客户端发送请求(都是post方式请求)：

<http://192.168.2.170:5370/>

header: content-type：application/json

tokenid为：1000

# 接口命名简单规则

/get/… :为获取接口

/set/.. :为新增接口

/update/… :为修改接口

/del/… :为删除接口

# 接口回复的参数格式

{

“result”:200,

“errorinfo”:”“,

“data”:

}

“result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；

“errorinfo:” 表示返回的错误信息(成功时显示”成功”)

“data”:接口返回的信息；()

data下：

“para”:, 没有双引号表示int类型；

“para2”:”“, 有双引号表示char类型；

# bs端界面参数数据获取接口

## 1.规约信息获取

|  |  |  |
| --- | --- | --- |
| interface: | /get/prot |  |
| request json: | {  “tokenid”: 1000  } | 获取规约数据 |
| response json: | {  “result”:200,  “errorinfo”:”“,  “data”:  [  {  “id”: ,  “desc”: ““,  “type”:  },  {…},  {…}  ]  } | “errorinfo”:表示返回的错误信息(成功时显示”成功”)，此字段一下接口含义相同  “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data中type表示规约类型，1表示采集规约，2表示转发规约。id/desc代办规约的id号和描述； |

id,desc,type从semodel.db中t002\_prot表获取信息。

## 2.通道设备类型信息获取

|  |  |  |
| --- | --- | --- |
| interface: | /get/comm |  |
| request json: | {  “tokenid”: 1000,  } | 获取通道设备类型数据 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  [ #无则为0  {  “id”: ,  “desc”: ““  },  {…},  {…}  ]  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误  id/desc表示通道设备类型的id和描述； |

id,desc从semodel.db中t001\_comm表获取信息。

# bs端界面系统服务页面接口

## 3.获取命令对照表信息

一般包括pa,pis,cctv通用命令等；

|  |  |  |
| --- | --- | --- |
| interface: | /get/cmdmap |  |
| request json: | {  “tokenid”: 1000,  } | 获取通用命令信息 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “tokenid”:”1000 ,  “data”:  [ #无则为0  {  “eigenno”: ,  “groupno”: ,  “rtuno”:  },  {…},  {…}  ]  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下：eigenno，groupno，rtuno从t032\_cmdmap获取  eigenno:本征号  groupno:组号  rtuno:采集rtu号 |

从semodel中t032\_cmdmap

## 4.新增命令对照表信息

|  |  |  |
| --- | --- | --- |
| interface: | /set/cmdmap |  |
| request json: | {  “tokenid”: 1000,  “eigenno”:，  “groupno”: ,  “rtuno”:  } | 新增通用命令信息，  eigenno，groupno，rtuno是数字；  eigenno:本征号  groupno:组号  rtuno:采集rtu号 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

写入semodel中t032\_cmdmap

## 5.删除命令对照表信息

|  |  |  |
| --- | --- | --- |
| interface: | /del/cmdmap |  |
| request json: | {  “tokenid”: 1000,  “eigenno”:，  “groupno”: ,  } | 删除通用命令信息  eigenno:本征号  groupno:组号 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

删除semodel中t032\_cmdmap表数据；

# bs端界面服务页面信息接口

## 6.rtu服务名信息获取

|  |  |  |
| --- | --- | --- |
| interface: | /get/rtulist |  |
| request json: | {  “tokenid”: 1000,  “rtutype”:  } | 获取类型rtu服务名列表，获取rtu服务名列表，rtutype:0表示采集服务，1表示虚拟点服务，2表示转发服务； |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  [ #无则为0  {  “rtuno”: ,  “desc”: ““,  “useflag”:  },  {…},  {…}  ]  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下：显示t004\_rtu中rtuno,desc和useflag的值；、  useflag启用标志  rtuno：rtu序号  desc：rtu描述 |

从semodel中t004\_rtu获取

## 7.rtu服务设置使用状态

|  |  |  |
| --- | --- | --- |
| interface: | /set/rtustate |  |
| request json: | {  “tokenid”: 1000,  “rtuno”:,  “useflag”:  } | 设置rtu的fuselag状态；1使用状态，0停用状态； |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

从semodel中t004\_rtu设置useflag的状态

## 8.获取虚拟点rtu服务信息

|  |  |  |
| --- | --- | --- |
| interface: | /get/vptrtuserver |  |
| request json: | {  “tokenid”: 1000,  “rtuno”:no,  } | 获取虚拟点rtu服务信息  rtuno:rtu序号 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  {  “rtuno”: ,  “desc”: ““,  “useflag”:  }  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下：虚拟点rtu的信息  useflag启用标志  rtuno：rtu序号  desc：rtu描述 |

从semodel中t004\_rtu获取 虚拟点rtu信息；（虚拟点的rtutype=1）

## 9.新增虚拟点rtu服务信息

|  |  |  |
| --- | --- | --- |
| interface: | /set/vptrtuserver |  |
| request json: | {  “tokenid”: 1000,  “desc”:”“,  } | 新增虚拟点rtu服务信息 ，  只需要填写rtu序号描述，rtuno后台自动生成 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

从semodel中t004\_rtu新增 虚拟点rtu信息；

## 10.修改虚拟点rtu服务信息

|  |  |  |
| --- | --- | --- |
| interface: | /update/vptrtuserver |  |
| request json: | {  “tokenid”: 1000,  “oldrtuno”:no,  “newdesc”:”“,  “newuseflag”:  } | 修改虚拟点rtu服务信息 ，  rtuno:修改前的rtuno  newdesc:修改后的虚拟点rtu描述  newuseflag：修改后的useflag标志 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下：虚拟点rtu的信息 |

从semodel中t004\_rtu修改 虚拟点rtu信息；

## 11.获取rtu采集转发服务信息

|  |  |  |
| --- | --- | --- |
| interface: | /get/rtuserver |  |
| request json: | {  “tokenid”:”1000 ,  “rtuno”:no,  “rtutype”:  } | 获取rtu服务信息, rtutype:0表示采集服务，2表示转发服务；  rtuno:rtu序号 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  {  “rtuno”: ,  “desc”: ““,  “protid”: ,  “useflag”: ,  “timesync”: ,  “maxcommerr”: ,  “alldatainterval”: ,  “countinterval”: ,  “waittime”: ,  “retrynum”: ,  “chan1”: chanflag 1,  “chan2”: chanflag 2,  “chan1info”:  {  “commtype”:,  “address”:”“,  “desc”:”“,  “threadno”:,  “baud”:,  “databits”:,  “paritybits”:,  “stopbits”:  },  “chan2info”:  {  “commtype”:,  “address”:”“,  “desc”:”“,  “threadno”:,  “baud”:,  “databits”:,  “paritybits”:,  “stopbits”:  },  “protcfg”:  [  {  “paraid”:,  “val”:  },  {…},  {…}  ]  }  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误  protid:规约id  timesync：对时时间  maxcommerr:最大连续通讯失败次数；  alldatainterval:总召时间  countintervall:电度总召时间  waitime：超时时间；  retrynum：重连次数；  chan1:通道1使用状态；  chan2：通道2使用状态；  （chan大于0表示启用，其他表示不启用 chaninfo）  chaninfo通道信息：  commtype：通道设备类型对比2号接口；  address:从站地址和端口号；  desc:从站地址描述  threadno：通道线程组号  baud：波特率  databits：数据位  paritybits：校验位  stopbits：停止位  protcfg：协议配置：  paraid:协议配置no  val:协议配置value |

获取三张表的信息t004\_rtu,t003\_chan,t033\_protcfg;

（chaninfo部分对应表t003\_chan信息，protcfg对应t033\_protcfg信息）

## 12.新增rtu采集转发服务信息

|  |  |  |
| --- | --- | --- |
| interface: | /set/rtuserver |  |
| request json: | {  “tokenid”: 1000,  “rtutype”:,  “desc”: ““,  “protid”: ,  “timesync”: ,  “maxcommerr”: ,  “alldatainterval”: ,  “countinterval”: ,  “waittime”: ,  “retrynum”: ,  “chan1”: chanflag 1,  “chan2”: chanflag 2,  “chan1info”:  {  “commtype”:,  “address”:”“,  “desc”:”“,  “threadno”:,  “baud”:,  “databits”:,  “paritybits”:,  “stopbits”:  },  “chan2info”:  {  “commtype”:,  “address”:”“,  “desc”:”“,  “threadno”:,  “baud”:,  “databits”:,  “paritybits”:,  “stopbits”:  },  “protcfg”:  [  {  “paraid”:,  “val”:  },  {…},  {…}  ]  } | 获取rtu服务信息, rtutype:0表示采集服务，2表示转发服务；  desc:rtu描述  protid:规约id  timesync：对时时间  maxcommerr:最大连续通讯失败次数；  alldatainterval:总召时间  countintervall:电度总召时间  waitime：超时时间；  retrynum：重连次数；  chan1:通道1使用状态；  chan2：通道2使用状态；  （通道状态1是使用，0是不使用，为1时表示对应的chaninfo的信息启用）  chaninfo通道信息：  commtype：通道设备类型对比接口2；  address:从站地址和端口号；  desc:从站地址描述  threadno：通道线程组号  baud：波特率  databits：数据位  paritybits：校验位  stopbits：停止位  protcfg：协议配置：  paraid:协议配置no  val:协议配置value |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误 |

新增三张表的信息t004\_rtu,t003\_chan,t033\_protcfg;

（chaninfo部分对应表t003\_chan信息，protcfg对应t033\_protcfg信息）

## 13.修改rtu采集转发服务信息

|  |  |  |
| --- | --- | --- |
| interface: | /update/rtuserver |  |
| request json: | {  “tokenid”: 1000,  “rtuno”: ,  “rtutype”:,  “useflag”: ,  “desc”: ““,  “protid”: ,  “timesync”: ,  “maxcommerr”: ,  “alldatainterval”: ,  “countinterval”: ,  “waittime”: ,  “retrynum”: ,  “chan1”: chanflag 1,  “chan2”: chanflag 2,  “chan1info”:  {  “commtype”:,  “address”:”“,  “desc”:”“,  “threadno”:,  “baud”:,  “databits”:,  “paritybits”:,  “stopbits”:  },  “chan2info”:  {  “commtype”:,  “address”:”“,  “desc”:”“,  “threadno”:,  “baud”:,  “databits”:,  “paritybits”:,  “stopbits”:  },  “protcfg”:  [  {  “paraid”:,  “val”:  },  {…},  {…}  ]  } | 获取rtu服务信息, rtutype:0表示采集服务，2表示转发服务；  获取rtu服务信息, rtutype:0表示采集服务，2表示转发服务；  rtuno:rtu序号  desc:rtu描述  protid:规约id  timesync：对时时间  maxcommerr:最大连续通讯失败次数；  alldatainterval:总召时间  countintervall:电度总召时间  waitime：超时时间；  retrynum：重连次数；  chan1:通道1使用状态；  chan2：通道2使用状态；  （通道状态1是使用，0是不使用，为1时表示对应的chaninfo的信息启用）  chaninfo通道信息：  commtype：通道设备类型对比接口2；  address:从站地址和端口号；  desc:从站地址描述  threadno：通道线程组号  baud：波特率  databits：数据位  paritybits：校验位  stopbits：停止位  protcfg：协议配置：  paraid:协议配置no  val:协议配置value |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误 |

修改三张表的信息t004\_rtu,t003\_chan,t033\_protcfg;

（chaninfo部分对应表t003\_chan信息，protcfg对应t033\_protcfg信息）

## 14.删除rtu服务信息

|  |  |  |
| --- | --- | --- |
| interface: | /del/rtuserver |  |
| request json: | {  “tokenid”: 1000,  “rtuno”:no,  } | 删除rtu服务信息；  rtuno：rtu序号 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

从semodel中t004\_rtu删除服务信息（附带把chan表，protcfg表删除）；

# bs端界面测点信息接口

## 16.新建ai,di,dd,ao,do测点信息接口

|  |  |  |
| --- | --- | --- |
| interface: | /set/pointinfo |  |
| request json: | {  “tokenid”: 1000,  “pointrtu”:,  “pointno”:,  “pointdesc”:”“,  “pointtype”:,  “mappointno”:  } | 采集rtu新增ai,di,dd测点，type:1 ai,2 di,3 dd;  type:4 ao,5 do;  pointno:点号  desc：点名  ao,do类型点使用mapppintno参数，其他通用；  mappointno对应点号（对应采集的点号一般与pointno相同） |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

从semodel中t038\_ai,t037\_di,t039\_acc新增 点信息；

## 17.获取ai,di,dd,ao,do测点信息接口

|  |  |  |
| --- | --- | --- |
| interface: | /get/pointinfo |  |
| request json: | {  “tokenid”: 1000,  “pointrtu”:,  “pointtype”:,  “listnum”:,  “pagenum”  } | 采集rtu获取ai,di,dd测点，type:1 ai,2 di,3 dd;  type:4 ao,5 do;  listnum 每页的数量，pagenum第多少页； |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  {  “pointnum”:,  data:  [ #无则为0  {  “pointno”:,  “pointdesc”:”“,  “pointtype”:,  “pointvalue”:,  “mappointno”:  },  {…},  {…}  ]  }  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  pointnum表示此类型点总数  data下：  no：点号  desc:点名  type：点类型  value：点值  ao,do类型点使用mapppintno参数，其他通用；ao,do不使用value参数 |

获取 点信息；

## 18.修改ai,di,dd,ao,do测点信息接口

|  |  |  |
| --- | --- | --- |
| interface: | /update/pointinfo |  |
| request json: | {  “tokenid”: 1000,  “oldpointno”:,  “oldpointrtu”:,  “oldpointtype”:,  “pointno”:,  “pointdesc”:”“,  “mappointno”:  } | 采集rtu修改ai,di,dd测点，type:1 ai,2 di,3 dd;  type:4 ao,5 do;  oldpointno修改前的点号  oldpointrtu修改前的rtu号  oldpointtype修改前的类型  pointno修改后的点号  pointdesc修改后的描述  ao,do类型点使用mapppintno参数，其他通用； |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下 |

从semodel中t038\_ai,t037\_di,t039\_acc新增 点信息；

## 22.新建虚拟点测点接口

|  |  |  |
| --- | --- | --- |
| interface: | /set/vptpointinfo |  |
| request json: | {  “tokenid”: 1000,  “pointrtu”:,  “vtno”:,  “vtclass”:,  “vtdesc”:”“,  “codedesc”:”“,  “eigenno”:,  “sysprop”:,  “initval”:,  “calcindex”:,  “eigenname”:”“  } | 虚拟点rtu新增虚拟点测点，vtclass表示虚拟点是什么类型的虚拟点：0 di,1 ai,2 acc,3 do,4 ao;  no：虚拟点号  class:点类型  desc:点描述  codedesc:虚拟点功能code  eigenno:特征号，  sysprop:sys值，initval初始值；  calcindex索引  eigenname特征描述 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

从semodel中t036\_vipnt新增 点信息；(根据vtclass,在ai,di,dd,ao,do表插入数据)

## 23.获取虚拟点测点接口

|  |  |  |
| --- | --- | --- |
| interface: | /get/vptpointinfo |  |
| request json: | {  “tokenid”: 1000,  “pointrtu”:  “listnum”:,  “pagenum”  } | 虚拟点rtu获取测点数据 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  {  “pointnum”:,  data:  [ #无则为0  {  “vtrtuno”:,  “vtno”:,  “vtclass”:,  “vtdesc”:”“,  “vtvalue”:,  “codedesc”:”“,  “eigenno”:,  “sysprop”:”“,  “initval”:,  “calcindex”:,  “eigenname”:”“  },  {…},  {…}  ]  }  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  pointnum表示此类型点总数  no：虚拟点号  class:点类型  desc:点描述  codedesc:虚拟点功能code  eigenno:特征号，  sysprop:sys值，initval初始值；  calcindex索引  eigenname特征描述 |

从semodel中t036\_vipnt获取 点信息；(根据vtclass,在ai,di,dd,ao,do表插入数据)

## 24.修改虚拟点测点接口

|  |  |  |
| --- | --- | --- |
| interface: | /update/vptpointinfo |  |
| request json: | {  “tokenid”: 1000,  “oldpointrtu”:,  “oldvtno”:,  “oldvtclass”:,  “vtno”:,  “vtclass”:,  “vtdesc”:”“,  “codedesc”:”“,  “eigenno”:,  “sysprop”:”“,  “initval”:,  “calcindex”:,  “eigenname”:”“  } | old表示之前的值：  no：虚拟点号  class:点类型  desc:点描述  codedesc:虚拟点功能code  eigenno:特征号，  sysprop:sys值，initval初始值； |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

从semodel中t036\_vipnt修改 点信息；(根据vtclass,修改在ai,di,dd,ao,do表插入数据)

## 25.新建zfai,zfdi,zfdd,zfao,zfdo测点接口

|  |  |  |
| --- | --- | --- |
| interface: | /set/zfpointinfo |  |
| request json: | {  “tokenid”: 1000,  “zfrtuno”:,  “zfpointno”:,  “zfpointdesc”:”“,  “zfpointtype”:,  “srcrtuno”:,  “srcpointno”:,  “factora”:,  “factorb”:,  “invert”:  } | 采集rtu新增zfai,zfdi,zfdd测点，type:11 zfai,12 zfdi,13 zfdd; 14 zfao,15 zfdo;  srcrtuno:对应采集rtu  srcpointno:对应采集点号  factora:转换斜率1  factorb:转换斜率2  invert：是否取反  zfai,zfacc:额外使用参数factora，factorb；  zfdi：额外使用参数invert；  其他通用 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下 |

从semodel中t038\_ai,t037\_di,t039\_acc新增 点信息；

## 26.批量新建zfai,zfdi,zfdd,zfao,zfdo测点接口

|  |  |  |
| --- | --- | --- |
| interface: | /set/zfpointinfo |  |
| request json: | {  “tokenid”: 1000,  “zfpointtype”:,  “zfrtuno”:,  data:[  “srcrtuno”:,  “srcpointno”:  ~~“factora”:,~~  ~~“factorb”:,~~  ~~“invert”:~~  ],  [...], [......]} | 采集rtu新增zfai,zfdi,zfdd测点，type:11 zfai,12 zfdi,13 zfdd; 14 zfao,15 zfdo;  srcrtuno:对应采集rtu  srcpointno:对应采集点号 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下 |

## 27.获取zfai,zfdi,zfdd,zfao,zfdo测点接口

|  |  |  |
| --- | --- | --- |
| interface: | /get/zfpointinfo |  |
| request json: | {  “tokenid”: 1000,  “zfrtuno”:,  “zfpointtype”:,  “listnum”:,  “pagenum”  } | 采集rtu新增zfai,zfdi,zfdd测点，type:11 zfai,12 zfdi,13 zfdd; 14 zfao,15 zfdo; |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  {  “pointnum”:,  data:  [ #无则为0  {  “zfpointno”:,  “zfpointtype”:,  “zfpointdesc”:”“,  “zfpointvalue”:,  “srcrtuno”:,  “srcpointno”:,  “factora”:,  “factorb”:,  “invert”:  },  {…},  {…}  ]  }  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  pointnum表示此类型点总数  data下  srcrtuno:对应采集rtu  srcpointno:对应采集点号  factora:转换斜率1  factorb:转换斜率2  invert：是否取反  zfai,zfacc:额外使用参数factora，factorb；  zfdi：额外使用参数invert；  其他通用  zfao,zfdo：不使用zfpointvalue参数 |

从semodel中t038\_ai,t037\_di,t039\_acc新增 点信息；

## 28.修改zfai,zfdi,zfdd,zfao,zfdo测点接口

|  |  |  |
| --- | --- | --- |
| interface: | /update/zfpointinfo |  |
| request json: | {  “tokenid”: 1000,  “oldzfrtuno”:,  “oldzfpointno”:,  “oldzfpointtype”:,  “zfpointno”:,  “zfpointdesc”:”“,  “zfpointtype”:,  “srcrtuno”:,  “srcpointno”:,  “factora”:,  “factorb”:,  “invert”:  } | 采集rtu新增zfai,zfdi,zfdd测点，type:11 zfai,12 zfdi,13 zfdd; 14 zfao,15 zfdo;  srcrtuno:对应采集rtu  srcpointno:对应采集点号  factora:转换斜率1  factorb:转换斜率2  invert：是否取反  zfai,zfacc:额外使用参数factora，factorb；  zfdi：额外使用参数invert；  其他通用 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下 |

从semodel中t038\_ai,t037\_di,t039\_acc新增 点信息；

## 31.删除测点接口（测点通用）

|  |  |  |
| --- | --- | --- |
| interface: | /del/pointinfo |  |
| request json: | {  “tokenid”: 1000,  “pointrtu”,  “pointtype”:,  “pointno”:  [  ]  } | 删除测点ai,di,dd,ao,do,vit测点，type:1 ai,2 di,3 dd,4 ao,5 do,6 vit,11 zfai,12 zfdi,13 zfdd,14 zfao,15 zfdo;  pointno按点号删，可以一次删多个点号 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下 |

从semodel中删除 点信息；

## 32.清空测点接口（测点通用）

|  |  |  |
| --- | --- | --- |
| interface: | /del/pointinfotype |  |
| request json: | {  “tokenid”: 1000,  “pointrtu”:,  “pointtype”:  } | 删除测点ai,di,dd,ao,do,vit测点，type:1 ai,2 di,3 dd,4 ao,5 do,6 vit,11 zfai,12 zfdi,13 zfdd,14 zfao,15 zfdo; |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  data下 |

从semodel中清空整个rtu 点信息；

# 新增接口：

# bs端界面系统服务页面接口

## 33.通知fep-ha重启接口（后端服务重启）

|  |  |  |
| --- | --- | --- |
| interface: | /set/sysrestart |  |
| request json: | {  “tokenid”: 1000,  } | 通知后台重启fep-ha（linux版会重启机器） |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

通知后端重启fep-ha进程或重启linux机器

# bs端新增接口

## 34.rtu通道连通状态接口

|  |  |  |
| --- | --- | --- |
| interface: | /get/rtuchanstate |  |
| request json: | {  “tokenid”: 1000,  “rtuno”:  } | 获取当前rtu状态和对应的通道状态 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  {  “rtudesc”:””,  “protdesc”:””,  “channum”:,  “chan1desc”:””,  “chan1state”:,  “chan2desc”:””,  “chan2state”:  }  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  rtudesc:rtu服务描述  portdesc:规约描述  channum：启用的chan数量  chan1desc:通道1描述  chan2desc:通道2描述  chan1state:通道1状态  chan2state:通道2状态 |

从实时库获取rtu状态和通道状态

# bs端按钮接口

## 35.采集服务批量导入接口：

|  |  |  |
| --- | --- | --- |
| interface: | /set/batchpointinf |  |
| request json: | {  “tokenid”: 1000,  “rtuno”:,  “pointtype”:,  “startpoint”:,  “endpoint”:  } | 批量创建测点；  Rtuno：服务rtuno  Pointtype:五个类型 ai,di,ao,do,dd;  type:1 ai,2 di,3 dd,4 ao,5 do;6 vpt,11 zfai,12 zfdi,13 zfdd; 14 zfao,15 zfdo;  startpoint:起始点号  endpoint:终止点号 |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:0  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误； |

批量设置点数据；

## 36.获取采集rtu点类型的所有点信息（去除zfrtu已经配用的采集rtu点）接口

(例如ai,采集rtu1有1-100号点，转发rtu10已经配置了采集rtu1的1-10号点，那么此接口就返回11-100号rtu1的信息点)

|  |  |  |
| --- | --- | --- |
| interface: | /get/filterpointinfo |  |
| request json: | {  “tokenid”: 1000,  “pointrtu”:,  “pointzfrtu”:,  “pointtype”:  } | 根据采集rtuno,type和zfrtuno获取采集rtu未选用的相关点号和点描述；  type:1 ai,2 di,3 dd,4 ao,5 do; |
| response json: | {  “result”: ,  “errorinfo”:”“,  “data”:  {  “pointnum”:,  data:  [ #无则为0  {  “pointno”:,  “pointdesc”:”“  },  {…},  {…}  ]  }  } | “result”:200表示成功，404失败，0表示请求数据格式错误，405表示未知错误；  pointnum表示此类型点总数  data下：  no：点号  desc:点名 |