1.如何创建窗口和改变窗口：

复制代码

import { BrowserWindow, globalShortcut, Menu } from 'electron'

import Common from '../common/common.js'

const winURL = process.env.NODE\_ENV === 'development'

? `http://localhost:${require('../../../../config').port}`

: `file://${\_\_dirname}/index.html`

function createWindow() {

var mainWindow = new BrowserWindow({

height: Common.WINDOW\_SIZE\_LOGIN.height,

width: Common.WINDOW\_SIZE\_LOGIN.width,

resizable: false,

frame: false,

});

mainWindow.loadURL(winURL);

mainWindow.on('closed', () => {

mainWindow = null

});

//前期为了调试方面，默认打开控制台

mainWindow.webContents.openDevTools({ detach: true });

//注册打开控制台的快捷键

globalShortcut.register('ctrl+shift+alt+e', function () {

let win = BrowserWindow.getFocusedWindow();

if (win) {

win.webContents.openDevTools({ detach: true });

}

});

//去掉默认菜单栏

Menu.setApplicationMenu(null);

// eslint-disable-next-line no-console

console.log('mainWindow opened');

//添加这段代码

BrowserWindow.mainWindow = mainWindow;

return mainWindow;

}

module.exports = {

createWindow

};

复制代码

如何把electron的按钮的最大化，最小化 关闭窗口改变；

复制代码

/\*\*

\* Created by Administrator on 2017/4/26.

\* 页面对窗口的一些操作封装，用于渲染进程

\*/

"use strict";

const Common = require('../common/common.js');

const { ipcRenderer, remote } = require('electron');

const RUN\_LOCATION = '\\Software\\Microsoft\\Windows\\CurrentVersion\\Run';

//const file = process.execPath;

//const WinReg = require('winreg');

let flashTrayTimer = null;

class WindowUtil {

// 窗口最小化

static minWindow() {

remote.getCurrentWindow().minimize();

}

// 窗口最大化

static maxWindow(isMaxed) {

const browserWindow = remote.getCurrentWindow();

if (!isMaxed) {

browserWindow.unmaximize();

} else {

browserWindow.maximize();

}

}

// 设置窗口是否能改变大小，参数true/false

static setResizable(resizable) {

remote.getCurrentWindow().setResizable(resizable);

}

// 下载文件

static download(url) {

remote.getCurrentWebContents().downloadURL(url);

}

// 隐藏窗口

static hide() {

const browserWindow = remote.getCurrentWindow();

browserWindow.hide();

}

// 显示窗口

static show() {

const browserWindow = remote.getCurrentWindow();

browserWindow.show();

}

// 窗口闪烁

static flashFrame() {

const browserWindow = remote.getCurrentWindow();

// if(browserWindow.isFocused() || browserWindow.isVisible())

if (!browserWindow.isFocused()) {

browserWindow.showInactive();

browserWindow.flashFrame(true);

}

}

// 设置窗口最前端显示

static setAlwaysOnTop(top) {

const browserWindow = remote.getCurrentWindow();

browserWindow.setAlwaysOnTop(top);

}

// 设置开机启动

static enableAutoStart(callback) {

let key = new WinReg({ hive: WinReg.HKCU, key: RUN\_LOCATION });

key.set('EUC', WinReg.REG\_SZ, file, (err) => {

console.log('设置自动启动' + err);

callback(err);

});

}

// 取消开机启动

static disableAutoStart(callback) {

let key = new WinReg({ hive: WinReg.HKCU, key: RUN\_LOCATION });

key.remove('EUC', (err) => {

console.log('取消自动启动' + err);

callback(err);

});

}

// 获取是否开机启动

static getAutoStartValue(callback) {

let key = new WinReg({ hive: WinReg.HKCU, key: RUN\_LOCATION });

key.get('EUC', function (error, result) {

console.log("查询自动启动:" + JSON.stringify(result));

console.log("file:" + file);

if (result) {

callback(true);

}

else {

callback(false);

}

});

}

/\*\*

\* 托盘图标闪烁

\* @param flash true：闪烁；false：停止

\*/

static flashTray(flash) {

let hasIcon = false;

const tayIcon = './imgs/logo.ico';

const tayIcon1 = './imgs/empty.png';

if (flash) {

if (flashTrayTimer) {

return;

}

flashTrayTimer = window.setInterval(() => {

ipcRenderer.send('ChangeTrayIcon', hasIcon ? tayIcon : tayIcon1);

hasIcon = !hasIcon;

}, 500);

} else {

if (flashTrayTimer) {

window.clearInterval(flashTrayTimer);

flashTrayTimer = null;

}

ipcRenderer.send('ChangeTrayIcon', tayIcon);

}

}

}

module.exports = WindowUtil;

复制代码

3、窗口大小的改变；

复制代码

添加一句话：BrowserWindow.mainWindow = mainWindow;不知道有用没有用

this.$router.push('/mainChat');//路由跳转mainChat

const remote = require('electron').remote;

const BrowserWindow = remote.BrowserWindow;

BrowserWindow.mainWindow.setSize(common.WINDOW\_SIZE\_MAINCIAT.width,common.WINDOW\_SIZE\_MAINCIAT.height);

BrowserWindow.mainWindow.center()

common.WINDOW\_SIZE\_MAINCIAT.width, 850

common.WINDOW\_SIZE\_MAINCIAT.height,600

复制代码

4.electron 的Api部分；

复制代码

win.setSize(width, height[, animate])

width Integer

height Integer

animate Boolean (可选) OS X

重新设置窗口的宽高值.

win.getSize()

返回一个数组，它包含了窗口的宽，高.

win.setContentSize(width, height[, animate])

width Integer

height Integer

animate Boolean (可选) OS X

重新设置窗口客户端的宽高值（例如网页界面）.

win.getContentSize()

返回一个数组，它包含了窗口客户端的宽，高.

win.setMinimumSize(width, height)

width Integer

height Integer

设置窗口最小化的宽高值.

win.getMinimumSize()

返回一个数组，它包含了窗口最小化的宽，高.

win.setMaximumSize(width, height)

width Integer

height Integer

设置窗口最大化的宽高值.

win.getMaximumSize()

返回一个数组，它包含了窗口最大化的宽，高.

win.setResizable(resizable)

resizable Boolean

设置窗口是否可以被用户改变size.

win.isResizable()

返回 boolean,窗口是否可以被用户改变size.

win.setMovable(movable) OS X Windows

movable Boolean

设置窗口是否可以被用户拖动. Linux 无效.

win.isMovable() OS X Windows

返回 boolean,窗口是否可以被用户拖动. Linux 总是返回 true.

win.setMinimizable(minimizable) OS X Windows

minimizable Boolean

设置窗口是否可以最小化. Linux 无效.

win.isMinimizable() OS X Windows

返回 boolean,窗口是否可以最小化. Linux 总是返回 true.

win.setMaximizable(maximizable) OS X Windows

maximizable Boolean

设置窗口是否可以最大化. Linux 无效.

win.isMaximizable() OS X Windows

返回 boolean,窗口是否可以最大化. Linux 总是返回 true.

win.setFullScreenable(fullscreenable)

fullscreenable Boolean

设置点击最大化按钮是否可以全屏或最大化窗口.

win.isFullScreenable()

返回 boolean,点击最大化按钮是否可以全屏或最大化窗口.

win.setClosable(closable) OS X Windows

closable Boolean

设置窗口是否可以人为关闭. Linux 无效.

win.isClosable() OS X Windows

返回 boolean,窗口是否可以人为关闭. Linux 总是返回 true.

win.setAlwaysOnTop(flag)

flag Boolean

是否设置这个窗口始终在其他窗口之上.设置之后，这个窗口仍然是一个普通的窗口，不是一个不可以获得焦点的工具箱窗口.

win.isAlwaysOnTop()

返回 boolean,当前窗口是否始终在其它窗口之前.

win.center()

窗口居中.

win.setPosition(x, y[, animate])

x Integer

y Integer

animate Boolean (可选) OS X

移动窗口到对应的 x and y 坐标.

win.getPosition()

返回一个包含当前窗口位置的数组.

win.setTitle(title)

title String

改变原窗口的title.

win.getTitle()

返回原窗口的title.

注意: 界面title可能和窗口title不相同.

win.flashFrame(flag)

flag Boolean

开始或停止显示窗口来获得用户的关注.

win.setSkipTaskbar(skip)