# 应用升级集群API(ProtocolBuffer 3) V1.1

1. 环境与配置
   1. 测试设备

代理机10.2.1.235:50052（供客户端连接）

RALT机：10.2.1.240 10.2.1.114

* 1. 依赖包

grpc、protobuf

* 1. RALT配置

开启缓存功能

开启缓存操作功能（myCI）

需要添加本机IP为RALT会员，否则无法通过http接口操作缓存

1. RALT状态及统计接口
   1. 获取ralt统计信息

|  |  |
| --- | --- |
| 功能 | 获取ralt统计信息 |
| 调用方法 | getRaltStats |
| 传入参数 | message GetRaltStatsReq {  string ip\_addr = 1; //升级平台ip地址，值为空则返回集群统计值  } |
| 返回参数 | message GetRaltStatsRsp{  uint32 start\_time = 1; //开始时间  uint32 end\_time = 2; //结束时间  uint32 logs\_space\_used\_mb = 3; //日志使用大小(单位mb)  uint32 logs\_space\_total\_mb = 4; //日志空间大小(单位mb)  uint32 flow\_incoming\_requests = 5; //客户端请求数（浏览器-->ATS）  uint32 flow\_incoming\_responses = 6; //ATS响应数（ATS-->浏览器）  uint32 flow\_total\_client\_connections\_ipv4 = 7; //ipv4客户端连接数  uint32 flow\_total\_client\_connections\_ipv6 = 8; //ipv6客户端连接数  uint32 flow\_total\_server\_connections = 9; //回源连接数  uint64 cache\_used\_mb = 10; //缓存使用大小(单位mb)  uint64 cache\_total\_mb = 11; //缓存空间大小(单位mb)  uint32 cache\_total\_hits = 12; //缓存命中总数  float cache\_hit\_ratio = 13; //缓存命中率  uint32 hostdb\_total\_hits = 14; //DNS缓存命中总数  float hostdb\_hit\_ratio = 15; //DNS缓存命中率  } |

* 1. 根据输入统计字段名称，输出统计响应字段值

|  |  |
| --- | --- |
| 功能 | 根据输入统计字段名称，输出统计响应字段值 |
| 调用方法 | getStatsField |
| 传入参数 | message GetStatsFieldReq{  string ip\_addr = 1; //升级平台ip地址，该字段不能为空  string field\_name = 2; //监控类字段名称  } |
| 返回参数 | message GetStatsFieldRsp{  string value = 1;//监控类字段值  } |

* 1. RALT首页展示

|  |  |
| --- | --- |
| 功能 | RALT首页展示 |
| 调用方法 | getHomePageData |
| 传入参数 | message HomePageReq {  uint32 type = 1; //客户端类型，目前网管约定为1  } |
| 返回参数 | message HomePageRsp {  uint32 domain\_num = 1;//会员数  uint32 cluster\_device = 2;//集群设备数  uint64 app\_transfor\_bytes = 3;//应用转换流量  uint32 app\_transfor\_connections = 4;应用升级总连接数  uint32 cache\_total\_hits = 5;//缓存命中数  uint64 cache\_bytes\_total = 6;//缓存总大小，单位为bytes  uint64 cache\_bytes\_used = 7;//已使用缓存大小，单位为bytes  uint32 cache\_num\_docs = 8;缓存文档数  } |

* 1. 查询缓存信息

|  |  |
| --- | --- |
| 功能 | 查询缓存信息 |
| 调用方法 | showCacheData |
| 传入参数 | message CacheLookUpReq {  string ip\_addr = 1; //升级平台ip地址，值为空则返回集群统计值  } |
| 返回参数 | message CacheResult {  uint32 cache\_total\_hits = 1;//缓存命中总数  uint64 bytes\_total = 2;//缓存总大小，单位为bytes  uint64 bytes\_used = 3;//已使用缓存大小，单位为bytes  uint32 num\_docs = 4;缓存驻留文档数  float cache\_hit\_ratio = 5;//缓存命中率  } |

* 1. 查询流量统计

|  |  |
| --- | --- |
| 功能 | 查询流量统计信息 |
| 调用方法 | showFlowStatData |
| 传入参数 | message FlowStatLookUpReq{  string ip\_addr = 1; //升级平台ip地址，值为空则返回集群统计值  } |
| 返回参数 | message FlowResult{  uint32 user\_agent\_current\_connections\_count = 1; //用户代理连接数  uint32 origin\_server\_current\_connections\_count = 2;//源端总连接数  uint32 completed\_requests = 3;//已完成请求数  uint32 total\_incoming\_connections = 4;//进入应用升级平台总连接数  uint32 incoming\_requests = 5;//进入应用升级平台请求数  uint32 outgoing\_requests = 6;//应用升级平台回源请求数  uint32 incoming\_responses = 7;// 应用升级平台响应数  uint32 total\_client\_connections = 8;//客户端总连接数  uint32 total\_client\_connections\_ipv4 = 9;//ipv4客户端总连接数  uint32 total\_client\_connections\_ipv6 = 10;//ipv6客户端总连接数  uint32 total\_server\_connections = 11;//服务端总连接数  uint64 user\_agent\_total\_bytes = 12;//用户代理端总连接数，单位为bytes  uint64 origin\_server\_total\_bytes = 13;//源端流量大小，单位为bytes  float bandwidth\_hit\_ratio = 14;//流量命中率  } |

* 1. 查询RALT日志状态

|  |  |
| --- | --- |
| 功能 | 查询RALT日志状态 |
| 调用方法 | showLogInfoData |
| 传入参数 | message LogInfoLookUpReq{  string ip\_addr = 1; //升级平台ip地址，值为空则返回集群统计值  } |
| 返回参数 | message LogResult{  uint32 log\_files\_space\_mb\_used = 1;//已使用日志文件大小，单位为mb  uint32 max\_space\_mb\_for\_logs = 2; //日志的存储空间的大小，单位为mb  } |

* 1. 获取RALT日志内容

|  |  |
| --- | --- |
| 功能 | 获取RALT日志内容 |
| 调用方法 | getRaltLogs |
| 传入参数 | message GetRaltLogsReq{  string ip\_addr = 1; //升级平台ip地址  } |
| 返回参数 | message RaltLogs{  bytes logs = 1; //ralt日志  } |

* 1. 获取所有缓存URL（待完善，具体需要与java组讨论）

|  |  |
| --- | --- |
| 功能 | 获取所有缓存URL（待优化，只是简单添加接口） |
| 调用方法 | getCacheUrl |
| 传入参数 | message GetCacheUrlReq{  string ip\_addr = 1; //升级平台ip地址  } |
| 返回参数 | message GetCacheUrlRsp{  string all\_url = 1; //所有缓存URL  } |

* 1. 判断URL是否已被缓存（待完善，具体需要与java组讨论）

|  |  |
| --- | --- |
| 功能 | 判断URL是否已被缓存（待优化，只是简单添加接口） |
| 调用方法 | isUrlInCache |
| 传入参数 | message IsUrlInCacheReq{  string ip\_addr = 1; //升级平台ip地址  string url = 2; //待查询缓存url  } |
| 返回参数 | message IsUrlInCacheRsp{  uint32 result = 1; //返回结果，0：已缓存 1：未缓存  } |

1. RALT配置接口
   1. 获取基本配置信息

|  |  |
| --- | --- |
| 功能 | 获取基本配置信息 |
| 调用方法 | getBasicConfig |
| 传入参数 | message GetBasicConfigReq {  string ip\_addr = 1; //升级平台ip地址  } |
| 返回参数 | message BasicConfig {  uint32 logging\_enabled = 1; //取值：0-不启用日志； 1-开启错误； 2-开启事务日志； 3-错误日志和事务日志一起开启  uint32 max\_space\_mb\_for\_logs = 2; //日志最大磁盘空间（单位为mb）  uint32 rolling\_enabled = 3; //日志是否备份 0-不开启； 1-开启日志（根据时间间隔备份）；2-开启日志（根据日志文件大小备份）  string server\_ports = 4; //ralt服务端口，默认值 8080 8080:ipv6  uint32 storage\_cache\_size = 5; //缓存磁盘大小，单位为mb，默认值 250（最小值为128）  uint32 http\_cache\_enabled = 6; //缓存开关，0：不开启 1：开启  uint32 connections\_throttle = 7; //最大并发量  string ip\_resolve = 8; //回源网络类型  }  message GetBasicConfigRsp {  BasicConfig basic\_config = 1;  } |

* 1. 修改基本配置信息

|  |  |
| --- | --- |
| 功能 | 修改基本配置信息 |
| 调用方法 | setBasicConfig |
| 传入参数 | message BasicConfig {  uint32 logging\_enabled = 1; //取值：0-不启用日志； 1-开启错误； 2-开启事务日志； 3-错误日志和事务日志一起开启  uint32 max\_space\_mb\_for\_logs = 2; //日志最大磁盘空间（单位为mb）  uint32 rolling\_enabled = 3; //日志是否备份 0-不开启； 1-开启日志（根据时间间隔备份）；2-开启日志（根据日志文件大小备份）  string server\_ports = 4; //ralt服务端口，默认值 8080 8080:ipv6  uint32 storage\_cache\_size = 5; //缓存磁盘大小，单位为M，默认值 250（最小值为128）  }  message SetBasicConfigReq {  string ip\_addr = 1; //升级平台ip地址  BasicConfig basic\_config = 2;  } |
| 返回参数 | message SetBasicConfig {  uint32 result = 1; //0：成功，1：失败  } |

* 1. 获取所有会员信息

|  |  |
| --- | --- |
| 功能 | 获取所有会员信息 |
| 调用方法 | getAllDomain |
| 传入参数 | message GetAllDomainReq{  string ip\_addr = 1; //升级平台ip地址  uint32 page\_size = 2; //每页会员数  uint32 page\_number = 3; //页数  } |
| 返回参数 | enum DomainType{  enum\_member\_domain = 0; //ralt\_member\_domain类型会员  enum\_subs\_domain = 1;//subs会员  }  message Domain{  DomainType type = 1; //会员类别  string domain\_str = 2; //会员域名  string append\_or\_replace\_str = 3; //追加或替换字符串(ralt\_member\_domain对应追加，ralt\_subs\_domais对应替换)  string port = 4; //会员端口  }  message GetAllDomainRsp{  repeated Domain domain = 1;  uint32 domain\_total = 2; //会员总数  } |

* 1. 更新会员信息（增、删、改）

|  |  |
| --- | --- |
| 功能 | 更新会员信息（增、删、改） |
| 调用方法 | updateDomain |
| 传入参数 | enum DomainType{  enum\_member\_domain = 0; //ralt\_member\_domain类型会员  enum\_subs\_domain = 1;// ralt\_subs\_domais类型会员  }  enum DomainType{  enum\_member\_domain = 0; // ralt\_member\_domain类型会员  enum\_subs\_domain = 1; // ralt\_subs\_domais类型会员  enum\_subs\_domain\_6to4 = 2; //ralt\_subs\_domain\_6to4类型会员  enum\_subs\_domain\_4to6 = 3; //ralt\_subs\_domain\_4to6类型会员  }  message Domain{  DomainType type = 1; //会员类别  string domain\_str = 2; //会员域名  string append\_or\_replace\_str = 3; //追加或替换字符串(ralt\_member\_domain对应追加，ralt\_subs\_domais对应替换)  string port = 4; //会员端口  }  message UpdateDomainReq{  string ip\_addr = 1; //升级平台ip地址  repeated Domain domain = 2;  } |
| 返回参数 | message UpdateDomainRsp{  uint32 result = 1;// 0：成功，1：失败  } |

* 1. 根据会员名称或者转换后域名获取该会员信息

|  |  |
| --- | --- |
| 功能 | 根据会员名称或者转换后域名获取该会员信息 |
| 调用方法 | getDomain |
| 传入参数 | message GetDomainReq{  string ip\_addr = 1; //升级平台地址  string domain\_str = 2; //根据域名模糊查询  string transformed\_domain = 3; //根据转换域名模糊查询  uint32 page\_size = 4; //每页会员数  uint32 page\_number = 5; //页数  } |
| 返回参数 | enum DomainType{  enum\_member\_domain = 0; //ralt\_member\_domain类型会员  enum\_subs\_domain = 1;// ralt\_subs\_domais类型会员  }  message Domain{  DomainType type = 1; //会员类别  string domain\_str = 2; //会员域名  string append\_or\_replace\_str = 3; //追加或替换字符串(ralt\_member\_domain对应追加，ralt\_subs\_domais对应替换)  string port = 4; //会员端口  }  message GetDomainRsp{  repeated Domain domain = 1;  uint32 domain\_total = 2; //会员总数  } |

* 1. 添加一条会员记录

|  |  |
| --- | --- |
| 功能 | 添加一条会员记录 |
| 调用方法 | addDomain |
| 传入参数 | enum DomainType{  enum\_member\_domain = 0; //ralt\_member\_domain类型会员  enum\_subs\_domain = 1;// ralt\_subs\_domais类型会员  }  message Domain{  DomainType type = 1; //会员类别  string domain\_str = 2; //会员域名  string append\_or\_replace\_str = 3; //追加或替换字符串(ralt\_member\_domain对应追加，ralt\_subs\_domais对应替换)  string port = 4; //会员端口  }  message AddDomainReq{  string ip\_addr = 1; //升级平台地址  Domain domain = 2;  } |
| 返回参数 | message AddDomainRsp {  uint32 result = 1; //0：成功，1：失败  } |

* 1. 删除一条会员记录

|  |  |
| --- | --- |
| 功能 | 删除一条会员记录 |
| 调用方法 | deleteDomain |
| 传入参数 | message DeleteDomainReq{  string ip\_addr = 1; //升级平台地址  string domain\_str = 2; //会员域名  } |
| 返回参数 | message DeleteDomainRsp {  uint32 result = 1; //0：成功，1：失败  } |

* 1. 获取RALT杂项配置信息

|  |  |
| --- | --- |
| 功能 | 获取RALT杂项配置信息 |
| 调用方法 | getMisc |
| 传入参数 | message GetMiscReq{  string ip\_addr = 1; //升级平台地址  } |
| 返回参数 | enum MiscSwitch{  switch\_off = 0;//开关关闭  switch\_on = 1;//开关打开  }  message GetMiscRsp{  MiscSwitch ralt\_filter\_type\_default = 1;//过滤类别开关  } |

* 1. 修改RALT杂项配置信息

|  |  |
| --- | --- |
| 功能 | 修改RALT杂项配置信息 |
| 调用方法 | modMisc |
| 传入参数 | enum MiscKey{  enum\_ralt\_filter\_type\_default = 0;  }  enum MiscSwitch{  switch\_off = 0;  switch\_on = 1;  }  message Misc{  MiscKey key = 1;  MiscSwitch value = 2;  }  message ModMiscOpReq{  string ip\_addr = 1; //升级平台地址  Misc misc = 2;  } |
| 返回参数 | message ModMiscOpRsp{  uint32 result = 1;// 0：成功，1：失败  } |

* 1. 获取所有RALT规则

|  |  |
| --- | --- |
| 功能 | 获取所有RALT规则 |
| 调用方法 | getRule |
| 传入参数 | message GetRuleReq{  string ip\_addr = 1; //应用升级平台地址  } |
| 返回参数 | //ralt rule  enum RuleType{ //类别  enum\_begin = 0; //该字段没有用，因为proto3的enum必须从0开始，此处是为了保持与ralt的规则类型一直  enum\_top\_level\_domain = 1;  enum\_hex\_domain = 2;  enum\_js\_buildin\_var = 3;  enum\_ipv4\_addr = 4;  enum\_email\_atsign = 5;  enum\_no\_modify = 6;  enum\_unhandle\_protocol = 7;  enum\_assist\_char = 8;  enum\_subs\_string = 9;  enum\_rsubs\_string = 10;  enum\_ipv6\_addr = 11;  enum\_just\_replace = 12;  enum\_just\_restore = 13;  }  message Rule{  RuleType type = 1; //类别  string search = 2; //查找字串  string replace = 3; //替换字串  string append = 4; //追加字串  string ralt\_domain = 5; //对应会员关联列表的开关  }  message GetRuleRsp{  repeated Rule rule = 1;  } |

* 1. 更新RALT会员规则（增、删、改）

|  |  |
| --- | --- |
| 功能 | 更新RALT会员规则（增、删、改） |
| 调用方法 | updateRule |
| 传入参数 | //ralt rule  enum RuleType{  enum\_begin = 0; //该字段没有用，因为proto3的enum必须从0开始，此处是为了保持与ralt的规则类型一直  enum\_top\_level\_domain = 1;  enum\_hex\_domain = 2;  enum\_js\_buildin\_var = 3;  enum\_ipv4\_addr = 4;  enum\_email\_atsign = 5;  enum\_no\_modify = 6;  enum\_unhandle\_protocol = 7;  enum\_assist\_char = 8;  enum\_subs\_string = 9;  enum\_rsubs\_string = 10;  enum\_ipv6\_addr = 11;  enum\_just\_replace = 12;  enum\_just\_restore = 13;  }  message Rule{  RuleType type = 1; //类别  string search = 2; //查找字串  string replace = 3; //替换字串  string append = 4; //追加字串  string ralt\_domain = 5; //对应会员关联列表的开关  }  message UpdateRuleReq{  string ip\_addr = 1; //应用升级平台地址  repeated Rule rule = 2;//更新的规则  } |
| 返回参数 | message UpdateRuleRsp{  uint32 result = 1; //返回结果，0-操作成功 1-操作失败  } |

1. RALT管理者操作接口
   1. 获取应用升级平台状态

|  |  |
| --- | --- |
| 功能 | 获取应用升级平台状态 |
| 调用方法 | getRaltStatus |
| 传入参数 | message RaltStatusReq {  string ip\_addr = 1; //升级平台ip地址，为空则返回集群内所有升级平台状态  } |
| 返回参数 | message RaltStatus {  string ip\_addr = 1; //ralt平台地址  uint32 status = 2;//0：平台已启动，1：平台未启动  } |

* 1. 管理员操作命令

|  |  |
| --- | --- |
| 功能 | 管理员操作命令 |
| 调用方法 | execCmd |
| 传入参数 | enum CommandType{  ralt\_start = 0; //启动应用升级  ralt\_stop = 1;//停止应用升级  ralt\_restart = 2;//重启应用升级  cache\_clear = 3;//清除缓存  reload\_config = 4;//重新加载配置  }  message ExecCmdReq {  string ip\_addr = 1; //应用升级平台地址  CommandType cmd = 1; //操作类型  } |
| 返回参数 | message ExecCmdRsp {  uint32 result = 1; //0：成功，1：失败  } |