Ice for JS Client

首先利用slice2js生成ice文件对应的js文件，在html或相应的前端页面中添加该js文件和Ice.js、Glacier2.js、IceGrid.js文件。没有Ice的js库可以到该连接下载http://cdnjs.com/libraries/ice。
注意：服务端配置要添加对ws协议的支持，因为js是通过websocket实现的。
registry.cfg的配置如下，多个协议要用英文字符“:”分割

```
IceGrid.Registry.Client.Endpoints=tcp -p 4061:ws -p 4062
```

grid.xml的配置如下，endpoints="tcp:ws"，多个协议要用英文字符“:”分割

```
<service name="TicketService" entry="com.zzwtec.iceTicketProject.ice.service.MyTicketService">
   <adapter name="TicketService" id="TicketService${id}" endpoints="tcp:ws" replica-
group="TicketServiceRep"></adapter>
</service>
```

前端页面代码如下

```
var communicator;
var iceData = new Ice.InitializationData();

Ice.Promise.try(
        function() {
           iceData.properties = Ice.createProperties();
           iceData.properties.setProperty("Ice.Default.Locator","IceGrid/Locator:ws -h 192.168.0.112 -p
4062");
           communicator = Ice.initialize(iceData);
           var proxy = communicator.stringToProxy("TicketService");
           return ticket.TicketServicePrx.checkedCast(proxy).then(
                function (ticketServicePrx) {
                        //调用
                        return ticketServicePrx.queryMyOrders("13631276694");
                }
           ).then(
                function (result){
                        //调用返回结果处理
                }
           );
        }
).finally(
        //释放communicator
        function() {
           if(communicator) {
              return communicator.destroy();
           }
        }
).exception(
        //异常处理
        function(ex) {
           console.log(ex.toString());
           //process.exit(1);
        }
);
```