aardio 范例: UPnP 端口映射查看工具

```
//端口映射管理
import win.ui;
/*DSG{{*/
var winform = win.form(text="UPnP 端口映射查看工具";right=966;bottom=622)
winform.add()
/*}}*/

import web.view;
var wb = web.view(winform);

import sys.upnp.nat;
var natUpnp = sys.upnp.nat();

//测试添加端口映射，除第一个参数以外的其他参数都可以省略
natUpnp.add(9973,"TCP",9973,,"添加端口映射测试")

wb.external = {
    getMappingCollection = function(){
        return web.json.stringifyArray(natUpnp.getTable());
    };
    deleteItem = function(externalPort,protocol){
        return natUpnp.remove(externalPort,protocol)
    };
    natUpnpValid = function(){
        return natUpnp.valid();
    };
}

wb.html = /**
<!DOCTYPE html><html>
<head>
    <meta charset="utf-8" />
    <title>WebView2</title>
    <script src="https://lib.baomitu.com/react/17.0.2/umd/react.production.min.js"></script>
    <script src="https://lib.baomitu.com/react-dom/17.0.2/umd/react-dom.production.min.js"></script>
    <script src="https://lib.baomitu.com/antd/4.17.0-alpha.3/antd.min.js"></script>
    <link rel="stylesheet" href="https://lib.baomitu.com/antd/4.17.0-alpha.3/antd.min.css">
    <script src="https://unpkg.com/@babel/standalone@7/babel.min.js"></script>
    <style type="text/css">
    </style>
</head>
<body>

<script type="text/babel">
    const { useState,useEffect,useCallback,useRef } =  React;
    const { Button,Table,DatePicker,Alert,Tooltip,Space,Tag,Popconfirm } = antd;

    const App = () => {
            const [data,setData] = useState([]);
            const [valid,setValid] = useState(true);

            useEffect(
                async ()=> {
                    var ds = JSON.parse(await aardio.getMappingCollection());
                    setData(ds);

                    if(!ds.length){
                        setValid(await aardio.natUpnpValid());
                    }
                },[]
            )

            return (
                <div style={{ width: '100%', margin: '0 auto', padding:'10px' }}>
                { !valid && <Alert message="当前网络不支持 UPnP 自动端口映射" type="error" showIcon  closable  /> }
                <Table size="middle"  dataSource={data} columns={[
                        {
                            title: '协议',
                            dataIndex: 'protocol',
                            width: 300,
                        },
                        {
                            title: '外网端口',
                            dataIndex: 'externalPort',
                        },
                        {
                            title: '内网端口',
                            dataIndex: 'internalPort',
                        },
                        {
                            title: '内网主机',
                            dataIndex: 'internalClient',
                        },
                        {
                            title: '描述',
                            dataIndex: 'description',
                        },
                        {
                            title: '启用',
                            dataIndex: 'enabled',
                            render: enabled => (
                                <>
                                    <Tag color={ enabled  ? 'green' : '#CCC' } >
                                        {enabled ? "启用" : "禁用"}
                                    </Tag>
                                </>)
                        },
                        {
                            title: '操作',
                            key: 'action',
                            render: (text, record) => (
                            <Space size="middle">
```

```
                                <Popconfirm title={"确定删除外网映射端口："+record.externalPort+" 协议："+record.protocol+" 吗"} onConfirm={ async ()=>{
                                    await aardio.deleteItem(record.externalPort,record.protocol)
                                    setData(JSON.parse(await aardio.getMappingCollection()));
                                } } okText="确认" cancelText="取消">
                                    <a>删除</a>
                                </Popconfirm>
                            </Space>
                            ),
                        },
                ]} />

                </div>
            );
    };

    ReactDOM.render(<App />, document.querySelector('#app'));
</script>

<div id="app"></div>
</body>
**/

winform.show();
win.loopMessage();
```

Markdown 格式