# 返回值与通信格式说明

* **用户通过各类Interface发来的指令，系统处理完后会整合成JSON格式的字符串返回给前端，前端可按照JSON的格式进行解码。比如设计Get指令将系统的一些状态值返回到前端**

std::vector<std::pair<std::string, std::any>> out\_param;

**out\_param.push\_back(std::make\_pair<std::string, std::any>("part\_pq", out\_data.part\_pq));**

**out\_param.push\_back(std::make\_pair<std::string, std::any>("end\_pq", out\_data.end\_pq));**

**out\_param.push\_back(std::make\_pair<std::string, std::any>("end\_pe", out\_data.end\_pe));**

**out\_param.push\_back(std::make\_pair<std::string, std::any>("motion\_pos", out\_data.motion\_pos));**

**按照如下格式依次将需要返回给前端的状态量加入到out\_param，最后将这个变量赋值给ret()**

**ret() = out\_param;**

* **对于增加的nrtController，控制系统与其下的被控设备需交换数据。控制系统下发的字符串是通过上述方法转变而来的，而控制系统接收回来的字符串需写成如下形式：**

**"{\"di\_vec\":[6,6,5,5,5,5],\"ai\_vec\":[3,9,5,5,5,5],\"cmd\_vec\":[\"wangwu\",\"wwu\"]}";**

**这里依次把nrtController下的IO量以及cmd\_vec信息组合，并发送给控制系统。**