	
certificateKeystorePassword	keystore 密码	字符	
certificateKeystoreFile	keystore 文件	字符	

### 11.3 单向认证 -- 证书格式为 PEM

注意：配置了新的 `endpoint` 需要在 `server` 中进行引用，配置了线程池也需要在 `server` 中进行引用。

```
<endpoint name="ams-https" port="8443" protocol="HTTP/1.1"
SSL-enabled="true" executor="http-thread-pool">
  <SSLHostConfig>
    <Certificate certificateKeyFile="conf/cert/private_key.pem"
certificateKeyPassword="changeit"
certificateFile="conf/cert/public_key.pem"
certificateChainFile="conf/cert/chain.pem" type="RSA" />
  </SSLHostConfig>
</endpoint>

<servers>
  <server name="ams-server" services="ams-service"
endpoints="ams-http,ams-https" executors="http-thread-pool"/>
</servers>
```

表24：金蝶Apusic应用服务器V10.0.11标准版证书格式为PEM的单向认证方式配置说明

配置参数	说明	值类型	默认值
SSL-enabled	是否开启 ssl 通讯	布尔值	

executor	使用的线程池	字符	
type	证书类型	字符	
certificateKeyFile	私钥文件	字符	
certificateKeyPassword	私钥密码, 可选	字符	
certificateFile	公钥文件	字符	
certificateChainFile	证书链文件, 可选	字符	

## 11.4 双向认证 -- 证书格式为 PEM

```

<endpoint name="ams-https" port="8443" protocol="HTTP/1.1"
SSLEnabled="true" executor="http-thread-pool">
  <UpgradeProtocol
className="com.apusic.connector.http2.Http2Protocol" />
  <SSLHostConfig caCertificateFile="conf/cert/public_ca_key.pem"
certificateVerification="need">
    <Certificate certificateKeyFile="conf/cert/private_key.pem"
certificateKeyPassword="changeit"
    certificateFile="conf/cert/public_key.pem"
certificateChainFile="conf/cert/chain.pem" type="RSA" />
  </SSLHostConfig>
</endpoint>

<servers>
  <server name="ams-server" services="ams-service"
endpoints="ams-http,ams-https" executors="http-thread-pool"/>
</servers>

```

表25: 金蝶Apusic应用服务器V10.0.11标准版证书格式为PEM的双向认证方式配置说明

配置参数	说明	值类型	默认值
SSLEnabled	是否是 ssl 通讯	布尔值	
executor	使用的线程池	字符	
type	证书类型	字符	
certificateKeyFile	私钥文件	字符	
certificateKeyPassword	私钥密码, 可选	字符	
certificateFile	公钥文件	字符	
certificateChainFile	证书链文件, 可选	字符	
caCertificateFile	根证书文件, 可选	字符	
certificateVerification	是否需要验证客户端证书, 即是否双向认证	字符	

## 12 国密配置说明

### 12.1 环境说明

需要使用 JDK17 版本和国密浏览器（如 360 国密浏览器）。

### 12.2 应用服务器证书配置

#### 12.2.1 单向认证

修改 `apusic.conf` 文件，增加如下的 `endpoint` 配置（如果已经存在，则进行修改）：

```
<endpoint name="https" port="6887" maxThreads="150"
SSLEnabled="true">
  <SSLHostConfig sslProtocol="SMv1.1" protocols="SMv1.1">
    <Certificate certificateKeystoreFile="conf/keystore.pfx"
certificateKeystorePassword="changeit"
      certificateKeyPassword="changeit" type="EC"
certificateKeystoreType="PKCS12" />
  </SSLHostConfig>
</endpoint>
```

表26：金蝶Apusic应用服务器V10.0.11标准版国密证书的单向认证方式配置说明

配置参数	说明	值类型	默认值
SSLEnabled	是否是 ssl 通讯	布尔值	
certificateKeystoreFile	表示证书文件的路径	字符	
protocols	需要指定为 SMv1.1	字符	
sslProtocol	需要指定为 SMv1.1	字符	
certificateKeystorePassword	为证书库文件的密码	字符	
certificateKeyPassword	为证书的密码	字符	
type	证书类型	字符	

certificateKeystoreType	keystore类型, 如PKCS12	字符	
-------------------------	---------------------	----	--

### 12.2.2 双向认证

修改 `apusic.conf` 文件, 增加如下的 `endpoint` 配置 (如果已经存在, 则进行修改):

```
<endpoint name="https" port="6887" maxThreads="150"
SSLEnabled="true">
  <SSLHostConfig sslProtocol="SMv1.1" protocols="SMv1.1"
truststoreFile="conf/truststore.jks"
  truststorePassword="changeit" certificateVerification="want"
truststoreType="JKS">
    <Certificate certificateKeystoreFile="conf/keystore.p12"
certificateKeystorePassword="changeit"
  certificateKeyPassword="changeit" type="EC"
certificateKeystoreType="PKCS12" />
  </SSLHostConfig>
</endpoint>
```

表27: 金蝶Apusic应用服务器V10.0.11标准版国密证书的双向认证方式配置说明

配置参数	说明	值类型	默认值
SSLEnabled	是否是 ssl 通讯	布尔值	
certificateKeystoreFile	表示证书文件的路径	字符	
protocols	需要指定为 SMv1.1	字符	
sslProtocol	需要指定为 SMv1.1	字符	
certificateKeystorePassword	为证书库文件的密码	字符	
certificateKeyPassword	为证书的密码	字符	
type	证书类型	字符	
certificateKeystoreType	keystore类型, 如PKCS12	字符	
truststoreFile	指定信任库文件, 其类型是JKS	字符	

truststorePassword	信任库密码	字符	
certificateVerification	是否进行证书验证，双向认证时为 need 值，可选客户端认证设置为 want	字符	

### 12.2.3 增加 endpoint 到 server

修改 `apusic.conf` 文件，修改如下的配置，其中 `endpoints` 属性中的 `https` 就是上面配置单向认证或双向认证的 `endpoint` 的名字：

```
<server name="ams-server" services="ams-service"
endpoints="http,https" executors="http-thread-pool" />
```

## 12.3 启动应用服务器

通过上面的步骤配置完成后，通过命令 `apusic.sh gm start` 或 `apusic.sh gm run` 可以启动国密支持的容器。

备注：可以通过在启动脚本中增加参数 `-Djavax.net.debug=ssl:handshake` 来打印具体的 SSL 处理过程，方便分析可能存在的问题。

## 13 日志配置

### 13.1 系统日志配置

通过配置系统日志文件配置日志信息，日志文件为 `logging.properties`。

内部采用 JULI (Apache Commons Logging 的定制分支) 并强制绑定 `java.util.logging`，以确保服务器与 Web 应用的日志系统相互隔离；若需更换日志框架，则需配置重定向并确保新框架支持多类加载器环境。

```
handlers = 1apusic.com.apusic.juli.AsyncFileHandler, 2localhost.com.apusic.juli.AsyncFileHandler, com.apusic.juli.logging.ApusicConsoleHandler

.handlers = 1apusic.com.apusic.juli.AsyncFileHandler, com.apusic.juli.logging.ApusicConsoleHandler

#####
# Handler specific properties.
# Describes specific configuration info for Handlers.
#####

1apusic.com.apusic.juli.AsyncFileHandler.level = FINE
1apusic.com.apusic.juli.AsyncFileHandler.directory = ${apusic.base}/logs
1apusic.com.apusic.juli.AsyncFileHandler.prefix = apusic.
1apusic.com.apusic.juli.AsyncFileHandler.maxDays = 90
1apusic.com.apusic.juli.AsyncFileHandler.encoding = UTF-8

2localhost.com.apusic.juli.AsyncFileHandler.level = FINE
2localhost.com.apusic.juli.AsyncFileHandler.directory = ${apusic.base}/logs
2localhost.com.apusic.juli.AsyncFileHandler.prefix = localhost.
2localhost.com.apusic.juli.AsyncFileHandler.maxDays = 90
2localhost.com.apusic.juli.AsyncFileHandler.encoding = UTF-8

com.apusic.juli.logging.ApusicConsoleHandler.level = FINE
com.apusic.juli.logging.ApusicConsoleHandler.formatter = com.apusic.juli.OneLineFormatter
com.apusic.juli.logging.ApusicConsoleHandler.encoding = UTF-8

#####
# Facility specific properties.
# Provides extra control for each logger.
#####

com.apusic.ams.core.ContainerBase.[ams-service].[localhost].level = INFO
com.apusic.ams.core.ContainerBase.[ams-service].[localhost].handlers = 2localhost.com.apusic.juli.AsyncFileHandler

# For example, set the com.apusic.ams.util.LifecycleBase logger to log
# each component that extends LifecycleBase changing state:
#com.apusic.ams.util.LifecycleBase.level = FINE
```

表28: 金蝶Apusic应用服务器V10.0.11标准版logging.properties日志配置说明

配置参数	说明	默认值
handlers	根日志处理器列表，所有日志默认会交给这些处理器处理	1apusic.com.apusic.juli.AsyncFileHandler, 2localhost.com.apusic.juli.AsyncFileHandler, com.apusic.juli.logging.ApusicConsoleHandler
.handlers	默认类/包级别的处理器，未单独配置的日志记录器 (Logger) 会继承此配置	1apusic.com.apusic.juli.AsyncFileHandler, com.apusic.juli.logging.ApusicConsoleHandler

表29: 金蝶Apusic应用服务器V10.0.11标准版logging.properties处理程序特定属性

配置参数	说明	默认值
com.apusic.juli.AsyncFileHandler.level	设置该日志处理器的日志级别，可设置为ALL、FINEST、FINER、FINE、INFO、WARNING、SEVERE	ALL
com.apusic.juli.AsyncFileHandler.directory	指定日志文件的存储目录	\${apusic.base}/logs
com.apusic.juli.AsyncFileHandler.prefix	设置日志文件的文件名前缀	apusic.
com.apusic.juli.AsyncFileHandler.maxDays	设置日志文件的最大保留天数	90
com.apusic.juli.AsyncFileHandler.encoding	设置日志文件的字符编码	UTF-8
com.apusic.juli.logging.ApusicConsoleHandler.level	设置控制台输出的日志级别	ALL
com.apusic.juli.logging.ApusicConsoleHandler.formatter	设置控制台输出的日志单行格式	com.apusic.juli.OneLineFormatter
com.apusic.juli.logging.ApusicConsoleHandler.encoding	设置控制台输出的日志编码	UTF-8

表30：金蝶Apusic应用服务器V10.0.11标准版logging.properties特定服务日志配置

配置参数	说明	默认值
com.apusic.ams.core.ContainerBase.[ams-service].[localhost].level	ams-server服务的日志级别 com.apusic.ams.core.ContainerBase:日志记录器的类名，对应AMS服务的容器基础类 [ams-service]:服务名称，限定日志作用于ams-service服务 [localhost]:虚拟主机名称，限定日志作用于localhost虚拟主机 level = INFO: 该服务的日志级别为INFO，只输出INFO及以上级别(WARNING/SEVERE)的日志，过滤掉调试信息	INFO
com.apusic.ams.core.ContainerBase.[ams-service].[localhost].handlers	该服务的日志只输出到2localhost异步文件处理器(对应localhost.log)，不会输	2localhost.com.apusic.juli.AsyncFileHandler

	出到全局 apusic.log 或控制台，实现服务日志隔离	
--	-------------------------------	--

表31: 金蝶Apusic应用服务器V10.0.11标准版logging.properties其他配置

配置参数	说明	默认值
com.apusic.ams.util.LifecycleBase.level	开启后会记录所有继承 LifecycleBase 的组件状态变化（如启动、停止、初始化），级别为 FINE（调试级）	FINE
com.apusic.connector.http2.level	开启后会记录 HTTP/2 协议处理的详细调试信息	FINE
com.apusic.aas.websocket.level	开启后会记录 WebSocket 连接、消息处理的详细调试信息	FINE

## 14 访问日志记录

Valve元素表示将插入到引擎、主机或上下文的请求处理管道中的组件。单个值具有不同的处理能力，下文将分别进行描述。

### 14.1 配置访问日志

访问日志可配置在 `application` 或者 `service` 里面。配置在 `host` 里面则该 `host` 下的应用访问都会被记录日志。

```
<Valve className="com.apusic.ams.valves.AccessLogValve"
directory="logs" prefix="access_log" suffix=".txt"
pattern="common" />
```

```
<services>
  <service name="ams-service" defaultHost="localhost">
    <realm className="com.apusic.ams.realm.LockOutRealm">
      <realm resourceName="userDatabase" className="com.apusic.ams.realm.UserDatabaseRealm"/>
    </realm>
    <host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
      <Valve className="com.apusic.ams.valves.AccessLogValve" directory="logs" prefix="access_log" suffix=".txt" pattern="common" />
      <application path="/examples" docBase="/opt/testz/aams/examples.war" useHttpOnly="true" />
      <Valve className="com.apusic.ams.valves.RemoteAddrValve" allow="172.24.6.160" deny="" />
    </host>
  </service>
</services>
```

表32：金蝶Apusic应用服务器V10.0.11标准版访问日志配置说明

配置参数	说明	值类型	默认值
className	要使用的实现的Java类名，用默认值的访问日志值，则必须将其设置为com.apusic.ams.valves.AccessLogValve	字符	com.apusic.ams.valves.AccessLogValve
buffered	确定日志记录是否将被缓冲。如果设置为false，则每次请求后都会写入访问日志。	字符	true
directory	日志存储目录	字符	
prefix	日志文件前缀	字符	
suffix	日志文件后缀	字符	
fileDateFormat	文件时间格式，一般文件命名为 prefix（前缀）+ 时间格式（一般是按天算）+.+suffix（后	字符	

	缀)，如果想改为每小时一个文件可设置为 <code>fileDateFormat="yyyy-MM-dd.HH"</code>		
pattern	日志格式化布局，用于标识要记录的请求和响应中的各种信息字段，或通用或组合的单词，以选择标准格式。有关配置此属性的更多信息，请参阅下文。	字符	
conditionIf	启用条件日志记录。如果设置，则只有当 <code>ServletRequest.getAttribute()</code> 不为空时，才会记录请求。例如，如果此值设置为 <code>important</code> ，则只有当 <code>ServletRequest.getAttribute("important") != null</code> 时才是有效的。使用过滤器是一种简单的方法，可以在许多不同的请求上设置/取消设置 <code>ServletRequest</code> 中的属性	字符	
conditionUnless	启用条件日志记录。如果设置，则只有当 <code>ServletRequest.getAttribute()</code> 为 <code>null</code> 时，才会记录请求。例如，如果此值设置为 <code>junk</code> ，则只有当 <code>ServletRequest.getAttribute("junk") == null</code> 时，才会记录特定请求。使用过滤器是一种简单的方法，可以在许多不同的请求上设置/取消设置 <code>ServletRequest</code> 中的属性	字符	
encoding	用于写入日志文件的字符集。空字符串表示使用默认字符集	字符	UTF-8
ipv6Canonical	用于确定IPv6地址是否应以RFC 5952定义的规范表示格式表示。如果设置为 <code>true</code> ，则IPv6地址将以规范格式写入（例如， <code>2001:db8::1:0:0:1, ::1</code> ），否则将以完整形式表示（例如， <code>2001:db8:0:0:1:0:0:1, 0:0:0:0:0:0:0:1</code> ）	字符	false
locale	用于格式化访问日志行中时间戳的区域设置。使用显式 <code>SimpleDateFormat</code> 模式配置的任何时间戳（ <code>%{xxx}t</code> ）在此区域设置中格式化。默认情况下，使用Java进程的默认区域设置。不支持在 <code>AccessLogValve</code> 初始化后切换区域设置。任何使用通用日志格式（CLF）的时间戳都始终在区域设置 <code>en_US</code> 中格式化。	字符	
maxDays	在删除之前，将保留轮换访问日志的最大天数。如果未指定，将使用默认值 <code>-1</code> ，这意味着永远不会删除旧文件。	字符	-1

maxLogMessageBufferSize	日志消息缓冲区通常会被回收和重复使用。为了防止过度使用内存，如果缓冲区增长超过此大小，它将被丢弃。默认值为256个字符。这应该设置为大于典型的访问日志消息大小。	字符	256
renameOnRotate	默认情况下，对于可旋转日志，活动访问日志文件名将包含fileDateFormat中的当前时间戳。在旋转过程中，文件被关闭，并创建和使用名称中具有下一个时间戳的新文件。当将renameOnRotate设置为true时，时间戳不再是活动日志文件名的一部分。只有在旋转过程中，文件才会关闭，然后重命名以包含时间戳。这类似于大多数日志框架在进行基于时间的轮换时的行为	字符	false
requestAttributesEnabled	设置为true可检查是否存在请求属性（通常由RemoteIpValve等设置），这些属性应用于覆盖远程地址、远程主机、服务器端口和协议请求返回的值。如果未设置属性，或者此属性设置为false，则将使用请求中的值。如果未设置，将使用默认值false。	字符	false
rotatable	是否进行日志轮换。如果设置为false，则此文件永远不会旋转，fileDateFormat将被忽略	字符	true

pattern: 参考下面内容:

%a: 远程 IP 地址

%A: 本地 IP 地址

%b: 发送的字节数，不包括 HTTP 头，如果为零则显示为"\_"

%B: 发送的字节数，不包括 HTTP 头

%F: 提交响应所用的时间（毫秒）

%h: 远程主机名（如果连接器的enableLookups为false，则为IP地址）

%H: 请求协议

%l (小写的 L): 来自identd的远程逻辑用户名（始终返回"-")

%m: 请求方法（GET、POST等）

%p: 接收此请求的本地端口

**%q:** 查询字符串（如果存在，则前缀为“?”）

**%r:** 请求的第一行（方法和请求URI）

**%s:** 响应的 HTTP 状态代码

**%S:** 用户会话 ID

**%t:** 日期和时间，通用日志格式

**%T:** 处理请求所用的时间，以秒为单位

**%u:** 已通过身份验证的远程用户（如果有的话），否则为“-”（如果需要，则转义）

**%U:** 请求的 URL 路径

**%v:** 本地服务器名

**%D:** 处理请求所用的时间（微秒）

**%T:** 处理请求所用的时间（秒）

**%I(大写的 i):** 当前请求线程名称（稍后可以与堆栈跟踪进行比较）

**%X:** 响应完成时的连接状态：

- X= 响应完成前连接已中止。
- += 发送响应后，连接可能会保持活动状态。
- -= 发送响应后，连接将关闭

还支持写入传入或传出头部、cookie、会话或请求属性以及特殊时间戳格式的信息。类似Apache语法，以下每个占位符都可以与不同的 `xxx` 键一起使用多次：

- `%{xxx}a`：写入远程地址（客户端）（`xxx=remote`）或连接对等方地址（`xxx=peer`）。
- `%{xxx}i`：写入名称为xxx的传入头部的值（如果需要，则进行转义）。
- `%{xxx}o`：写入名称为xxx的传出头部的值（如果需要，则进行转义）。
- `%{xxx}c`：写入名称为xxx的cookie的值（用逗号分隔，如果需要，则进行转义）。
- `%{xxx}r`：写入名称为xxx的ServletRequest属性的值（如果需要，则进行转义；如果请求为null，则值为??）。
- `%{xxx}s`：写入名称为xxx的HttpSession属性的值（如果需要，则进行转义；如果请求为null，则值为??）。
- `%{xxx}p`：写入本地（服务器）端口（`xxx=local`）或远程（客户端）端口（`xxx=remote`）。
- `%{xxx}t`：使用增强的 `SimpleDateFormat` 模式xxx格式化请求结束时的时间戳。

- `%{xxx}T` : 使用单位xxx写入处理请求所花费的时间, 其中有效单位为ns (纳秒)、us (微秒)、ms (毫秒)、fracsec (秒的小数部分) 或s (整秒)。`%{s}T` 与 `%T` 等效, `%{us}T` 与 `%D` 等效。

在 `%{xxx}t` 中, 允许使用 `SimpleDateFormat` 支持的所有格式。此外, 还添加了以下扩展:

- `sec` : 自纪元 (epoch) 以来的秒数。
- `msec` : 自纪元 (epoch) 以来的毫秒数。
- `msec_frac` : 毫秒的小数部分。

这些格式不能与 `SimpleDateFormat` 格式在同一个格式标记中混合使用。此外, 还可以定义是记录请求开始时间的戳还是响应结束时间的戳:

- `begin` 或前缀 `begin:` : 选择请求开始时间。
- `end` 或前缀 `end:` : 选择响应结束时间。

通过在模式中添加多个 `%{xxx}t` 标记, 也可以同时记录这两个时间戳。

转义规则如下: 双引号 (") 被转义为", 反斜杠 (\) 被转义为\。对于\f、\n、\r和\t, 使用标准C转义规则。任何其他控制字符或代码点高于127的字符都使用标准Java Unicode转义 (\uXXXX) 进行编码。

简写模式 `pattern="common"` 对应由 `'%h %l %u %t "%r" %s %b'` 定义的通用日志格式 (Common Log Format)。

简写模式 `pattern="combined"` 在 `common` 模式的基础上, 将 `Referer` 和用户代理头 (User-Agent headers) 的值 (每个值都用双引号括起来) 追加到日志中, 其格式为 `"%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-Agent}i\""`。

使用未知模式标识符的字段将被记录为 `???X???`, 其中X是未知标识符。对于包含未知模式标识符和 `{xxx}` 键的字段, 将被记录为 `???`。

## 14.2 访问控制

### 14.2.1 远程地址

限制远程地址访问, 即允许将提交此请求的客户端的IP地址与一个或多个正则表达式进行比较, 并根据比较结果允许请求继续或拒绝处理来自此客户端的请求。

将 `addConnectorPort` 属性设置为true后, 可以在服务器连接器端口后附加一个分号 (;), 以便为每个连接器使用不同的表达式。

将 `usePeerAddress` 属性设置为true后, 该Valve将在其检查中使用连接对等地址。如果在AAS前面结合AJP协议或HTTP协议加 `RemoteIp (Valve | Filter)` 使用反向代理, 则这将与客户端IP不同。

被拒绝的请求将收到状态码为403的响应。可以使用 `denyStatus` 属性来覆盖此状态码。

将 `invalidAuthenticationWhenDeny` 属性设置为`true`后，可以改变被拒绝请求的行为，不是拒绝请求，而是设置一个无效的身份验证头部。这与上下文属性 `preemptiveAuthentication="true"` 结合使用时非常有用。

注意：在使用IPv6地址时，有一个警告。该阀正在处理的IP地址的格式取决于用于获取它的API。如果地址是使用 `Inet6Address`类从Java套接字获得的，那么它的格式将是 `x:x:x:x:x:x:x:x`。也就是说，localhost的IP地址将是 `0:0:0:0:0:0:0:1`，而不是更广泛使用的 `::1`。实际值请参考访问日志。

远程地址的属性包括：

表33：金蝶Apusic应用服务器V10.0.11标准版远程地址的属性

配置参数	说明	值类型	默认值
<code>classname</code>	要使用的实现的Java类名。必须设置为 <code>com.apusic.ams.valves.RemoteAddrValve</code>	字符	<code>com.apusic.ams.valves.RemoteAddrValve</code>
<code>allow</code>	允许远程访问的IP地址，如果指定了此属性，则远程地址必须匹配才能接受此请求。如果未指定此属性，则除非远程地址与拒绝模式匹配，否则将接受所有请求。	字符	
<code>deny</code>	禁止远程访问的IP地址。如果指定了此属性，则远程地址必须不匹配才能接受此请求。如果未指定此属性，则请求接受仅由 <code>allow</code> 属性决定。	字符	
<code>denyStatus</code>	拒绝请求时使用的HTTP响应状态码	字符	403
<code>addConnectorPort</code>	将服务器连接器端口附加到客户端IP地址后，并使用分号（";"）分隔。如果设置为 <code>true</code> ，则使用 <code>allow</code> 和 <code>deny</code> 配置的表达式将与 <code>ADDRESS;PORT</code> 进行比较。其中 <code>ADDRESS</code> 是客户端IP地址， <code>PORT</code> 是接收请求的AAS连接器端口。	字符	<code>false</code>
<code>invalidAuthenticationWhenDeny</code>	当应拒绝请求时，不拒绝请求而是设置无效的身份验证头。这仅当上下文具有属性 <code>preemptiveAuthentication="true"</code> 设置时才有效。不会覆盖已存在的身份验证头。实际上，这将触发身份验证而不是拒绝，即使应用程序没有配置安全约束也是如此。这可以与 <code>addConnectorPort</code> 结合使用，以便根据客户端和用于访问应用程序的连接器来触发身份验证	字符	
<code>usePeerAddress</code>	使用连接对等方地址而不是客户端IP地址。如果AAS前面使用了反向代理，并且与AJP协议或HT	字符	

TP协议（加上Remotelp（Valve|Filter））结合使用，那么它们将会有所不同。

配置示例：

只允许 172.24.6.\* 访问。

```
<Valve className="com.apusic.ams.valves.RemoteAddrValve"
allow="172.24.6.*" deny="" />
```

```
<services>
  <service name="ams-service" defaultHost="localhost">
    <realm className="com.apusic.ams.realm.LockOutRealm">
      <realm resourceName="userDatabase" className="com.apusic.ams.realm.UserDatabaseRealm"/>
    </realm>
    <host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
      <Valve className="com.apusic.ams.valves.AccessLogValve" directory="logs" prefix="access_log" suffix=".txt" pattern="common" />
      <application path="/examples" docBase="/opt/testz/aams/examples.war" useHttpOnly="true" />
      <application path="/test2" docBase="/opt/testz/aams/demo/examples1.war" useHttpOnly="true" />
      <Valve className="com.apusic.ams.valves.RemoteAddrValve" allow="172.24.6.*" deny="" />
    </host>
  </service>
</services>
```

## 14.2.2 远程主机名

限制远程主机名访问，允许将提交此请求的客户端的主机名与一个或多个正则表达式进行比较，并根据比较结果允许请求继续或拒绝处理来自此客户端的请求。

在设置属性 `addConnectorPort` 为true后，可以在正则表达式后附加服务器连接器端口（使用分号“;”分隔），以便为每个连接器使用不同的表达式。

拒绝的请求将使用状态码403进行响应。可以使用属性 `denyStatus` 覆盖此状态码。

通过设置属性 `invalidAuthenticationWhenDeny` 为true，可以更改拒绝请求时的行为，不是拒绝请求，而是设置无效的 `authentication` 头。这在与上下文属性 `preemptiveAuthentication="true"` 结合使用时很有用。

注意：该Valve会处理 `ServletRequest.getRemoteHost()` 方法返回的值。为了允许该方法返回正确的主机名，你必须在连接器上启用“DNS查找”功能。

远程主机名的属性包括：

表34：金蝶Apusic应用服务器V10.0.11标准版远程主机名的属性

配置参数	说明	值类型	默认值
classname	要使用的实现的Java类名。必须设置为com.apusic.ams.valves.RemoteHostValve	字符	com.apusic.ams.valves.RemoteHostValve

allow	允许远程访问的主机名，如果指定了此属性，则远程地址必须匹配才能接受此请求。如果未指定此属性，则除非远程主机名与拒绝模式匹配，否则将接受所有请求。	字符	
deny	禁止远程访问的主机名。如果指定了此属性，则远程主机名必须不匹配才能接受此请求。如果未指定此属性，则请求接受仅由allow属性决定。	字符	
denyStatus	拒绝请求时使用的HTTP响应状态码	字符	403
addConnectorPort	将服务器连接器端口附加到客户端主机名后，并使用分号 (“;”) 分隔。如果设置为true，则使用allow和deny配置的表达式将与HOSTNAME;PORT进行比较。其中HOSTNAME是客户端主机名，PORT是接收请求的AAS连接器端口。	字符	false
invalidAuthenticationWhenDeny	当应拒绝请求时，不拒绝请求而是设置无效的身份验证头。这仅当上下文具有属性preemptiveAuthentication="true"设置时才有效。不会覆盖已存在的身份验证头。实际上，这将触发身份验证而不是拒绝，即使应用程序没有配置安全约束也是如此。这可以与addConnectorPort结合使用，以便根据客户端和用于访问应用程序的连接器来触发身份验证	字符	

### 14.2.3 远程CIDR

限制网络掩码访问。允许将提交此请求的客户端的IP地址与一个或多个遵循CIDR表示法的网络掩码进行比较，并根据比较结果允许请求继续或拒绝处理来自此客户端的请求。支持IPv4和IPv6。

此值类似Apache的Order、Allow from和Deny from指令，但存在以下限制：

- Order总是遵循“allow, deny”的顺序；
- 不支持使用点分十进制表示法来指定网络掩码（即，不能写成 `192.168.1.0/255.255.255.0`，而必须写成 `192.168.1.0/24`）；
- 不支持使用快捷方式，如 `10.10.`（这相当于 `10.10.0.0/16`）；
- 正如值的名称所示，这是一个仅支持CIDR的值，因此不支持像 `.mydomain.com` 这样的子域表示法。

将 `addConnectorPort` 属性设置为 `true` 后，可以在服务器连接器端口后附加一个分号 (;)，以便为每个连接器使用不同的表达式。

通过将 `usePeerAddress` 属性设置为`true`，该值将在检查中使用连接的对等地址。如果AAS前面使用了反向代理，并且与AJP协议或HTTP协议以及 `RemoteIp(Valve|Filter)` 结合使用，那么此对等地址将与客户端IP不同。

被拒绝的请求将收到状态码为403的响应。可以使用 `denyStatus` 属性来覆盖此默认状态码。

将 `invalidAuthenticationWhenDeny` 属性设置为`true`后，当请求被拒绝时，行为将更改为不直接拒绝，而是设置一个无效的身份验证头部。这与上下文属性 `preemptiveAuthentication="true"` 结合使用时非常有用。

此值的其他一些特性包括：

- 如果省略了CIDR前缀，此值将变为一个单IP值；
- 与Remote Host Valve不同，它可以处理压缩形式的IPv6地址（如：`::1`、`fe80::/71`等）。

远程CIDR的属性包括：

表35：金蝶Apusic应用服务器V10.0.11标准版远程CIDR的属性

配置参数	说明	值类型	默认值
<code>classname</code>	要使用的实现的Java类名。必须设置为 <code>com.apusic.ams.valves.RemoteCIDRValve</code>	字符	<code>com.apusic.ams.valves.RemoteCIDRValve</code>
<code>allow</code>	一个由逗号分隔的IPv4或IPv6网络掩码或地址列表，用于与远程客户端的IP地址进行匹配。如果指定了此属性，则远程地址必须匹配才能接受此请求。如果未指定此属性，则除非远程IP与 <code>deny</code> 属性中的网络掩码匹配，否则将接受所有请求	字符	
<code>deny</code>	一个由逗号分隔的IPv4或IPv6网络掩码或地址列表，用于与远程客户端的IP地址进行匹配。如果指定了此属性，则远程地址必须不匹配才能接受此请求。如果未指定此属性，则请求接受仅由 <code>allow</code> 属性决定。	字符	
<code>denyStatus</code>	拒绝请求时使用的HTTP响应状态码	字符	403
<code>addConnectorPort</code>	将服务器连接器端口附加到客户端IP地址后，并使用分号（";"）分隔。如果设置为 <code>true</code> ，则使用 <code>allow</code> 和 <code>deny</code> 配置的表达式将与 <code>ADDRESS;PORT</code> 进行比较。其中 <code>ADDRESS</code> 是客户端IP地址， <code>PORT</code> 是接收请求的AAS连接器端口。	字符	<code>false</code>
<code>invalidAuthenticationWhenDeny</code>	当应拒绝请求时，不拒绝请求而是设置无效的身份验证头。这仅当上下文具有属性 <code>preemptiveAuthentication="true"</code> 设置时才有效。不会覆盖已存在的身份验证头。实际上，这将触发身份验证	字符	

	而不是拒绝，即使应用程序没有配置安全约束也是如此。这可以与addConnectorPort结合使用，以便根据客户端和用于访问应用程序的连接器来触发身份验证		
usePeerAddress	使用连接对等方地址而不是客户端IP地址。如果AAS前面使用了反向代理，并且与AJP协议或HTTP协议（加上RemoteIp（Valve Filter））结合使用，那么它们将会有所不同。	字符	

配置示例：

只允许从localhost连接的客户端访问。

```
<Valve className="com.apusic.ams.valves.RemoteCIDRValve"
  allow="127.0.0.1, ::1"/>
```

## 15 附录

### 15.1 附录A listeners配置说明

配置参数	说明	类名
aprlifecycleListener	使用 APR 进行接收客户端请求，在应用服务器初始化之前、初始化之前尝试初始化 APR 库，成功则使用 APR 接受处理客户端请求，应用服务器销毁之后，该监听器会做 APR 的清理工作	com.apusic.ams.core.AprLifecycleListener
jreMemoryLeakPreventionListener	防止 JRE 内存泄露	com.apusic.ams.core.JreMemoryLeakPreventionListener
globalResourcesLifecycleListener	初始化JNDI 资源的 MBean	com.apusic.ams.mbeans.GlobalResourcesLifecycleListener
threadLocalLeakPreventionListener	防止 ThreadLocal 对象带来的内存泄漏	com.apusic.ams.core.ThreadLocalLeakPreventionListener

### 15.2 附录B resources配置说明

配置参数	说明	值类型	默认值
name	资源名称, 必须唯一。如"userDatabase"表示配置用户权限	字符	
auth	授权范围	字符	Container
type	类型	字符	com.apusic.ams.UserDatabase
description	描述	字符	User database that can be updated and saved
factory	资源工厂	字符	com.apusic.ams.users.MemoryUserDatabaseFactory
pathname	路径名称	字符	conf/aas-users.xml

## 15.3 附录C executors 配置说明

配置参数	说明	值类型	默认值
name	线程池名称, 供配置文件其他地方引用, 必须唯一	字符	http-thread-pool
namePrefix	线程池里面的线程的前缀	字符	http-exec-
maxThreads	最大线程数	整型	200
minSpareThreads	最小空闲线程数	整型	25
maxIdleTime	当线程数大于 minSpareThreads 时, 空闲线程销毁前最大空闲时间, 单位毫秒	整型	60000
maxQueueSize	线程池队列最大任务数, 超过后将被拒绝	整型	Integer.MAX_VALUE

## 15.4 附录D endpoints配置说明

配置参数	说明	值类型	默认值
port	监听端口	整型	6888
protocol	处理协议, 可选的有 HTTP/1.1、AJP/1.3	字符	HTTP1.1
connectionTimeout	连接超时时间, 读取 HTTP 请求行超时时间。单位毫秒	数字	3000
redirectPort	如果是 https, 重定向端口	整型	6443
executor	使用的线程池	字符	http-thread-pool
maxConnections	最大并发数	整型	10000
enableLookups	是否启用 DNS 查询, 如果设置 true, 则 request.getRemoteHost() 时, 将调用 DNS 查询, 否则只返回 IP 值	布尔值	false
acceptCount	backlog 大小, 达到 maxConnections 后, 最大队列数, 超出后, 请求将被拒绝	整型	100

maxPostSize	FORM URL 方式的最大提交数据。单位字节	数字	104857600
maxHttpHeaderSize	请求或响应的最大 http header 大小。单位字节	数字	8192 (8KB)
compression	是否启用压缩, 可选值为: off, on, force	字符	on
compressionMinSize	压缩文件值, 小于该值不会进行压缩	整型	2048
compressableMimeType	逗号隔开的可压缩的 MIME 类型	字符	text/html,text/plain,text/css,application/javascript,application/json,application/x-font-ttf,application/x-font-otf,image/svg+xml,image/jpeg,image/png,image/gif,audio/mpeg,video/mp4
connectionUploadTimeout	数据上传超时时间。仅当 disableUploadTimeout=false 时生效。单位毫秒	数字	
disableUploadTimeout	是否禁用上传超时	布尔值	true
URIEncoding	配置 url 中参数编码	字符	UTF-8
processorCache	缓存 Processor 的最大数量	整型	20000
tcpNoDelay	是否禁用了 Nagle 算法, 允许小包的发送	布尔值	true
keepAliveTimeout	keepalive 的超时时间。单位毫秒	数字	15000
maxKeepAliveRequests	保持 keepalive 的连接的个数, 超过就会关闭连接。-1 不限制	整型	100
address	服务器监听地址	IP地址	0.0.0.0 或 ::
maxCookieCount	最大 Cookie 数量	整型	200
relaxedPathChars	允许请求行中的特殊字符	字符	

relaxedQueryChars	允许请求行中参数的特殊字符	字符	
useAsyncIO	是否使用异步 IO API	布尔值	true
useSendfile	是否使用 sendfile 功能。使用 sendfile 后将不压缩响应	布尔值	true
SSLEnabled	是否允许 SSL 通信	布尔值	false
longThreadThreshold	执行时间超过该时间的线程, 判定为“长线程 / 卡死线程”, 单位毫秒	整型	60000
longThreadCycleTime	设置长线程检测间隔, 单位毫秒	整型	3000
longThreadRollingInterval	长线程日志滚动时间间隔, 单位毫秒	整型	300000
enabledLongThread	是否开启长线程检测功能, 值为true或false	布尔值	false
longThreadLogPath	长线程日志存放目录	字符	\${APUSIC_HOME}/logs/longThread
sdosBlacklistEnable	是否开启慢攻击检测, 值为true或false	布尔值	false
sdosThreshold	阈值, 超过指定值加入黑名单, 如果小于0, 则直接加入黑名单	整型	3
sdosTimeWindowMinutes	在30分钟内有多次被判定为慢攻击, 并且次数超过sdosThreshold, 就会被加入黑名单, 如果小于0则直接加入黑名单。单位分钟	整型	30
sdosBlacklistTtlMinutes	黑名单过期时间, 如果小于0, 则不过期	整型	30

#### 15.4.1 SSLHostConfig配置参数

配置参数	说明	值类型	默认值
------	----	-----	-----

protocols	允许与客户端通信的协议名字。可以是以下的组合:SSLv2Hello/SSLv3/TLSv1/TLSv1.1/TLSv1.2/TLSv1.3/SMv1.1/all	字符	all
sslProtocol	服务器使用的 SSL 协议, 用于创建 SSLContext	字符	TLS
truststoreFile	信任库文件, 用于验证客户端证书	字符	系统属性 javax.net.ssl.trustStore
truststoreType	信任库类型	字符	JKS
truststorePassword	信任库密码	字符	系统属性 javax.net.ssl.trustStorePassword
certificateVerification	证书验证, 如果设置为: required: 客户端必须返回有效证书。 optional: 希望客户端返回证书, 若客户端不发证书, 也不报错。 none 不需要客户端证书, 除非客户端请求 "CLIENT-CERT" 限制的资源	字符	none
ciphers	允许使用的密码套件	字符	

#### 15.4.2 Certificate配置参数

配置参数	说明	值类型	默认值
certificateFile	PEM 编码的证书文件	字符	
certificateChainFile	PEM 编码的证书链文件	字符	
certificateKeyFile	PEM 编码的密钥文件	字符	
certificateKeyPassword	密钥文件密码	字符	
certificateKeyAlias	服务器使用的密码和证书在密钥库的别名	字符	
certificateKeystoreFile	密钥库文件	字符	
certificateKeystorePassword	密钥库文件密码	字符	
certificateKeystoreType	密钥库文件类型	字符	JKS
type	证书类型, 用于确认密码套件和证书是匹配的, 必须是为 UNDEFINED,RSA,DSS 和 EC 之一。如果 SSLHostConfig 只有一个 Certificate 元素,type 属性不是必须的。若	字符	UNDEFINED

有多个 Certificate 元素，则必须指定 type 属性
----------------------------------

## 15.5 附录E servers 配置说明

配置参数	说明	值类型	默认值
name	服务名称，必须全局唯一	字符	
defaultHost	默认虚拟主机名称	字符	
jvmRoute	在负载均衡使用场景中，Session 粘滞时使用的标签，在集群节点中必须唯一。此标签将后缀在 Session ID 中，此后同一会话的请求将被同一 AAMS 实例处理。	字符	

### 15.5.1 Realm

配置参数	说明	值类型	默认值
className	Java 类名，此类必须实现 com.apusic.ams.Realm	字符	
dataSourceName	JDBC 数据源的JNDI名字	字符	
userTable	用户表名，表中必须有 userNameCol 和 userCredCol 属性指定的列	字符	
userNameCol	用户名列	字符	
userCredCol	用户凭据列（例：密码）	字符	
userRoleTable	用户角色表，表中必须有 userNameCol 和 roleNameCol 属性指定的列	字符	
roleNameCol	角色列	字符	
resourceName	UserDatabase 资源的名字	字符	UserDatabase
cacheSize	缓存验证失败用户数	整型	1000
failureCount	锁定前验证失败次数	整型	5
lockOutTime	用户锁死时间，在锁死时间内，验证失败将重置计时器，验证成功不起效。单位为秒	整型	300

### 15.5.2 host

配置参数	说明	值类型	默认值
name	虚拟名称, 必须全局唯一, 通常为在 DNS 注册的网络名	字符	
appBase	应用部署基路径	字符	webapps
unpackWARs	是否解压 war 进行部署	布尔值	true
autoDeploy	是否自动部署, 如果为 true, AAMS 将周期性的检查 appBase 目录, 部署新发现的应用, 重新部署更新过的应用	布尔值	true

### 15.5.3 application

配置参数	说明	值类型	默认值
path	应用上下文, 访问路径。同一虚拟主机里, 必须唯一。如果 docBase 在 host 的 appBase 内, 则此属性必须不能设置, 不然同一应用可能会被部署两次。	字符	
docBase	应用的根目录 (应用以目录形式打包), 或者应用 (WAR 包) 的路径, 可以是绝对路径或者相对与 host 元素的 appBase 属性的相对路径	字符	
useHttpOnly	设置为 true 表示防止客户端脚本读取 JSESSIONID, 防止 CSRF/XSS 恶意攻击	布尔值	true
cookie	是否允许 cookies 传递 Session ID	布尔值	true
crossContext	若设置为 true, 则 ServletContext.getContext() 能获得同一虚拟主机下其他应用的 ServletContext. 若设置为 false, 则 ServletContext.getContext() 返回 null.	布尔值	false
logEffectiveWebXml	若为 true, 服务器在应用启动时输出生效的 Web.xml 日志 (INFO) 等级, 生效的 web.xml 包括默认 web.xml, 应用提供的 web.xml、web-fragment.xml 和 annotations 解析结果	布尔值	false

override	若设置为 true，则忽略所有全局和 host 的默认 context 的设置	布尔值	true
reloadable	如果设置为 true，则监控 /WEB-INF/classes/ 和 /WEB-INF/lib/, 若文件有变化，则重新部署应用	布尔值	true
sessionCookieDomain	设置此应用中 Session cookies 的 domain 值	字符	
sessionCookieName	设置 Session cookies 的名字	字符	JSESSIONID
swallowOutput	若设置为 true，应用中输出到 System.out 和 System.err 的内容将被重定向到应用的 logger 中	布尔值	false

#### 15.5.4 servers

配置参数	说明	值类型	默认值
name	定义的server名称，全局唯一	字符	
services	引用的服务, 仅可添加一个服务	字符	
endpoints	引用的网络和协议配置，多个用逗号隔开	字符	
executors	引用的线程池，多个用逗号隔开，注意：该 server 里面的 endpoints 里面引用的线程池也必须在这里引用，如上面的 http-thread-pool	字符	

## 15.6 附录F containers 配置说明

配置参数	说明	值类型	默认值
name	container名称，全局唯一	字符	
port	侦听端口，用于关闭应用服务器，不使用可设置为 -1，建议不使用	整型	
servers	该应用服务器实例引用的服务器，多个可用逗号隔开，建议使用单个	字符	
resources	该应用服务器实例引用的资源，比如用户名密码等，多个可用逗号隔开	字符	
listeners	该应用服务器实例引用的监听器，多个可用逗号隔开。引用的是文件开头配置的监听器	字符	

## 15.7 附录G Resource数据源配置说明

配置参数	说明	值类型	默认值
name	数据源名称	字符	
auth	容器负责资源的连接	字符	
type	绑定类型	字符	
url	数据库连接 url	字符	
username	用户名	字符	
password	密码	字符	
maxTotal	最大连接数	字符	
maxIdle	最大空闲连接数	字符	
maxWaitMillis	最大等待时长（单位为毫秒，-1表示无限期等待）	长整型	0
minIdle	最小空闲连接数	字符	
validationQuery	连接有效性检测语句	字符	
validationQueryTimeout	检测语句超时时间，-1表示不超时	字符	-1
testOnBorrow	true 或 false，从池中拿连接的时，是否检测连接有效性	布尔值	true
testOnReturn	true 或 false，还回连接到池里面时，是否检测连接有效性	布尔值	false
removeAbandonedOnBorrow	true 或 false，是否移除失效的连接	布尔值	false
logAbandoned	true 或 false，是否打印移除失效连接	布尔值	false
testWhileIdle	true 或 false，连接空闲时，是否检测连接有效性	布尔值	false
minEvictableIdleTimeMillis	连接空闲时间，超过该时间会从池中移除（单位为毫秒）	整型	

timeBetweenEvictionRunsMillis	每隔多少时间去检测一次空闲连接是否超时，默认不检测（单位为毫秒）	整型	
-------------------------------	----------------------------------	----	--

全国统一服务热线  
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。

**Apusic**  
金蝶天燕

云计算国家标准制定企业  
金蝶集团旗下基础软件企业  
信息技术应用创新核心企业  
官网: [www.apusic.com](http://www.apusic.com)

