Electron --v1.4.0

需要实现:

1) 可拖拽 -- 已完成

2) 托盘图标,右键点击托盘选择功能 -- 已完成

3) 初始化快捷键，实现用户可自定义快捷键 -- 已完成快捷键初始化，还未实现自定义快捷键

4) 界面美化(包括：loading页面，切换在线离线状态，消息闪烁提醒,声音提醒) -- 已完成

5) 应用程序打包(windows,mac,linux系统) --已完成

主窗口属性：{

Width:800(窗口宽度,单位像素. 默认是 800)

Height:600(窗口高度,单位像素. 默认是 600)

resizable:false(布尔值，是否可调整宽高)

center: true,(是否居中)

show: true,(是否显示)

frame: false,(是否可拖拽)

title:String,(窗口默认title 默认 "Electron")

autoHideMenuBar:true,(除非点击 Alt，否则隐藏菜单栏.默认为 false.)

skipTaskbar:true(是否在任务栏中显示窗口. 默认是false)

titleBarStyle:hidden (隐藏标题栏，内容充满整个窗口, 然后它依然在左上角，仍然受标准窗口控制.)

alwaysOnTop: true,(窗口是否总是显示在其他窗口之前. 在 Linux 上无效. 默认为 false)

icon: 'assets/icon.png',(窗口图标)

}

const { app, Menu, nativeImage, Tray,ipcMain,ipcRenderer,BrowserWindow} = require('electron'); -- 所需的electron对象

主要API:

win = new BrowserWindow({props}); -- 创建窗口

win.loadURL() -- 渲染文件的路径

win.openDevTools() -- 打开控制台，方便调试

app.on(‘ready’,function(){}) -- app 加载完成后，执行回调函数

app.quit() -- 关闭所有窗口

tray = new Tray(‘png’) -- 创建托盘程序，并指定托盘图标

tray.setToolTip('XiaoNeng') -- 设置托盘程序名称

const constextMenu = Menu.buildFromTemplate([{ label: '帮助反馈',

enabled: false, -- 不可点击

Function(){

Console.log(‘callback’)

}}]) -- 创建菜单,点击时执行回调函数

tray.setContextMenu(constextMenu); -- 将菜单绑定至托盘

ipcMain.on('global-shortcut',(event,num) => {}); -- 主进程对象，收到渲染进程 消息后，执行回调函数，其中event对象 有方法event.sender.send(‘global-shortcut’,’’) 回应消息。

ipcRenderer -- 渲染进程专有对象，有方法 ipcRenderer.send('global-shortcut',input.value),发 送消息至主进程

打包：1) 下载eletron-package

npm install electron-packager --save-dev

2) 开始打包

Windows:electron-packager ./ XiaoNeng --platform=win32 --arch=x64 --version=1.4.0 --out=./ --asar --overwrite --icon=./window/XiaoNeng.ico

Mac:electron-packager . XiaoNeng --platform=darwin --arch=x64 --version=1.4.0 --out=./ --asar --overwrite --icon=./window/XiaoNeng.icns

Linux：electron-packager . XiaoNeng --platform=linux --arch=x64 --version=1.4.0 --out=./ --asar --overwrite --icon=./window/XiaoNeng.png