定义前端界面刷新，或SVG图切换，jar包请求ISDataServer的逻辑梳理，接口定义：

# 1.刷新页面：

前端发全量点，ISDataServer需要回复全量点。

## 1.1前端的界面上有点：

前端界面下发：

|  |  |
| --- | --- |
| Json格式示例 |  |
| [  {  "id": "G14\_S04\_Z45",  "type": "yx"  },  {  "id": "G14\_S04\_Z46",  "type": "yx"  }  ] | id: 点代码  type: 点类型，分为yx（遥信），yc（遥测）； |

Jar包同上，直接转；

ISDataServer回复：

|  |  |
| --- | --- |
| Json格式示例 |  |
| {  "stateCode": 1,  "errMsg": "",  "pointData": [{  "id": "G14\_S04\_Z45",  "idonly": "",  "inccount": "0",  "status": "384",  "switchcount": "0",  "type": "yx",  "val": "0"  }, {  "id": "G14\_S04\_Z46",  "idonly": "",  "inccount": "0",  "status": "384",  "switchcount": "0",  "type": "yx",  "val": "0"  }]  } | stateCode：为状态码，为1则正常；为2则为Json格式错误，为3则为clientid为首包，但点信息发的为””;  errMsg：为前端界面提示错误信息，状态码为1，errMsg为””；  pointData：为点的数据，首包回复是全量信息，格式如左边描述。 |

## 1.2前端界面上没点：

前端界面下发：

|  |  |
| --- | --- |
| Json格式示例 |  |
| [] | 因为界面没点，所以发的是空json包 |

Jar包收到后，则直接处理，不再下发给ISDataServer；

## 1.3 前端界面上配置的点在实时库无记录

前端界面下发：

|  |  |
| --- | --- |
| Json格式示例 |  |
| [  {  "id": "G14\_S04\_Z45",  "type": "yx"  },  {  "id": "G14\_S04\_Z46",  "type": "yx"  }  ] | id: 点代码  type: 点类型，分为yx（遥信），yc（遥测）； |

Jar包同上，直接转；

ISDataServer回复：

|  |  |
| --- | --- |
| Json格式示例 |  |
| {  "stateCode": 1,  "errMsg": "",  "pointData": [{  "id": "G14\_S04\_Z45",  "type": "yx",  "warning": "Rdb not exist the id!",  }, {  "id": "G14\_S04\_Z46",  "idonly": "",  "inccount": "0",  "status": "384",  "switchcount": "0",  "type": "yx",  "val": "0"  }]  } | stateCode：为状态码，为1则正常；为2则为Json格式错误，为3则为clientid为首包，但点信息发的为””;  errMsg：为前端界面提示错误信息，状态码为1，errMsg为””；  pointData：为点的数据，首包回复是全量信息，如果某个点或者所有的点都是不存在，则该不存在的点提供：id，type，waring来回复前端，格式如左边描述。 |

# 2.当界面不刷新：

前端发””，ISDataServer需要回复变化的点数据。

前端界面下发：””(注意区别于NULL)

Jar包直接转：””

ISDataServer回复：

如果变化点数据不为空，则如下：

|  |  |
| --- | --- |
| Json格式示例 |  |
| {  "stateCode": 1,  "errMsg": "",  "pointData": [{  "id": "G14\_S04\_Z45",  "status": "384",  "switchcount": "0",  "val": "0"  }, {  "id": "G14\_S04\_Z46",  "inccount": "0",  "status": "384",  "switchcount": "1",  "val": "0"  }]  } | stateCode：为状态码，为1则正常；为2则为Json格式错误，为3则为clientid为首包，但点信息发的为””;  errMsg：此时为正常状态，则错误描述为””；  pointData：描述为变化点的变化字段信息数据，如左边描述。 |

如果变化点数据为空，则如下：

|  |  |
| --- | --- |
| Json格式示例 |  |
| {  "stateCode": 1,  "errMsg": "",  "pointData": []  } | stateCode：为状态码，为1则正常；为2则为Json格式错误，为3则为clientid为首包，但点信息发的为””;  errMsg：此时为正常状态，则错误描述为””；  pointData：描述为变化点的变化字段信息数据，如左边描述。 |

# 3.当切换界面时：

包括SVG图切换Tab页和新增网页，具体罗列如下：

## 3.1SVG图切换：

clientID不变，点信息变化：

前端界面下发：

|  |  |
| --- | --- |
| Json格式示例 |  |
| [  {  "id": "G14\_S04\_Z46",  "type": "yx"  }  ] | id: 点代码  type: 点类型，分为yx（遥信），yc（遥测），acc（电度）； |

jar包转发，同上；

ISDataServer回复：

|  |  |
| --- | --- |
| Json格式示例 |  |
| {  "stateCode": 1,  "errMsg": "",  "pointData": [{  "id": "G14\_S04\_Z46",  "idonly": "",  "inccount": "0",  "status": "384",  "switchcount": "0",  "type": "yx",  "val": "0"  }]  } | stateCode：为状态码，为1则正常；为2则为Json格式错误，为3则为clientid为首包，但点信息发的为””;  errMsg：为前端界面提示错误信息；状态码为1，errMsg为””；  pointData：为点的数据，首包回复是全量信息，格式如左边描述。 |

## 3.2新增网页：

新增clientID，点信息：

具体过程参考”1.所做的动作-刷新页面”

# 4.错误情况：

目前分为：

\*Json解析格式错误；

\*ISDataServer收到新的clientID，但点信息为空；

实例如下：

## 4.1Json解析格式错误：

前端界面下发错误的Json格式包；

Jar包同上，直接转；

ISDataServer回复：

|  |  |
| --- | --- |
| Json格式示例 |  |
| {  "stateCode": 2,  "errMsg": "Json format error!",  "pointData": []  } | stateCode：为状态码，为1则正常；为2则为Json格式错误，为3则为clientid为首包，但点信息发的为””;  errMsg：为前端界面提示错误信息，状态码为2，则errMsg提示”Json format error!”；  pointData：为点的数据，此处为空。 |

## 4.2ISDataServer收到新的clientID，但点信息为空

前端界面下发Json格式请求包；

|  |  |
| --- | --- |
| Json格式示例 |  |
| [  {  "id": "G14\_S04\_Z46",  "type": "yx"  }  ] | id: 点代码  type: 点类型，分为yx（遥信），yc（遥测），acc（电度）； |

Jar包错误地将变化请求包发送另一个ISDataServer；

ISDataServer回复：

|  |  |
| --- | --- |
| Json格式示例 |  |
| {  "stateCode": 3,  "errMsg": "New ClientId send empty pointInfo!",  "pointData": []  } | stateCode：为状态码，为1则正常；为2则为Json格式错误，为3则为clientid为首包，但点信息发的为””;  errMsg：为前端界面提示错误信息，状态码为3，则errMsg提示”New ClientId send empty pointInfo!”；  pointData：为点的数据，首包回复是全量信息，格式如左边描述。 |