

# Electron 蓝牙链路

## Electron 脚手架地址

- [Electron-Vue-Element](#)
- 按照 [README](#) 安装依赖并启动



Node.js版本不得高于16.x

## 框架说明

- 谷歌cef内核的桌面端开发框架，等同于WebView
- `src/background.js` 为壳的启动目录，桌面端主进程

```
let window = new BrowserWindow();//创建窗体
window.loadURL("https://...");//连接到远程
```

- `src/preload.js` 为页面预加载渲染进程,将 `ipcRenderer` 的方法暴露到js全局对象池中

```
const { contextBridge, ipcRenderer } = require("electron");
const ipc = {
  listen: (handler, callback) => {
    ipcRenderer.on(handler, function (_, ...args) {
      callback(...args);
    });
  },
}
contextBridge.exposeInMainWorld("electron", ipc);
```

在js 中

```
window.electron.listen("log", console.log);
```

- [进程间通信](#)
  - `ipcMain` Electron主进程，消息进程

```
ipcMain.on("handler", (sender, ...args) => {}); // 接收来自ipcRenderer的send
```

- `ipcRenderer` Electron渲染进程

```
ipcRenderer.on("handler", (sender, ...args) => {}); // 接收来自window.webContents.send的消息  
ipcRenderer.send("handler", ...args); // 给ipcMain发送消息
```

- `webContents` Web控制层

```
window.webContents.send("handler", ...args); // 给ipcRenderer发送消息
```

## 当前连接样例

- 页面发起 `navigator.bluetooth.requestDevice`
- 对应的 `window` 接收 `"select-blue-device"`
- 将得到的蓝牙列表通过 `webContents` 发送到 `ipcRenderer` 推送到页面
- 页面选择蓝牙id通过 `ipcRenderer` 发送到 `ipcMain` 进行连接

```
import { BrowserWindow, ipcMain } from "electron";
let window : BrowserWindow;
/**
 * 当接收到来自WebView的
 * @example {navigator.bluetooth.requestDevice()}时触发
 */
window.webContents.on("select-bluetooth-device",
  (event, deviceList, callback) => {
    event.preventDefault();
    if (!isConnectingBluetooth) {
      isConnectingBluetooth = true;
      let connectHandler = () => {};
      let cancelHandler = () => {};
      connectHandler = (e, ...args) => {
        console.log(...args);
        ipcMain.off("selectBluetooth", connectHandler);
        ipcMain.off("cancelBluetooth", cancelHandler);
        isConnectingBluetooth = false;
        callback(...args);
      };
      cancelHandler = (e, _) => {
        application.$Logger.warn("Cancel request");
        ipcMain.off("selectBluetooth", connectHandler);
        ipcMain.off("cancelBluetooth", cancelHandler);
        isConnectingBluetooth = false;
        callback("");
      };
      ipcMain.on("selectBluetooth", connectHandler);
      ipcMain.on("cancelBluetooth", cancelHandler);
    }
    if (deviceList && deviceList.length > 0) {
      window.webContents.send("bluetoothList", deviceList);
    }
  }
);
```

## 需要

- 前端的笔列表信息的推送
- 笔搜索状态，搜索是否结束
- 给出在 `preload.js` 中的回调样例，以及 `background.js` 中的加载样例