# global

本节描述了一些全局基本的JSON-RPC命令。全局命令的客户端由IClient组件实现。

## Illustration

### GlobalErrorCode

略

### ErrorCode

通用错误码定义在Component组件的头文件Defs.h和ProtoDefs.h，一个为组件级，一个为协议、业务层级。

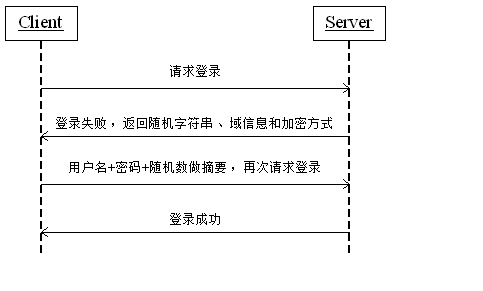
|  |  |  |
| --- | --- | --- |
| **参数** | **说明** | **举例** |
| 0x10030000 | 组件接口没有找到(废弃，见0x10070002) | InterfaceNotFound |
| 0x10030001 | 组件实现类没有找到 | ClassNotFound |
| 0x10030002 | 需要的客户端组件没有找到 | ClientNotFound |
| 0x10030003 | 客户端实例不存在 | ClientNotExist |
| 0x10030004 | 没有登录 | NotLoginYet |
| 0x10030005 | 非阻塞登录，正在尝试 | TryingLogin |
| 0x10030006 | 用户名无效 | UserNameNotValid |
| 0x10030007 | 密码无效 | PasswordNotValid |
| 0x10030008 | 地址被屏蔽了 | PeerHasBeenBlocked |
| 0x10030009 | 地址在黑名单中 | PeerInBlackList |
| 0x1003000a | 用户名已经在使用，而且不能复用 | UserNameHasBeenUsed |
| 0x1003000b | 超出了最大用户连接数 | TooManyConnections |
| 0x1003000c | 没有对应的操作权限 | HasNoAuthority |
| 0x1003000d | 组件工厂创建对象失败 | CreateObjectFailed |
| 0x1003000e | 组件工厂获取实例失败 | GetInstanceFailed |
| 0x1003000f | 登录质询，需要根据返回的随机字符串再次登录 | LoginChallenge |
| 0x10030010 | 组件接口方法没有实现 | UnimplementedOperation |
| 0x10030011 | 帐号被锁定 | AccountHasBeenBlocked |
| 0x10030012 | 登陆时网络连接失败 | LoginConnectFailed |
| 0x10030013 | 登陆连接成功，但收响应超时 | LoginTimeout |
| 0x10030014 | 登陆失败，原因未知 | LoginFailure |
| 0x10030015 | 登陆失败，用户名或密码错误(替代6、7错误码) | UserOrPwdNotValid |
| 0x10030016 | 登陆失败，未初始化无用户 | HasNotInit |
| 0x10030017 | 登陆失败，admin用户需要强制修改密码 | NeedModifyAdminPwd |
| 0x10030018 | 登陆连接超时 | LoginConnectTimeout |
| 0x10030019 | 登陆失败，密码有效期失效 | PasswordOverdue |
| 0x1003001a | 登陆失败，提示用户登录方式不对，需要切换登录方式 | Login Failed – The login method is incorrect. |
| 0x1003001b | 服务端返回消息格式错误(客户端设置) | Format Error – The message format received from the server is incorrect. |
| 0x1003001c | 登录失败，接入协议不匹配, 如以GB28181登录加强版的GB35114端 | Login Failed – The protocol version mismatched. |
| 0x1003001d | 登录失败，登录受限，受限条件有限制IP、登录时间 | Login Failed – Not allowed, maybe IP range or login time limitations. |
| 0x1003001e | 登录失败，设备不支持高安全等级登录 | Login Failed – The safety level is not allowed. |
| 0x1003001f | 组件接口操作不在产品规格内，被拦截 | componentErrorSCIUnSupportedOperation |
| 0x10030020 | 内部异常引起协议解析错误 Internal error when parsing protocol packages | ParseFailed |
| 0x10030021 | 组件接口操作超过设备限定能力，设备资源不足 | componentErrorResourceLimit |
| 0x10030022 | 账号已登录 | componentErrorUserHasLogined |
| 0x10030023 | 版本不对，无法登陆 | componentErrorInvalidVersion |
| 0x10030024 | 找不到网络主机 | componentErrorNotFindNetworkHost |
| 0x10030025 | 服务端返回消息错误 | componentErrorServerReturnMessage |
| 0x10030026 | 设备不匹配 | componentErrorDeviceTypeMismatch |
| 0x10030027 | 域名解析失败 | componentErrorDomainNameAnalyse |
| 0x10030028 | 视频已加密 | componentErrorVideoEncrypted |
| 0x10030101 | 设备的状态机，在C状态时，不接受B状态时才有的业务请求。 | Bad Request |
| 0x10030102 | 参数值，数量、类型、范围等不符合要求 | Invalid Arguments – The parameter is not correct, maybe the value, quantity, types, range error. |
| 0x10030103 | 正在处理业务，不接受重复业务请求。 | In Progress – Please wait. |
| 0x10030104 | 业务处理过程中出现未知错误 | UnKnown Error |
| 0x10030105 | 在处理业务请求过程中，资源不足 | No Resource – The hardware resource is not enough. |
| 0x10030106 | 当前不支持该业务 | UnSupported – Such service is not supported on the device. |
| 0x10030107 | 业务处理超时 | Timeout |
| 0x10030108 | 下发的业务请求过多 | Too Many Requests – The device can not handle too much requests simultaneously. |
| 0x10030109 | 获取了过多的数据，设备无法处理 | Requested Too Much Data at one time. |
| 0x10030110 | 在业务批量执行过程中有错误发生 | Batch Process Error |
| 0x10030111 | 某种原因，业务被取消执行 | Operation Cancelled |
| 0x10030201 | 设备未准备就绪，不接受业务的处理 | Device WarmUP - Please wait for a while. |
| 0x10030202 | 设备型号不对或定制原因，无法进一步处理 | Device Invalid |
| 0x10030203 | 无法获取到设备状态信息 | Device Unavailable – Can not get the device status. |

### 二次认证与鉴权

略

## RPC

### login(first)



本协议更恰当的命名用negotiate或类似词汇，两次login的参数、返回都不相同却使用相同的方法名，容易导致代码不清晰。目前可以根据是否带authorityType、nc、cnonce等参数来进行区分，第一次不应带，第二次应该带。

关于加密方式和登录鉴权方式，以及具体的密码计算方式，参考《大华信息交换格式-内部信息分册.docx》中的“加密与登录鉴权说明”章节。

#### Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **权限** | admin权限 | | | |
| **参数** | **类型** | **必选** | **说明** | **举例** |
| method | char[] | 是 | "global.login" 和服务器协商登陆信息 | "global.login" |
| params | object | 是 | 接口参数 |  |
| +userName | char[128] | 是 | 用户名 | "admin" |
| +password | char[32] | 否 | 首次登陆无意义 | "" |
| +clientType | char[] | 是 | 客户端类型 参见客户类型ClientType定义 | "Dahua3.0" |
| +loginType | enumchar[32] | 否 | 登录方式 enumchar[32]{ /\*为空表示"Direct"\*/ "Direct": 点对点登录 "CMS": 通过CMS服务器登录 "LDAP": 通过LDAP服务器登录 "ActiveDirectory": 通过AD服务器登录 "Loopback": 本地环回登录(不校验密码，校验IP)  "TemporaryToken":web跳转免密码通过临时Token登陆  "Fingerprint":指纹登陆 } | "Direct" |
| +deviceId | char[64] | 否 | 设备ID，唯一标识平台管理下的设备，作为代理级联定位目标设备时使用 | "xxx:xxx" |
| +stochasticId | int | 否 | 用于防盗版验证的随机数，仅第一次请求时增加此值 | 123141 |
| id | int | 空 | 请求ID号，返回时需要对应 | 1 |
| session | uint32 | 是 | 服务器分配的id | 81375 |
| **示例** | | | | |
| {  "method" : "global.login",  "params" : {  "userName" : "admin",  "password" : "",  "clientType" : "Dahua3.0",  "loginType" : "Direct",  "deviceId" : "xxx:xxx",  "stochasticId" :123141  },  "id" : 1  "session" : 81375, } | | | | |

#### Response Success

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **类型** | **必选** | **说明** | **举例** |
| result | bool | 是 | 返回值，首次一定为false | false |
| error | object | 空 |  |  |
| +code | int | 空 | 错误码，要求发起质询时返回0x1003000f | 268632079 |
| +message | char[128] | 空 | 错误消息 | "Component error: login challenge!" |
| params | object | 是 | 返回参数 |  |
| +realm | char[] | 空 | 域信息 | "Login to XXXXXX" |
| +random | char[] | 空 | 随机字符串，  error.code为0x1003000f时表示服务器返回的随机字符串，发起质询，客户端下次需要根据随机字符串和原始密码生成摘要，该摘要作为密码再次登录。  相当于http digest的nonce字段。 | "57231468" |
| +encryption | char[] | 空 | 加密方式，详见内部文档  如果有authorityType字段，则本字段可以忽略 | "Default" |
| +authorityType | char[32][] | 否 | 服务端支持的认证鉴权方式列表，客户端应遵从安全等级从高到低选用。  鉴权方式详见内部文档  由于MD5不安全，新设备应该优先使用SHA256以上的鉴权方式 | ["DigestSHA2\_256 ","Default"] |
| +qop | char[16][] | 否 | 同标准Http Digest鉴权中的qop（quality of protection），当前仅支持取值"auth"  （DigestSHA2\_256时必填） | [ "auth" ] |
| +opaque | char[33] | 否 | 同标准Http Digest鉴权中的opaque，由服务端随机生成，客户端请求时需要携带。建议使用16进制字符串  （DigestSHA2\_256时必填） | "828cb5bf79617f3c1c9083ccb0598fdf" |
| +authorization | char[] | 空 | 防盗版验证信息，第一次登陆回复即可 | "adbadfasd" |
| ~~+mac~~ | ~~char[]~~ | ~~否~~ | ~~应对欧盟GDPR法案，删除非必须字段 设备eth0的mac地址,无分隔符的12个字节大写字符串~~ | ~~"52544CFA0001"~~ |
| session | int | 空 | 会话编号 第一次登陆时返回会话的编号，以后每次请求和响应都应该包含该编号 ，参见 | 454651 |
| id | int | 空 | 回应ID号，和请求时对应 | 1 |
| **示例** | | | | |
| {  "result" : false,  "error" : {  "code" : 268632079,  "message" : "Component error: login challenge!"  },  "params" : {  "realm" : "Login to XXXXXX",  "random" : "57231468",  "encryption" : "Default",  "authorityType" : [ "DigestSHA2\_256 ","Default" ],  "qop" : [ "auth" ],  "opaque" : "828cb5bf79617f3c1c9083ccb0598fdf",  "authorization" : "adbadfasd",  },  "session" : 454651,  "id" : 1 } | | | | |

### login(second)

#### Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **权限** | admin权限 | | | |
| **参数** | **类型** | **必选** | **说明** | **举例** |
| method | char[] | 是 | "global.login" 登陆到服务器 | "global.login" |
| params | object | 是 | 接口参数 |  |
| +userName | char[128] | 是 | 用户名 | "admin" |
| +password | char[129] | 是 | 密码  loginType为"TemporaryToken"时该字段值为使用临时Token进行二次鉴权。  loginType为"Fingerprint"时该字段值为使用指纹id进行二次鉴权 | "A9DEC16312A4465E97D044EF78D695B9" |
| +clientType | char[] | 是 | 客户端类型  参见客户类型ClientType说明 | "Dahua3.0" |
| +loginType | char[32] | 否 | 登录方式  取值参见login(first)的loginType字段说明 | "Direct" |
| +ipAddr | char[48] | 否 | 客户端ip地址 | "127.0.0.1" |
| +authorityType | char[32] | 是 | 鉴权方式，详见内部文档  应该从login(first)的设备应答中，选择设备支持的列表中安全性高的鉴权方式，例如DigestSHA2\_256以上 | "Default" |
| +passwordType | char[32] | 否 | 加密方式(SDK未填写)，详见内部文档  （仅供兼容使用，新的鉴权方式不再使用此字段） | "OldDigest" |
| +realm | char[] | 否 | 域信息，和login(first)返回的值一致  （可不填，但为了解释和HttpDigest相似时建议填） | "Login to XXXXXX" |
| +random | char[] | 否 | 随机字符串，相当于http digest的nonce字段。和login(first)返回的值一致  （可不填，但为了解释和HttpDigest相似时建议填） | "57231468" |
| +qop | char[16] | 否 | 同标准Http Digest鉴权中的qop（quality of protection）。应该从login(first)返回的列表中选择一个，但当前仅支持取值"auth"  （可不填，但为了解释和HttpDigest相似时建议填） | "auth" |
| +nc | char[9] | 否 | 同标准Http Digest鉴权中的nc（nonce count"）,16进制字符串  （authorityType=DigestSHA2\_256时必填） | "00000001" |
| +cnonce | char[33] | 否 | 同标准Http Digest鉴权中的cnonce,16进制字符串  （authorityType=DigestSHA2\_256时必填） | "dcd98b7102dd2f0e8b11d0f600bfb0c0" |
| +request-uri | char[] | 否 | 请求资源路径，大华二、三协议默认填"/"，另外Http Digest鉴权中的Method固定取POST  （authorityType=DigestSHA2\_256时必填） | "/" |
| +opaque | char[33] | 否 | 同标准Http Digest鉴权中的opaque，和login(first)返回的值一致  （authorityType=DigestSHA2\_256时必填） | "828cb5bf79617f3c1c9083ccb0598fdf" |
| +authorityInfo | char[] | 空 | 附加鉴权信息(SDK未填写) "authorityType" 为 "Default" 时，其值为空。 "authorityType" 为 "HttpDigest" 时，其值为：nc:cnonce:qop:ha2。  其他情况时其值为空，详见内部文档 | "xxx" |
| +proxyToken | char[] | 空 | 代理令牌，用于代理服务器鉴权使用 | "xxx" |
| id | int | 空 | 请求ID号，返回时需要对应 | 2 |
| session | int | 空 | 本次会话的编号 登陆后所有请求都包含此编号，为节约篇幅，余下暂时略去。 | 454651 |
| **示例** | | | | |
| {  "method" : "global.login",  "params" : {  "userName" : "admin",  "password" : "A9DEC16312A4465E97D044EF78D695B9",  "clientType" : "Dahua3.0",  "loginType" : "Direct",  "ipAddr" : "127.0.0.1",  "authorityType" : "Default",  "passwordType" : "OldDigest",  "realm" : "Login to XXXXXX",  "random" : "57231468",  "qop" :"auth",  "nc": "00000001",  "cnonce": "dcd98b7102dd2f0e8b11d0f600bfb0c0",  "request-uri": "/",  "opaque" : "828cb5bf79617f3c1c9083ccb0598fdf",  "authorityInfo" : "xxx",  "proxyToken" : "xxx"  }  "id" : 2,  "session" : 454651 } | | | | |

#### Response Success

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **类型** | **必选** | **说明** | **举例** |
| result | bool | 是 | 返回值 | true |
| params | object | 是 | 登陆失败参数 |  |
| +keepAliveInterval | int | 空 | 保活周期，单位：秒 登陆成功后，由服务端返回给客户端的最大保活周期，如果消息中包含这个字段，则应以小于此值发送保活包；如果没有则按照原先的流程执行 | 30 |
| +isPwdOverdue | bool | 否 | 用户密码是否过期。只有登陆成功后才返回这个字段 | true |
| +remainLoginTimes | uint8 | 否 | 剩余登录次数 登陆失败字段 | 1 |
| +remainLockSecond | uint16 | 否 | 锁定剩余时间，单位秒 登陆失败字段 | 300 |
| id | int | 空 | 回应ID号，和请求时对应 | 2 |
| session | int | 空 | 会话编号 登陆后所有响应都包含此编号 | 454651 |
| **示例** | | | | |
| {  "result" : true,  "params" : {  "keepAliveInterval" : 30,  "isPwdOverdue": true,  "remainLoginTimes" : 1,  "remainLockSecond" : 300  }  "id" : 2,  "session" : 454651 } | | | | |

### logout

#### Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **权限** | admin权限 | | | |
| **参数** | **类型** | **必选** | **说明** | **举例** |
| method | char[] | 是 | "global.logout" 从服务器登出 | "global.logout" |
| params | object/null | 空 | 接口参数 | null |
| id | int | 空 | 请求ID号，返回时需要对应 | 12 |
| session | int | 空 | 本次会话的编号 登陆后所有请求都包含此编号，为节约篇幅，余下暂时略去。 | 454651 |
| **示例** | | | | |
| {  "method" : "global.logout",  "params" : null,  "id" : 12,  "session" : 454651 } | | | | |

#### Response Success

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **类型** | **必选** | **说明** | **举例** |
| result | bool | 是 | 返回值 | true |
| params | object/null | 空 | 返回参数 | null |
| id | int | 空 | 回应ID号，和请求时对应 | 12 |
| session | int | 空 | 会话编号 登陆后所有响应都包含此编号 | 454651 |
| **示例** | | | | |
| {  "result" : true,  "params" : null,  "id" : 12,  "session" : 454651 } | | | | |

### keepAlive

#### Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **权限** | admin权限 | | | |
| **参数** | **类型** | **必选** | **说明** | **举例** |
| method | char[] | 是 | "global.keepAlive" 客户端和服务器之间保活 | "global.keepAlive" |
| params | object | 是 | 接口参数 |  |
| +timeout | int | 空 | 超时时间，以服务端返回数值为准，单位为秒 | 60 |
| +active | bool | 否 | Web客户端是否有操作 解决Web上若干时间无操作后，清除空闲会话的问题。空闲会话是指客户端无操作达到指定时间，服务端清除该会话 如果保活周期内有操作，空闲会话时长清0，无操作则累加空闲会话时长，达到指定时间如30分钟，清除该会话 true该保活周期内Web有操作 false:Web无操作(默认) | true |
| session | uint32 | 是 | 服务器分配的id | 81375 |
| id | uint32 | 是 | 会话id,最小值为1 | 1 |
| **示例** | | | | |
| {  "method" : "global.keepAlive",  "params" : {  "timeout" : 60,  "active" : true  }  "session" : 81375,  "id" : 1 } | | | | |

#### Response Success

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **类型** | **必选** | **说明** | **举例** |
| result | bool | 是 | 返回值 | true |
| params | object | 是 | 接口参数 |  |
| +timeout | int | 是 | 超时时间，若客户端设置超时时间过长则服务端返回默认最大值，若客户端设置超时时间合法则原值返回。如果客户端到下次发送的时间间隔大于这个时间，保活失败，服务器主动踢出用户。 | 60 |
| session | uint32 | 是 | 服务器分配的id | 81375 |
| id | uint32 | 是 | 会话id,最小值为1 | 1 |
| **示例** | | | | |
| {  "result" : true,  "params" : {  "timeout" : 60  }  "session" : 81375,  "id" : 1 } | | | | |