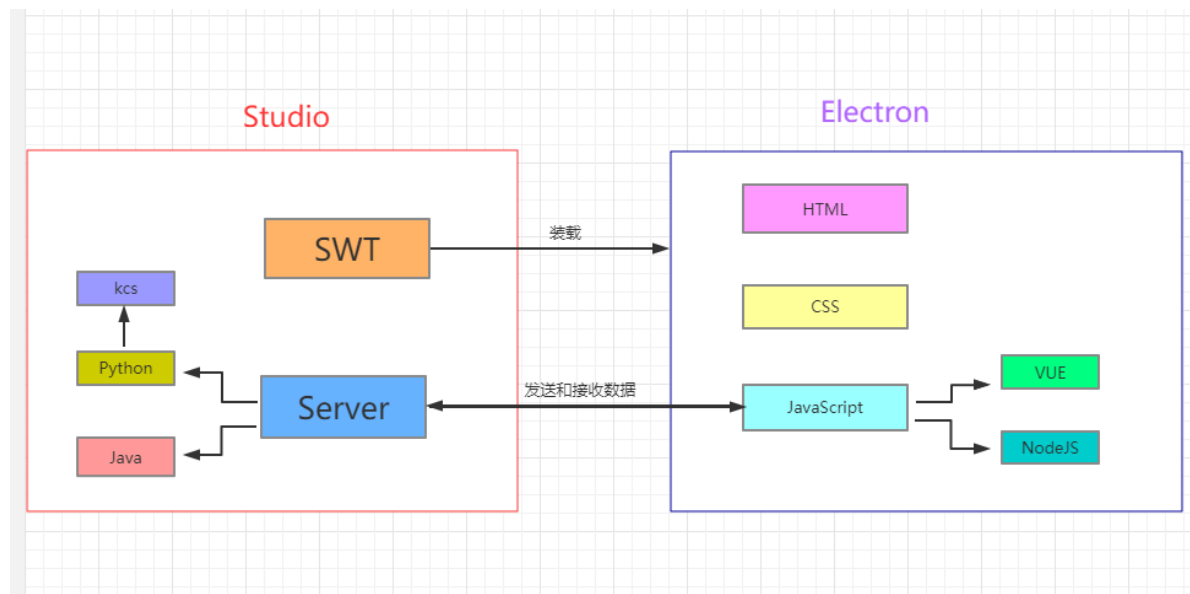


RT-Thread Settings Web化方案二 集成 electron

框架



生命周期

- 启动 RT-Thread Settings
 - 开启本地服务 Server，Server 的端口号为 Java 随机生成的可用端口号 port_number
- 前后端通过端口号绑定
 - 执行 electronAPP.exe port_number，其中 electronAPP.exe 为 启动的可执行程序，port_number 传递给前端绑定对应的 Server
 - 前端获取端口号

```
const port = process.argv[1]
```

- 前端载入本地服务

```
let URL = 'http://localhost:'+port;  
mainWindow.loadURL(URL);
```

- Election 与 Studio 融合
 - 将 Election 平面化地嵌入到 Studio 的 编辑器 (org.eclipse.ui.part.MultiPageEditorPart) 中

```
OS.SetParent(hwnd, composite.handle); // hwnd为 Election 的窗口句柄，  
composite为父容器
```

- Election 与 Studio 数据传递
 - 前后端数据发送与接收示例
 - 前端

```
<body>
```

```

<input class="name" type="text" placeholder="name" /> </br>

<button id="send" class="send"> Send </button>

</body>

$("#send").click(function(){
    var name = $(".name").val();
    console.log(name);
    $.ajax({
        url:'http://localhost:80/test',
        type:'GET', // 请求类型
        data:{ // 发送json数据给后端
            'name':name
        },
        success:function (arg) { // 接收后端发送的数据
            console.log(arg)
        },
    })
});

```

■ 后端

```

JSONObject modifidMap = new JSONObject();
public class ServerHandler extends HttpServlet {
    // 接收并处理前端的 Get 请求
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        // TODO Auto-generated method stub
        resp.setHeader("Access-Control-Allow-Origin", "*");
        resp.setHeader("Access-Control-Allow-Headers", "X-Requested-
With");
        resp.setHeader("Access-Control-Allow-Methods",
"PUT,POST,GET,DELETE,OPTIONS");
        resp.setHeader("X-Powered-By", "3.2.1");
        resp.setHeader("Content-Type",
"application/json;charset=utf-8");
        JSONObject ret = new JSONObject();
        String name = req.getParameter("name");
        System.out.println(name);
        try {
            ret.put("ret", "0");
            ret.put("name", name);
        } catch (Exception ex){
            ret.put("ret", "-1");
            ret.put("error", ex.getMessage());
        }
        if(req.getParameter("callback")!=null) {
            resp.getWriter().write(req.getParameter("callback")+
("+ret.toString()+")");
        } else {
            resp.getWriter().write(ret.toString());
        }
        modifyMapByFrontend();
    }
}

```

```
/**
 * 根据前端发送的数据修改 modifidMap
 */
private void modifyMapByFrontend() {

    // 修改modifidMap

}
}
```

- 关闭 RT-Thread Settings
 - 关闭相应端口的本地服务
 - kcs 执行 modifidMap 组成的命令，modifidMap 为前端修改的所有记录
 - 更新软件包

```
pkgs --update

scons --target = eclipse
```

更新

- Electron 有自己的一套更新机制，需要提供一个站点用于存放最新版本的 Electron App
- 打开 RT-Thread Settings 时，检测 RT-Thread Settings 是否有版本更新，提示用户是否更新。选择更新后，关闭 studio 其他 RT-Thread Settings，然后执行更新的操作

其他细节

其他前端与后端的细节，例如原型设计、接口等，请见 【RT-Thread Settings Web 化 方案一】