



APUSIC
固若长城
睿比世界

Open - API

金蝶Apusic分布式配置中心V1.0.230

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。

本档如有更新，不另行通知。对本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

 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 概述部分
 - .1.1 API 统一返回体格式
 - .1.2 API 错误码汇总
 - .1.3 API 上下文路径
 - .1.4 API 鉴权
- .2 配置管理
 - .2.1 获取配置
 - .2.2 发布配置
 - .2.3 删除配置
 - .2.4 查询配置历史列表
 - .2.5 查询具体版本的历史配置
 - .2.6 查询配置上一版本信息
 - .2.7 查询指定命名空间下的配置列表
 - .2.8 提交配置但暂不发布
- .3 服务发现
 - .3.1 注册实例
 - .3.2 注销实例
 - .3.3 更新实例
 - .3.4 查询实例详情
 - .3.5 查询指定服务的实例列表
 - .3.6 批量更新实例元数据
 - .3.7 批量删除实例元数据
 - .3.8 更新实例健康状态
 - .3.9 创建服务
 - .3.10 删除服务
 - .3.11 修改服务
 - .3.12 查询服务详情
 - .3.13 查询服务列表
 - .3.14 查询系统开关
 - .3.15 修改系统开关
 - .3.16 查询系统当前数据指标
 - .3.17 查询客户端列表
 - .3.18 查询客户端信息
 - .3.19 查询客户端的注册信息
 - .3.20 查询客户端的订阅信息
 - .3.21 查询注册指定服务的客户端信息
 - .3.22 查询订阅指定服务的客户端信息
- .4 命名空间
 - .4.1 查询命名空间列表
 - .4.2 查询具体命名空间的信息
 - .4.3 创建命名空间
 - .4.4 编辑命名空间
 - .4.5 删除命名空间
- .5 集群管理
 - .5.1 查询当前节点信息
 - .5.2 查询集群节点列表

- 5.3 查询当前节点健康状态
- 5.4 切换集群寻址模式
- 6 连接负载管理
 - 6.1 查询当前节点客户端连接列表
 - 6.2 重新加载当前节点客户端连接数量
 - 6.3 智能平衡集群节点的连接数
 - 6.4 重置指定客户端的连接
 - 6.5 获取集群的SDK指标

1 概述部分

1.1 API 统一返回体格式

所有接口请求的响应均为 json 类型的返回体，返回体具有相同的格式

```
{
  "code": 0,
  "message": "success",
  "data": {}
}
```

返回体中各字段的含义如下表所示

名称	类型	描述
code	int	错误码，0代表执行成功，非0代表执行失败的某一种情况
message	String	错误码提示信息，执行成功为"success"
data	任意类型	返回数据，执行失败时为详细出错信息

由于执行成功的情况下code字段与message字段相同，后续在介绍接口的返回结果时，只介绍返回数据的data字段

1.2 API 错误码汇总

API接口返回体中的错误码及对应提示信息汇总见下表

错误码	提示信息	含义
0	success	成功执行
10000	parameter missing	参数缺失
10001	access denied	访问拒绝
10002	data access error	数据访问错误
20001	'tenant' parameter error	tenant参数错误
20002	parameter validate error	参数验证错误
20003	MediaType Error	请求的MediaType错误
20004	resource not found	资源未找到
20005	resource conflict	资源访问冲突
20006	config listener is null	监听配置为空
20007	config listener error	监听配置错误
20008	invalid dataId	无效的dataId (鉴权失败)
20009	parameter mismatch	请求参数不匹配
21000	service name error	serviceName服务名错误
21001	weight error	weight权重参数错误
21002	instance metadata error	实例metadata元数据错误
21003	instance not found	instance实例不存在

21004	instance error	instance实例信息错误
21005	service metadata error	服务metadata元数据错误
21006	selector error	访问策略selector错误
21007	service already exist	服务已存在
21008	service not exist	服务不存在
21009	service delete failure	存在服务实例，服务删除失败
21010	healthy param miss	healthy参数缺失
21011	health check still running	健康检查仍在运行
22000	illegal namespace	命名空间namespace不合法
22001	namespace not exist	命名空间不存在
22002	namespace already exist	命名空间已存在
23000	illegal state	状态state不合法
23001	node info error	节点信息错误
23002	node down failure	节点离线操作出错
...
30000	server error	其他内部错误

1.3 API 上下文路径

ADCC for nacos 的API上下文路径取决于配置文件 ``${ADCC_HOME}/config/application.properties`` 中 `server.servlet.contextPath` 的取值，默认值如下

```
server.servlet.contextPath=/adcc
```

举例说明，在获取ADCC for nacos配置的URL中，`{adcc_contextpath}` 为上下文路径的变量值：

```
http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/config?
dataId=adcc.example&group=DEFAULT_GROUP&namespaceId=public
```

当设置`server.servlet.contextPath=/adcc`时，`{adcc_contextpath}`的变量值应为`adcc`，则其实际请求URL为：

```
http://127.0.0.1:8848/adcc/v2/cs/config?dataId=adcc.example&group=DEFAULT_GROUP&namespaceId=public
```

当设置`server.servlet.contextPath=/nacos`时，`{adcc_contextpath}`的变量值应为`nacos`，则其实际请求URL为：

```
http://127.0.0.1:8848/nacos/v2/cs/config?
dataId=adcc.example&group=DEFAULT_GROUP&namespaceId=public
```

当设置为`{adcc_contextpath}/v2/cs/temp`时，表示提交配置但没有即时发布。

```
http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/temp?
dataId=adcc.example&group=DEFAULT_GROUP&namespaceId=public
```


2 配置管理

2.1 获取配置

- 接口描述

获取指定配置

- 请求方式

GET

- 请求URL

`/{adcc_contextpath}/v2/cs/config`

- 请求参数

参数名	类型	必填	参数描述
namespaceId	String	否	命名空间, 默认为public与''相同
group	String	是	配置分组名
dataId	String	是	配置名
tag	String	否	标签

- 返回数据

参数名	参数类型	描述
data	String	配置内容

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/config?
dataId=adcc.example&group=DEFAULT_GROUP&namespaceId=public'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": "contentTest"
}
```

2.2 发布配置

- 接口描述

发布指定配置

当配置已存在时, 则对配置进行更新

- 请求方式

POST

Content-Type:application/x-www-form-urlencoded

- 请求URL

/adcc/v2/cs/config

- 请求Body

参数名	类型	必填	参数描述
namespaceId	String	否	命名空间, 默认为public与''相同
group	String	是	配置组名
dataId	String	是	配置名
content	String	是	配置内容
tag	String	否	标签
appName	String	否	应用名
srcUser	String	否	源用户
configTags	String	否	配置标签列表, 可多个, 逗号分隔
desc	String	否	配置描述
use	String	否	-
effect	String	否	-
type	String	否	配置类型
schema	String	否	-

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'dataId=adcc.example' \
  -d 'group=DEFAULT_GROUP' \
  -d 'namespaceId=public' \
  -d 'content=contentTest' \
  -X POST 'http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/config'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

2.3 删除配置

- 接口描述

删除指定配置

- 请求方式

DELETE

- 请求URL

```
/{adcc_contextpath}/v2/cs/config
```

- 请求参数

参数名	类型	必填	参数描述
namespaceId	String	否	命名空间, 默认为public与''相同
group	String	是	配置分组名
dataId	String	是	配置名
tag	String	否	标签

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -X DELETE 'http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/config?
dataId=adcc.example&group=DEFAULT_GROUP&namespaceId=public'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

2.4 查询配置历史列表

- 接口描述

获取指定配置的历史版本列表

- 请求方式

GET

- 请求URL

```
/{adcc_contextpath}/v2/cs/history/list
```

- 请求参数

参数名	类型	必填	参数描述
namespaceId	String	否	命名空间, 默认为public与''相同
group	String	是	配置分组名
dataId	String	是	配置名
pageNo	int	否	当前页, 默认为1
pageSize	int	否	页条目数, 默认为100, 最大为500

- 返回数据

参数名	参数类型	描述说明
data	Object	分页查询结果
data.totalCount	int	总数
data.pageNumber	int	当前页
data.pagesAvailable	int	总页数
data.pageItems	Object[]	历史配置项列表, 参见 历史配置项信息

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/history/list?
dataId=adcc.example&group=DEFAULT_GROUP&namespaceId=public'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "totalCount": 1,
    "pageNumber": 1,
    "pagesAvailable": 1,
    "pageItems": [
      {
        "id": "203",
        "lastId": -1,
        "dataId": "adcc.example",
        "group": "DEFAULT_GROUP",
        "tenant": "",
        "appName": "",
        "md5": "9f67e6977b100e00cab385a75597db58",
        "content": "contentTest",
        "srcIp": "0:0:0:0:0:0:1",

```

```

        "srcUser": null,
        "opType": "I",
        "createdTime": "2010-05-04T16:00:00.000+0000",
        "lastModifiedTime": "2020-12-05T01:48:03.380+0000"
    }
]
}
}

```

2.5 查询具体版本的历史配置

- 接口描述

获取指定版本的历史配置

- 请求方式

GET

- 请求URL

`/ {adcc_contextpath} /v2/cs/history`

- 请求参数

参数名	类型	必填	参数描述
namespaceId	String	否	命名空间, 默认为public与''相同
group	String	是	配置分组名
dataId	String	是	配置名
nid	long	是	历史配置id

- 返回数据

参数名	参数类型	描述说明
data	Object	历史配置项
data.id	String	配置id
data.lastId	int	
data.dataId	String	配置名
data.group	String	配置分组
data.tenant	String	租户信息 (命名空间)
data.appName	String	应用名
data.md5	String	配置内容的md5值
data.content	String	配置内容
data.srcIp	String	源ip
data.srcUser	String	源用户
data.opType	String	操作类型

data.createdTime	String	创建时间
data.lastModifiedTime	String	上次修改时间
data.encryptedDataKey	String	

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/history?
dataId=adcc.example&group=DEFAULT_GROUP&namespaceId=&nid=203'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "id": "203",
    "lastId": -1,
    "dataId": "adcc.example",
    "group": "DEFAULT_GROUP",
    "tenant": "",
    "appName": "",
    "md5": "9f67e6977b100e00cab385a75597db58",
    "content": "contentTest",
    "srcIp": "0:0:0:0:0:0:0:1",
    "srcUser": null,
    "opType": "I",
    "createdTime": "2010-05-04T16:00:00.000+0000",
    "lastModifiedTime": "2020-12-05T01:48:03.380+0000"
  }
}
```

2.6 查询配置上一版本信息

- 接口描述

获取指定配置的上一版本

- 请求方式

GET

- 请求URL

```
{adcc_contextpath}/v2/cs/history/previous
```

- 请求参数

参数名	类型	必填	参数描述
-----	----	----	------

namespaceId	String	否	命名空间, 默认为public与''相同
group	String	是	配置分组名
dataId	String	是	配置名
id	long	是	配置id

- 返回数据

参数名	参数类型	描述说明
data	Object	历史配置项

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/history/previous?
id=309135486247505920&dataId=adcc.example&group=DEFAULT_GROUP&namespaceId='
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "id": "203",
    "lastId": -1,
    "dataId": "adcc.example",
    "group": "DEFAULT_GROUP",
    "tenant": "",
    "appName": "",
    "md5": "9f67e6977b100e00cab385a75597db58",
    "content": "contentTest",
    "srcIp": "0:0:0:0:0:0:0:1",
    "srcUser": null,
    "opType": "I",
    "createdTime": "2010-05-04T16:00:00.000+0000",
    "lastModifiedTime": "2020-12-05T01:48:03.380+0000"
  }
}
```

2.7 查询指定命名空间下的配置列表

- 接口描述

获取指定命名空间下的配置信息列表

- 请求方式

GET

- 请求URL

```
{adcc_contextpath}/v2/cs/history/configs
```

- 请求参数

参数名	类型	必填	参数描述
namespaceId	String	是	命名空间

- 返回数据

参数名	参数类型	描述说明
data	Object[]	配置信息列表
data.id	String	配置id
data.dataId	String	配置名
data.group	String	配置分组
data.content	String	配置内容
data.md5	String	配置内容的md5值
data.encryptedDataKey	String	
data.tenant	String	租户信息 (命名空间)
data.appName	String	应用名
data.type	String	配置文件类型
data.lastModified	long	上次修改时间

返回数据中的配置信息只有 dataId, group, tenant, appName, type 字段有效, 其他字段为默认值

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/cs/history/configs?namespaceId='
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": [
    {
      "id": "0",
      "dataId": "adcc.example",
      "group": "DEFAULT_GROUP",
      "content": null,
      "md5": null,
      "encryptedDataKey": null,
      "tenant": "",
      "appName": "",
      "type": "yaml",
      "lastModified": 0
    }
  ]
}
```

```

    }
  ]
}

```

2.8 提交配置但暂不发布

- 接口描述

支持提交配置，但暂时不发布。

- 请求URL

```
/{adcc_contextpath}/v2/cs/temp
```

- 示例

新增和修改:

```
curl -d 'dataId=nacos.example' -d 'group=DEFAULT_GROUP' -d 'namespaceId=public' -d
'content=contentTest' -X POST 'http://127.0.0.1:8848/adcc/v2/cs/temp'
```

查询:

```
curl 'http://127.0.0.1:8848/adcc/v2/cs/temp?
dataId=nacos.example&group=DEFAULT_GROUP&namespaceId=public'
```

删除:

```
curl -d 'dataId=nacos.example' -d 'group=DEFAULT_GROUP' -d 'namespaceId=public' -X
DELETE 'http://127.0.0.1:8848/adcc/v2/cs/temp'
```

发布:

```
curl -d 'dataId=nacos.example' -d 'group=DEFAULT_GROUP' -d 'namespaceId=public' -X POST
'http://127.0.0.1:8848/adcc/v2/cs/temp/publish'
```

3 服务发现

3.1 注册实例

- 接口描述

注册一个实例

- 请求方式

```
POST
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
/{adcc_contextpath}/v2/ns/instance
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
ip	String	是	IP地址
port	int	是	端口号
clusterName	String	否	集群名称, 默认为DEFAULT
healthy	boolean	否	是否只查找健康实例, 默认为true
weight	double	否	实例权重, 默认为1.0
enabled	boolean	否	是否可用, 默认为true
metadata	JSON格式String	否	实例元数据
ephemeral	boolean	否	是否为临时实例

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'serviceName=test_service' \
-d 'ip=127.0.0.1' \
-d 'port=8090' \
-d 'weight=0.9' \
-d 'ephemeral=true' \
-X POST 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/instance'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

3.2 注销实例

- 接口描述

注销指定实例

- 请求方式

```
DELETE
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
{adcc_contextpath}/v2/ns/instance
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
ip	String	是	IP地址
port	int	是	端口号
clusterName	String	否	集群名称, 默认为DEFAULT
healthy	boolean	否	是否只查找健康实例, 默认为true
weight	double	否	实例权重, 默认为1.0
enabled	boolean	否	是否可用, 默认为true
metadata	JSON格式String	否	实例元数据
ephemeral	boolean	否	是否为临时实例

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'serviceName=test_service' \
-d 'ip=127.0.0.1' \
-d 'port=8090' \
-d 'weight=0.9' \
```

```
-d 'ephemeral=true' \
-X DELETE 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/instance'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

3.3 更新实例

- 接口描述

修改实例信息

通过该接口更新的元数据拥有更高的优先级，且具有记忆能力；会在对应实例删除后，依旧存在一段时间，如果在此期间实例重新注册，该元数据依旧生效；您可以通过 `nacos.naming.clean.expired-metadata.expired-time` 及 `nacos.naming.clean.expired-metadata.interval` 对记忆时间进行修改

- 请求方式

```
PUT
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
{adcc_contextpath}/v2/ns/instance
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
ip	String	是	IP地址
port	int	是	端口号
clusterName	String	否	集群名称, 默认为DEFAULT
healthy	boolean	否	是否只查找健康实例, 默认为true
weight	double	否	实例权重, 默认为1.0
enabled	boolean	否	是否可用, 默认为true
metadata	JSON格式String	否	实例元数据
ephemeral	boolean	否	是否为临时实例

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'serviceName=test_service' \
  -d 'ip=127.0.0.1' \
  -d 'port=8090' \
  -d 'weight=0.9' \
  -d 'ephemeral=true' \
  -X PUT 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/instance'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

3.4 查询实例详情

- 接口描述

查询某个具体实例的详情信息

- 请求方式

GET

- 请求URL

`{adcc_contextpath}/v2/ns/instance`

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
clusterName	String	否	集群名称, 默认为DEFAULT
ip	String	是	IP地址
port	int	是	端口号

- 返回数据

参数名	参数类型	描述说明
data	Object	实例详情信息
data.serviceName	String	服务名
data.ip	String	IP地址

data.port	int	端口号
data.clusterName	String	集群名称
data.weight	double	实例权重
data.healthy	boolean	是否健康
data.instanceId	String	实例id
data.metadata	map	实例元数据

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/instance?namespaceId=public&groupName=&serviceName=test_service&ip=127.0.0.1&port=8090'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "serviceName": "DEFAULT_GROUP@@test_service",
    "ip": "127.0.0.1",
    "port": 8090,
    "clusterName": "DEFAULT",
    "weight": 1.0,
    "healthy": true,
    "instanceId": null,
    "metadata": {
      "value": "1"
    }
  }
}
```

3.5 查询指定服务的实例列表

- 接口描述

查询指定服务下的实例详情信息列表

- 请求方式

GET

- 请求URL

```
{adcc_contextpath}/v2/ns/instance/list
```

- 请求头

参数名	参数类型	是否必填	描述说明
-----	------	------	------

User-Agent	String	否	用户代理, 默认为空
Client-Version	String	否	客户端版本, 默认为空

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
clusterName	String	否	集群名称, 默认为DEFAULT
ip	String	否	IP地址, 默认为空, 表示不限制IP地址
port	int	否	端口号, 默认为0, 表示不限制端口号
healthyOnly	boolean	否	是否只获取健康实例, 默认为false
app	String	否	应用名, 默认为空

- 返回数据

参数名	参数类型	描述说明
data		指定服务的实例列表
data.name	String	分组名@@服务名
data.groupName	String	分组名
data.clusters	String	集群名
data.cacheMillis	int	缓存时间
data.hosts	Object[]	实例列表
data.hosts.ip	String	实例IP
data.hosts.port	int	实例端口号
data.hosts.weight	double	实例权重
data.hosts.healthy	boolean	实例是否健康
data.hosts.enabled	boolean	实例是否可用
data.hosts.ephemeral	boolean	是否为临时实例
data.hosts.clusterName	String	实例所在的集群名称
data.hosts.serviceName	String	服务名
data.hosts.metadata	map	实例元数据
data.hosts.instanceHeartBeatTimeOut	int	实例心跳超时时间
data.hosts.ipDeleteTimeout	int	实例删除超时时间
data.hosts.instanceHeartBeatInterval	int	实例心跳间隔
data.lastRefTime	int	上次刷新时间
data.checksum	int	校验码
data.allIPs	boolean	

data.reachProtectionThreshold	boolean	是否到达保护阈值
data.valid	boolean	是否有效

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/instance/list?
serviceName=test_service&ip=127.0.0.1'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "name": "DEFAULT_GROUP@@test_service",
    "groupName": "DEFAULT_GROUP",
    "clusters": "",
    "cacheMillis": 10000,
    "hosts": [
      {
        "ip": "127.0.0.1",
        "port": 8080,
        "weight": 1.0,
        "healthy": true,
        "enabled": true,
        "ephemeral": true,
        "clusterName": "DEFAULT",
        "serviceName": "DEFAULT_GROUP@@test_service",
        "metadata": {
          "value": "1"
        },
        "instanceHeartBeatTimeOut": 15000,
        "ipDeleteTimeout": 30000,
        "instanceHeartBeatInterval": 5000
      }
    ],
    "lastRefTime": 1662554390814,
    "checksum": "",
    "allIPs": false,
    "reachProtectionThreshold": false,
    "valid": true
  }
}
```

3.6 批量更新实例元数据

- 接口描述

批量更新实例的元数据,

对应元数据的键不存在时, 则添加对应元数据

- 请求方式

```
PUT
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
/{adcc_contextpath}/v2/ns/instance/metadata/batch
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
consistencyType	String	否	持久化类型, 默认为空
instances	JSON格式String	否	需要更新的实例列表, 默认为空
metadata	JSON格式String	是	实例元数据

- 参数说明

- `consistencyType`: 实例的持久化类型, 当为 `persist`, 表示对持久化实例的元数据进行更新; 否则表示对临时实例的元数据进行更新
- `instances`: 待更新的实例列表, `json` 数组, 通过 `ip+port+ephemeral+cluster` 定位到某一实例, 为空则表示更新指定服务下所有实例的元数据

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'serviceName=test_service' \
-d 'consistencyType=persist' \
-d 'instances=[{"ip":"127.0.0.1","port": "8090","ephemeral":"false"},
{"ip":"2.2.2.2","port":"8080","ephemeral":"false"}]' \
-d 'metadata={"age":"20","name":"test_update44"}' \
-X PUT 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/instance/metadata/batch'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
```

```
"data": true
}
```

3.7 批量删除实例元数据

- 接口描述

批量删除实例的元数据,

对应元数据的键不存在时, 则不做操作

- 请求方式

```
DELETE
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
{adcc_contextpath}/v2/ns/instance/metadata/batch
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
consistencyType	String	否	持久化类型, 默认为空
instances	JSON格式String	否	需要更新的实例列表, 默认为空
metadata	JSON格式String	是	实例元数据

- 参数说明

- `consistencyType`: 实例的持久化类型, 当为 `persist`, 表示对持久化实例的元数据进行删除; 否则表示对临时实例的元数据进行
- `instances`: 待更新的实例列表, `json` 数组, 通过 `ip+port+ephemeral+cluster` 定位到某一实例, 为空则表示更新指定服务下所有实例的元数据

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'serviceName=test_service' \
-d 'ip=127.0.0.1' \
-d 'port=8090' \
-d 'weight=0.9' \
-d 'ephemeral=false' \
-X DELETE 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/instance'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

3.8 更新实例健康状态

- 接口描述

更新实例的健康状态,仅在集群的健康检查关闭时才生效,当集群配置了健康检查时,该接口会返回错误

- 请求方式

```
PUT
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
{adcc_contextpath}/v2/ns/health/instance
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
clusterName	String	否	集群名, 默认为DEFAULT
ip	String	是	IP地址
port	int	是	端口号
healthy	boolean	是	是否健康

- 返回数据

参数名	参数类型	描述
data	String	"ok"表示执行成功

- 请求示例

```
curl -d 'serviceName=test_service' \
  -d 'ip=127.0.0.1' \
  -d 'port=8090' \
  -d 'healthy=false' \
  -X PUT 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/health/instance'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": "ok"
}
```

3.9 创建服务

- 接口描述

创建一个服务

服务已存在时会创建失败

- 请求方式

```
POST
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
{adcc_contextpath}/v2/ns/service
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
metadata	JSON格式String	否	服务元数据, 默认为空
ephemeral	boolean	否	是否为临时实例, 默认为false
protectThreshold	float	否	保护阈值, 默认为0
selector	JSON格式String	否	访问策略, 默认为空

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'serviceName=adcc.test.service001' \
  -d 'ephemeral=true' \
  -d 'metadata={"k1":"v1"}' \
  -X POST 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/service'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

3.10 删除服务

- 接口描述

删除指定服务

服务不存在时会报错，且服务还存在实例时会删除失败

- 请求方式

DELETE

- 请求URL

`/{adcc_contextpath}/v2/ns/service`

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -X DELETE 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/service?
serviceName=adcc.test.service001'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

3.11 修改服务

- 接口描述

更新指定服务

服务不存在时会报错

- 请求方式

```
POST
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
/{adcc_contextpath}/v2/ns/service
```

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
metadata	JSON格式String	否	服务元数据, 默认为空
protectThreshold	float	否	保护阈值, 默认为0
selector	JSON格式String	否	访问策略, 默认为空

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'serviceName=adcc.test.service001' \
-d 'metadata={"k1":"v2"}' \
-X PUT 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/service'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

3.12 查询服务详情

- 接口描述

查询某个具体服务的详情信息

服务不存在时会报错

- 请求方式

GET

- 请求URL

```
{adcc_contextpath}/v2/ns/service
```

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名

- 返回数据

参数名	参数类型	描述说明
data		服务信息
data.namespace	String	命名空间
data.groupName	String	分组名
data.serviceName	String	服务名
data.clusterMap	map	集群信息
data.metadata	map	服务元数据
data.protectThreshold	float	保护阈值
data.selector	Object	访问策略
data.ephemeral	Boolean	是否为临时实例

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/service?
serviceName=adcc.test.service001'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "namespace": "public",
    "serviceName": "adcc.test.service001",
    "groupName": "DEFAULT_GROUP",
    "clusterMap": {},
    "metadata": {},
    "protectThreshold": 0,
    "selector": {
      "type": "none",
```

```

        "contextType": "NONE"
    },
    "ephemeral": false
}
}

```

3.13 查询服务列表

- 接口描述

查询符合条件的服务列表

- 请求方式

GET

- 请求URL

```
/{adcc_contextpath}/v2/ns/service/list
```

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
selector	JSON格式String	是	访问策略
pageNo	int	否	当前页, 默认为1
pageSize	int	否	页条目数, 默认为20, 最大为500

- 返回数据

参数名	参数类型	描述说明
data		服务列表信息
data.count	String	服务数目
data.services	String[]	分页后的服务列表

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/service/list'
```

- 返回示例

```

{
  "code": 0,
  "message": "success",
  "data": {
    "count": 2,
    "services": [
      "adcc.test.service001",

```

```

        "adcc.test.service002"
    ]
}
}

```

3.14 查询系统开关

- 接口描述

查询系统开关

- 请求方式

GET

- 请求URL

```
/{adcc_contextpath}/v2/ns/operator/switches
```

- 返回数据

参数名	参数类型	描述说明
data	Object	系统开关信息

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/operator/switches'
```

- 返回示例

```

{
  "code": 0,
  "message": "success",
  "data": {
    "masters": null,
    "adWeightMap": {
    },
    "defaultPushCacheMillis": 10000,
    "clientBeatInterval": 5000,
    "defaultCacheMillis": 3000,
    "distroThreshold": 0.7,
    "healthCheckEnabled": true,
    "autoChangeHealthCheckEnabled": true,
    "distroEnabled": true,
    "enableStandalone": true,
    "pushEnabled": true,
    "checkTimes": 3,
    "httpHealthParams": {
      "max": 5000,
    }
  }
}

```

```

        "min": 500,
        "factor": 0.85
    },
    "tcpHealthParams": {
        "max": 5000,
        "min": 1000,
        "factor": 0.75
    },
    "mysqlHealthParams": {
        "max": 3000,
        "min": 2000,
        "factor": 0.65
    },
    "incrementalList": [

    ],
    "serverStatusSynchronizationPeriodMillis": 2000,
    "serviceStatusSynchronizationPeriodMillis": 5000,
    "disableAddIP": false,
    "sendBeatOnly": false,
    "lightBeatEnabled": true,
    "limitedUrlMap": {

    },
    "distroServerExpiredMillis": 10000,
    "pushGoVersion": "0.1.0",
    "pushJavaVersion": "0.1.0",
    "pushPythonVersion": "0.4.3",
    "pushCVersion": "1.0.12",
    "pushCSharpVersion": "0.9.0",
    "enableAuthentication": false,
    "overriddenServerStatus": null,
    "defaultInstanceEphemeral": true,
    "healthCheckWhiteList": [

    ],
    "checksum": null,
    "name": "00-00---000-NACOS_SWITCH_DOMAIN-000---00-00"
}
}

```

3.15 修改系统开关

- 接口描述

修改系统开关

- 请求方式

```
PUT
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
{adcc_contextpath}/v2/ns/operator/switches
```

- 请求Body

参数名	参数类型	是否必填	描述说明
entry	String	是	开关名
value	String	是	开关值
debug	boolean	否	是否只在本机生效,true表示本机生效,false表示集群生效

- 返回数据

参数名	参数类型	描述
data	String	"ok"表示执行成功

- 请求示例

```
curl -d 'entry=pushEnabled' \
-d 'value=false' \
-d 'debug=true' \
-X PUT 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/operator/switches'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": "ok"
}
```

3.16 查询系统当前数据指标

- 接口描述

查询系统当前数据指标

- 请求方式

```
GET
```

- 请求URL

```
{adcc_contextpath}/v2/ns/operator/metrics
```

- 请求参数

参数名	参数类型	是否必填	描述说明
onlyStatus	boolean	否	只显示状态，默认为true

当 `onlyStatus` 设置为 `true` 时，只返回表示系统状态的字符串

- 返回数据

参数名	参数类型	描述说明
data	Object	系统当前数据指标
data.status	String	系统状态
data.serviceCount	int	服务数量
data.instanceCount	int	实例数量
data.subscribeCount	int	订阅数量
data.raftNotifyTaskCount	int	Raft通知任务数量
data.responsibleServiceCount	int	
data.responsibleInstanceCount	int	
data.clientCount	int	客户端数量
data.connectionBasedClientCount	int	连接数量
data.ephemeralIpPortClientCount	int	临时客户端数量
data.persistentIpPortClientCount	int	持久客户端数量
data.responsibleClientCount	int	
data.cpu	float	cpu使用率
data.load	float	负载
data.mem	float	内存使用率

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/operator/metrics?
onlyStatus=false'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "status": "UP",
    "serviceCount": 2,
    "instanceCount": 2,
    "subscribeCount": 2,
    "raftNotifyTaskCount": 0,
    "responsibleServiceCount": 0,
    "responsibleInstanceCount": 0,
```

```

    "clientCount": 2,
    "connectionBasedClientCount": 2,
    "ephemeralIpPortClientCount": 0,
    "persistentIpPortClientCount": 0,
    "responsibleClientCount": 2,
    "cpu": 0,
    "load": -1,
    "mem": 1
  }
}

```

3.17 查询客户端列表

- 接口描述

查询当前所有的客户端列表

- 请求方式

GET

- 请求URL

`/ {adcc_contextpath} /v2/ns/client/list`

- 返回数据

参数名	参数类型	描述说明
data	String[]	客户端id列表

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/client/list'
```

- 返回示例

```

{
  "code": 0,
  "message": "success",
  "data": [
    "127.0.0.1:8099#false",
    "127.0.0.1:8090#false",
    "127.0.0.1:8088#false",
    "1706235962402_172.24.6.163_55736",
    "172.24.4.42:80#false",
    "172.24.4.27:6868#false",
    "127.0.0.1:8848#false"
  ]
}

```

```
]
}
```

ADCC支持不同版本的nacos client, 建立客户端的方式不同。

对于 1.x 版本的 nacos client, 每个实例会建立两个基于 ip+port 的客户端, 分别对应实例注册与服务订阅, clientId 格式为 ip:port#ephemeral

对于 2.x 版本的 nacos client, 每个实例会建立一个 RPC 连接, 对应一个基于 RPC 连接的客户端, 兼具注册与订阅功能, clientId 格式为

```
time_ip_port
```

3.18 查询客户端信息

- 接口描述

查询指定客户端的详细信息

客户端不存在时会报错

- 请求方式

GET

- 请求URL

```
/{adcc_contextpath}/v2/ns/client
```

- 请求参数

参数名	参数类型	是否必填	描述说明
clientId	String	是	客户端id

- 返回数据

参数名	参数类型	描述说明
data	Object	客户端信息
data.clientId	String	客户端id
data.ephemeral	boolean	是否为临时实例
data.lastUpdatedTime	int	上次更新时间
data.clientType	String	客户端类型
data.clientIp	String	客户端IP
data.clientPort	String	客户端端口
data.connectType	String	连接类型
data.appName	String	应用名
data.Version	String	客户端版本

只有当 clientType 为 connection 时, 会显示 connectType, appName 和 appName 字段

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/client?
clientId=1706235962402_172.24.6.163_55736'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "clientId": "1706235962402_172.24.6.163_55736",
    "ephemeral": true,
    "lastUpdatedTime": 1706235962563,
    "clientType": "connection",
    "connectType": "GRPC",
    "appName": "unknown",
    "version": "Nacos-Java-Client:v2.3.0",
    "clientIp": "172.24.6.163",
    "clientPort": "55736"
  }
}
```

3.19 查询客户端的注册信息

- 接口描述

查询指定客户端的注册信息

客户端不存在时会报错

- 请求方式

GET

- 请求URL

`/ {@adcc_contextpath} /v2/ns/client/publish/list`

- 请求参数

参数名	参数类型	是否必填	描述说明
clientId	String	是	客户端id

- 返回数据

参数名	参数类型	描述说明
data	Object[]	客户端注册的服务列表
data.namespace	String	命名空间
data.group	String	分组名
data.serviceName	String	服务名
data.registeredInstance	Object	该服务下注册的实例
data.registeredInstance.ip	String	IP地址
data.registeredInstance.port	int	端口号

data.registeredInstance.cluster	String	集群名
---------------------------------	--------	-----

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/client/publish/list?
clientId=1706235962402_172.24.6.163_55736'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": [
    {
      "namespace": "public",
      "group": "DEFAULT_GROUP",
      "serviceName": "test_service421",
      "registeredInstance": {
        "ip": "172.24.4.163",
        "port": 6888,
        "cluster": "DEFAULT"
      }
    }
  ]
}
```

3.20 查询客户端的订阅信息

- 接口描述

查询指定客户端的订阅信息

客户端不存在时会报错

- 请求方式

GET

- 请求URL

```
{adcc_contextpath}/v2/ns/client/subscribe/list
```

- 请求参数

参数名	参数类型	是否必填	描述说明
clientId	String	是	客户端id

- 返回数据

参数名	参数类型	描述说明
data	Object[]	客户端订阅的服务列表

data.namespace	String	命名空间
data.group	String	分组名
data.serviceName	String	服务名
data.subscriberInfo	Object	订阅信息
data.subscriberInfo.app	String	应用
data.subscriberInfo.agent	String	客户端信息
data.subscriberInfo.addr	String	地址

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/client/subscribe/list?
clientId=1706235962402_172.24.6.163_55736'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": [
    {
      "namespace": "public",
      "group": "DEFAULT_GROUP",
      "serviceName": "test_service421",
      "subscriberInfo": {
        "app": "unknown",
        "agent": "Nacos-Java-Client:v2.3.0",
        "addr": "172.24.6.163"
      }
    }
  ]
}
```

3.21 查询注册指定服务的客户端信息

- 接口描述

查询注册指定服务的客户端信息

- 请求方式

```
GET
```

- 请求URL

```
{adcc_contextpath}/v2/ns/client/service/publisher/list
```

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
ephemeral	boolean	否	是否为临时实例
ip	String	否	IP地址, 默认为空, 不限制IP地址
port	int	否	端口号, 默认为空, 表示不限制端口号

- 返回数据

参数名	参数类型	描述说明
data		客户端列表
data.clientId	String	客户端id
data.ip	String	客户端IP
data.port	int	客户端端口

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/client/service/publisher/list?serviceName=test_service421&ip=&port='
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": [
    {
      "clientId": "1706235962402_172.24.6.163_55736",
      "ip": "10.128.164.35",
      "port": 9950
    }
  ]
}
```

3.22 查询订阅指定服务的客户端信息

- 接口描述

查询订阅指定服务的客户端信息

- 请求方式

```
GET
```

- 请求URL

```
{adcc_contextpath}/v2/ns/client/service/subscriber/list
```

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	否	命名空间Id, 默认为public
groupName	String	否	分组名, 默认为DEFAULT_GROUP
serviceName	String	是	服务名
ephemeral	boolean	否	是否为临时实例
ip	String	否	IP地址, 默认为空, 不限制IP地址
port	int	否	端口号, 默认为空, 表示不限制端口号

- 返回数据

参数名	参数类型	描述说明
data		客户端列表
data.clientId	String	客户端id
data.ip	String	客户端IP
data.port	int	客户端端口

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/ns/client/service/subscriber/list?serviceName=nacos.test.1&ip=&port='
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": [
    {
      "clientId": "1664527125645_127.0.0.1_4443",
      "ip": "10.128.164.35",
      "port": 0
    },
    {
      "clientId": "172.24.144.1:54126#true",
      "ip": "172.24.144.1",
      "port": 54126
    }
  ]
}
```

4 命名空间

4.1 查询命名空间列表

- 接口描述

查询当前所有的命名空间

- 请求方式

GET

- 请求URL

```
{adcc_contextpath}/v2/console/namespace/list
```

- 返回数据

参数名	参数类型	描述说明
data	Object[]	命名空间列表
data.namespace	String	命名空间ID
data.namespaceShowName	String	命名空间名称
data.namespaceDesc	String	命名空间描述
data.quota	int	命名空间的容量
data.configCount	int	命名空间下的配置数量
data.type	int	命名空间类型

命名空间分为3种类型, 0 - 全局命名空间 1 - 默认私有命名空间 2 - 自定义命名空间

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/console/namespace/list'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": [
    {
      "namespace": "",
      "namespaceShowName": "public",
      "namespaceDesc": null,
      "quota": 200,
      "configCount": 1,
      "type": 0
    }
  ]
}
```

```
]
}
```

4.2 查询具体命名空间的信息

- 接口描述

查询具体命名空间的信息

命名空间不存在时会报错

- 请求方式

GET

- 请求URL

```
/{adcc_contextpath}/v2/console/namespace
```

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	是	命名空间Id

- 返回数据

参数名	参数类型	描述说明
data	Object	命名空间信息
data.namespace	String	命名空间ID
data.namespaceShowName	String	命名空间名称
data.namespaceDesc	String	命名空间描述
data.quota	int	命名空间的容量
data.configCount	int	命名空间下的配置数量
data.type	int	命名空间类型

命名空间分为3种类型, 0 - 全局命名空间 1 - 默认私有命名空间 2 - 自定义命名空间

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/console/namespace?
namespaceId=test_namespace'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": {
    "namespace": "test_namespace",
    "namespaceShowName": "test",
```

```

    "namespaceDesc": null,
    "quota": 200,
    "configCount": 0,
    "type": 2
  }
}

```

4.3 创建命名空间

- 接口描述

创建一个命名空间

命名空间已存在时会报错

- 请求方式

```

POST
Content-Type:application/x-www-form-urlencoded

```

- 请求URL

```
/{adcc_contextpath}/v2/console/namespace
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	是	命名空间Id
namespaceName	String	是	命名空间名称
namespaceDesc	String	否	命名空间描述

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```

curl -d 'namespaceId=test_namespace' \
-d 'namespaceName=test' \
-X POST 'http://127.0.0.1:8848/{adcc_contextpath}/v2/console/namespace'

```

- 返回示例

```

{
  "code": 0,
  "message": "success",
  "data": true
}

```

4.4 编辑命名空间

- 接口描述

编辑命名空间信息

- 请求方式

```
PUT
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
/{adcc_contextpath}/v2/console/namespace
```

- 请求Body

参数名	参数类型	是否必填	描述说明
namespaceId	String	是	命名空间Id
namespaceName	String	是	命名空间名称
namespaceDesc	String	否	命名空间描述

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'namespaceId=test_namespace' \
-d 'namespaceName=test.adcc' \
-X PUT 'http://127.0.0.1:8848/{adcc_contextpath}/v2/console/namespace'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

4.5 删除命名空间

- 接口描述

删除指定命名空间

- 请求方式

```
DELETE
```

- 请求URL

```
/{adcc_contextpath}/v2/console/namespace
```

- 请求参数

参数名	参数类型	是否必填	描述说明
namespaceId	String	是	命名空间Id

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'namespaceId=test_namespace' \  
-X DELETE 'http://127.0.0.1:8848/{adcc_contextpath}/v2/console/namespace'
```

- 返回示例

```
{  
  "code": 0,  
  "message": "success",  
  "data": true  
}
```

5 集群管理

5.1 查询当前节点信息

- 接口描述

查询当前ADCC节点信息

- 请求方式

GET

- 请求URL

`/ {adcc_contextpath} /v2/core/cluster/node/self`

- 返回数据

参数名	参数类型	描述说明
data	Object	当前节点信息
data.ip	String	节点IP地址
data.port	int	节点端口
data.state	String	节点状态
data.extendInfo	Object	节点扩展信息
data.address	String	节点地址 (IP:port)
data.failAccessCnt	int	失败访问次数
data.abilities	Object	

请求示例:

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/core/cluster/node/self'
```

返回示例:

```
{
  "code": 0,
  "message": "success",
  "data": {
    "ip": "172.24.4.152",
    "port": 8848,
    "state": "UP",
    "extendInfo": {
      "lastRefreshTime": 1705917920473,
      "raftMetaData": {
        "metaDataMap": {
          "naming_instance_metadata": {
            "leader": "172.24.4.152:7848",
            "raftGroupMember": [
              "172.24.4.163:7848",
```

```

        "172.24.4.154:7848",
        "172.24.4.152:7848"
    ],
    "term": 1
},
"naming_persistent_service": {
    "leader": "172.24.4.152:7848",
    "raftGroupMember": [
        "172.24.4.163:7848",
        "172.24.4.154:7848",
        "172.24.4.152:7848"
    ],
    "term": 1
},
"naming_persistent_service_v2": {
    "leader": "172.24.4.152:7848",
    "raftGroupMember": [
        "172.24.4.163:7848",
        "172.24.4.154:7848",
        "172.24.4.152:7848"
    ],
    "term": 1
},
"naming_service_metadata": {
    "leader": "172.24.4.152:7848",
    "raftGroupMember": [
        "172.24.4.163:7848",
        "172.24.4.154:7848",
        "172.24.4.152:7848"
    ],
    "term": 1
}
}
},
"raftPort": "7848",
"readyToUpgrade": true,
"version": "1.0.230"
},
"address": "172.24.4.152:8848",
"failAccessCnt": 0,
"abilities": {
    "remoteAbility": {
        "supportRemoteConnection": true,
        "grpcReportEnabled": true
    },
    "configAbility": {
        "supportRemoteMetrics": false
    },
    "namingAbility": {
        "supportJraft": true
    }
}
}

```

```

    },
    "grpcReportEnabled": true
  }
}

```

5.2 查询集群节点列表

- 接口描述

查询集群节点列表

- 请求方式

GET

- 请求URL

```
/{adcc_contextpath}/v2/core/cluster/node/list
```

- 请求参数

参数名	参数类型	是否必填	描述说明
address	String	否	节点地址，默认为空
state	String	否	节点状态，默认为空

`address` 对应于需要查询的节点地址的前缀匹配条件，为空时不做限制

`state` 对应节点状态的筛选条件，为空时不做限制

- 返回数据

参数名	参数类型	描述说明
data	Object[]	节点列表，详情参见节点详情

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{adcc_contextpath}/v2/core/cluster/node/list'
```

- 返回示例

```

{
  "code": 0,
  "message": "success",
  "data": [
    {
      "ip": "172.24.4.154",
      "port": 8848,
      "state": "UP",
      "extendInfo": {
        "lastRefreshTime": 1706087752497,
        "raftMetaData": {
          "metaDataMap": {

```

```

    "naming_instance_metadata": {
      "leader": "172.24.4.152:7848",
      "raftGroupMember": [
        "172.24.4.163:7848",
        "172.24.4.154:7848",
        "172.24.4.152:7848"
      ],
      "term": 1
    },
    "naming_persistent_service": {
      "leader": "172.24.4.152:7848",
      "raftGroupMember": [
        "172.24.4.163:7848",
        "172.24.4.154:7848",
        "172.24.4.152:7848"
      ],
      "term": 1
    },
    "naming_persistent_service_v2": {
      "leader": "172.24.4.152:7848",
      "raftGroupMember": [
        "172.24.4.163:7848",
        "172.24.4.154:7848",
        "172.24.4.152:7848"
      ],
      "term": 1
    },
    "naming_service_metadata": {
      "leader": "172.24.4.152:7848",
      "raftGroupMember": [
        "172.24.4.163:7848",
        "172.24.4.154:7848",
        "172.24.4.152:7848"
      ],
      "term": 1
    }
  },
  "raftPort": "7848",
  "readyToUpgrade": true,
  "version": "1.0.230"
},
"address": "172.24.4.154:8848",
"failAccessCnt": 0,
"abilities": {

```

```

    "remoteAbility": {
      "supportRemoteConnection": true,
      "grpcReportEnabled": true
    },
    "configAbility": {
      "supportRemoteMetrics": false
    },
    "namingAbility": {
      "supportJraft": true
    }
  },
  "grpcReportEnabled": true
},
{
  "ip": "172.24.4.152",
  "port": 8848,
  "state": "UP",
  "extendInfo": {
    "lastRefreshTime": 1706087752877,
    "raftMetaData": {
      "metaDataMap": {
        "naming_instance_metadata": {
          "leader": "172.24.4.152:7848",
          "raftGroupMember": [
            "172.24.4.163:7848",
            "172.24.4.154:7848",
            "172.24.4.152:7848"
          ],
          "term": 1
        },
        "naming_persistent_service": {
          "leader": "172.24.4.152:7848",
          "raftGroupMember": [
            "172.24.4.163:7848",
            "172.24.4.154:7848",
            "172.24.4.152:7848"
          ],
          "term": 1
        },
        "naming_persistent_service_v2": {
          "leader": "172.24.4.152:7848",
          "raftGroupMember": [
            "172.24.4.163:7848",
            "172.24.4.154:7848",
            "172.24.4.152:7848"
          ]
        }
      }
    }
  }
}

```

```

        ],
        "term": 1
    },
    "naming_service_metadata": {
        "leader": "172.24.4.152:7848",
        "raftGroupMember": [
            "172.24.4.163:7848",
            "172.24.4.154:7848",
            "172.24.4.152:7848"
        ],
        "term": 1
    }
}

},
"raftPort": "7848",
"readyToUpgrade": true,
"version": "1.0.230"
},
"address": "172.24.4.152:8848",
"failAccessCnt": 0,
"abilities": {
    "remoteAbility": {
        "supportRemoteConnection": true,
        "grpcReportEnabled": true
    },
    "configAbility": {
        "supportRemoteMetrics": false
    },
    "namingAbility": {
        "supportJraft": true
    }
},
"grpcReportEnabled": true
},
{
    "ip": "172.24.4.163",
    "port": 8848,
    "state": "UP",
    "extendInfo": {
        "lastRefreshTime": 1705989255350,
        "raftMetaData": {
            "metaDataMap": {
                "naming_instance_metadata": {
                    "leader": "172.24.4.152:7848",
                    "raftGroupMember": [

```



```

    },
    "configAbility": {
      "supportRemoteMetrics": false
    },
    "namingAbility": {
      "supportJraft": true
    }
  },
  "grpcReportEnabled": true
}
]
}

```

5.3 查询当前节点健康状态

- 接口描述

查询当前ADCC节点健康状态

- 请求方式

GET

- 请求URL

`/{{adcc_contextpath}}/v2/core/cluster/node/self/health`

- 返回数据

参数名	参数类型	描述说明
data	String	当前节点健康状态

节点共有 `STARTING` , `UP` , `SUSPICIOUS` , `DOWN` , `ISOLATION` 五种状态

- 请求示例

```
curl -X GET 'http://127.0.0.1:8848/{{adcc_contextpath}}/v2/core/cluster/node/self/health'
```

- 返回示例

```

{
  "code": 0,
  "message": "success",
  "data": "UP"
}

```

5.4 切换集群寻址模式

- 接口描述

切换集群寻址模式

- 请求方式

```
PUT
Content-Type:application/x-www-form-urlencoded
```

- 请求URL

```
/{adcc_contextpath}/v2/core/cluster/lookup
```

- 请求Body

参数名	参数类型	是否必填	描述说明
type	String	是	寻址模式

寻址模式有两种: `file` (文件配置) 和 `address-server` (地址服务器)

- 返回数据

参数名	参数类型	描述
data	boolean	是否执行成功

- 请求示例

```
curl -d 'type=file' \
-X PUT 'http://127.0.0.1:8848/{adcc_contextpath}/v2/core/cluster/lookup'
```

- 返回示例

```
{
  "code": 0,
  "message": "success",
  "data": true
}
```

6 连接负载管理

6.1 查询当前节点客户端连接列表

- 接口描述

查询当前ADCC节点上的客户端连接列表

- 请求方式

GET

- 请求URL

`/ {adcc_contextpath} /v2/core/loader/current`

- 返回数据

参数名	参数类型	描述说明
traced	Boolean	是否监控
abilityTable	Map	能力表
metaInfo	Object	元信息
connected	Integer	是否连接
labels	Map	标签

- 请求示例

```
curl -X GET 'http://localhost:8848/{adcc_contextpath}/v2/core/loader/current'
```

- 返回示例

```
{
  "1705989257929_172.24.4.152_42806": {
    "traced": false,
    "abilityTable": {
    },
    "metaInfo": {
      "connectType": "GRPC",
      "clientId": "172.24.4.152",
      "localPort": 9849,
      "version": "Nacos-Java-Client:v1.0.230",
      "connectionId": "1705989257929_172.24.4.152_42806",
      "createTime": "2024-01-23T13:54:17.944+08:00",
      "lastActiveTime": 1706088386309,
      "appName": "-",
      "tenant": null,
      "labels": {

```

```
        "source": "cluster",
        "tls.enable": "false"
    },
    "tag": null,
    "clusterSource": true,
    "sdkSource": false
},
"connected": true,
"labels": {
    "source": "cluster",
    "tls.enable": "false"
}
},
"1705989257993_172.24.4.154_36220": {
    "traced": false,
    "abilityTable": {

    },
    "metaInfo": {
        "connectType": "GRPC",
        "clientIp": "172.24.4.154",
        "localPort": 9849,
        "version": "Nacos-Java-Client:v1.0.230",
        "connectionId": "1705989257993_172.24.4.154_36220",
        "createTime": "2024-01-23T13:54:18.002+08:00",
        "lastActiveTime": 1706088389756,
        "appName": "-",
        "tenant": null,
        "labels": {
            "source": "cluster",
            "tls.enable": "false"
        },
        "tag": null,
        "clusterSource": true,
        "sdkSource": false
    },
    "connected": true,
    "labels": {
        "source": "cluster",
        "tls.enable": "false"
    }
}
}
```

6.2 重新加载当前节点客户端连接数量

- 接口描述

重新加载当前 Nacos 节点的客户端连接数量

- 请求方式

GET

- 请求URL

```
{adcc_contextpath}/v2/core/loader/current/reloadCurrent
```

- 请求参数

参数名	参数类型	是否必填	描述说明
count	Integer	是	连接数量
redirectAddress	String	否	重定向地址

- 返回数据

参数名	参数类型	描述
data	String	是否执行成功

- 请求示例

```
curl -X GET 'http://localhost:8848/{adcc_contextpath}/v2/core/loader/reloadCurrent?count=1&redirectAddress=127.0.0.1:8848'
```

- 返回示例

```
success
```

6.3 智能平衡集群节点的连接数

- 接口描述

智能平衡 Nacos 集群中所有节点的客户端连接数

- 请求方式

GET

- 请求URL

```
{adcc_contextpath}/v2/core/loader/current/smartReloadCluster
```

- 请求参数

参数名	参数类型	是否必填	描述说明
loaderFactor	Float	否	加载因子, 默认加载因子0.1, 每个节点的SDK数量 $(1-loaderFactor) * avg \sim (1+loaderFactor) * avg$
force	String	否	是否强制

- 返回数据

参数名	参数类型	描述
data	String	是否执行成功

- 请求示例

```
curl -X GET 'http://localhost:8848/{adcc_contextpath}/v2/core/loader/smartReloadCluster?
loaderFactor=1'
```

- 返回示例

```
Ok
```

6.4 重置指定客户端的连接

- 接口描述

根据 SDK 连接 ID 发送连接重置请求

- 请求方式

```
GET
```

- 请求URL

```
{adcc_contextpath}/v2/core/loader/current/reloadClient
```

- 请求参数

参数名	参数类型	是否必填	描述说明
connectionId	String	是	连接ID
redirectAddress	String	否	重置地址

- 返回数据

参数名	参数类型	描述
data	String	是否执行成功

- 请求示例

```
curl -X GET 'http://localhost:8848/{adcc_contextpath}/v2/core/loader/reloadClient?
connectionId=1&redirectAddress=127.0.0.1:8848'
```

- 返回示例

```
success
```

6.5 获取集群的 SDK 指标

- 接口描述

获取 Nacos 集群中所有的SDK指标

- 请求方式

GET

- 请求URL

`{adcc_contextpath}/v2/core/loader/cluster`

- 返回数据

参数名	参数类型	描述
total	Integer	当前集群节点数
min	Integer	最小负载值
avg	Integer	平均负载值
max	Integer	最大负载值
memberCount	Integer	当前节点的成员数
metricsCount	Integer	负载信息数量
threshold	Float	负载阈值。阈值的计算公式为：平均负载值 * 1.1
detail	List	包含每个节点的详细负载信息
detail.address	String	节点地址
detail.metric	Map<String,String>	指标信息
completed	Boolean	表示是否已完成负载信息的收集，如果为 true，则表示所有节点的负载信息均已收集，否则为 false

- 请求示例

```
curl -X GET 'http://localhost:8848/{adcc_contextpath}/v2/core/loader/cluster'
```

- 返回示例

```
{
  "total": 1,
  "min": 0,
  "avg": 0,
  "max": 1,
  "memberCount": 3,
  "metricsCount": 3,
  "threshold": 0.0,
  "detail": [
    {
      "address": "172.24.4.152:8848",
      "metric": {
        "load": "0.03",
        "sdkConCount": "0",
        "cpu": "0.008567107",
```

```
        "conCount": "2"
    }
},
{
    "address": "172.24.4.154:8848",
    "metric": {
        "load": "0.0",
        "sdkConCount": "0",
        "cpu": "0.0026181098",
        "conCount": "2"
    }
},
{
    "address": "172.24.4.163:8848",
    "metric": {
        "load": "0.0",
        "sdkConCount": "1",
        "cpu": "0.0044228476",
        "conCount": "3"
    }
}
],
"completed": true
}
```

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

