

DOCUMENTACIÓ ELECTRON I IPC

EVENTS



Marc Sans Vera

25 d'oct. de 2023

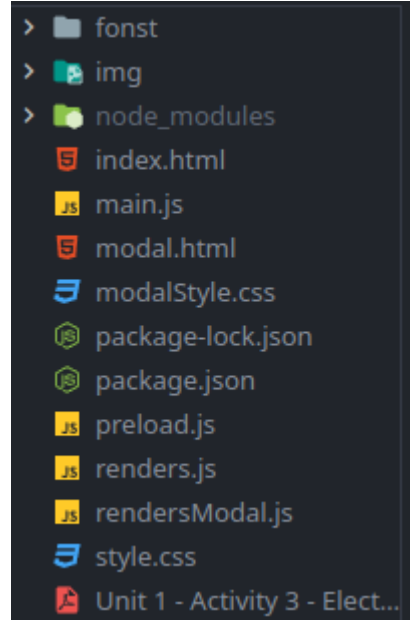
Índex

Implementació	3
Estructura del Projecte	3
Proceso Principal (main.js)	3
Función createWindow()	3
Función app.whenReady()	4
Evento 'window-all-closed'	4
Proceso de Renderizado (preload.js)	4
Función contextBridge.exposeInMainWorld()	4
Proceso de Renderizado (renderer.js)	5
Evento 'DOMContentLoaded'	5
Evento 'click' en el Botón	5
Proceso de Renderizado Modal (modalRenderer.js)	5
Evento 'DOMContentLoaded'	5
Evento 'click' en un Elemento	6

Implementació

Estructura del Projecte

Una vegada teniem creat l'html i css del projecte he plantejat una estructura de javascript de tres arxius, un arxiu main, on es creen i es carreguen les finestres i des d'on cridam les funcions que volem implementaer. Després tenim un arxiu readers per cada finestra, renders per la finestra principal i rendersModal per la finestra modal, on cream les funcions que necessitem i un arxiu preload que serveix que fa com a pont entre l'arxiu main i els dos arxius renders.



Proceso Principal (main.js)

Función createWindow()

- Crea una ventana principal de la aplicación.
- Configura la ventana principal con opciones como tamaño y archivo de precarga.
- Carga un archivo HTML en la ventana principal.
- Escucha el evento 'open-modal' para abrir una ventana modal.

```
// Función para crear la ventana principal
function createWindow() {
  const mainWindow = new BrowserWindow({ ...
});

  // Carga el archivo HTML principal en la
  mainWindow.loadFile('index.html');

  // Escucha el evento 'open-modal' para ab
  ipcMain.on('open-modal', () => { ...
});
}
```

Función `app.whenReady()`

- Devuelve una promesa que se resuelve cuando la aplicación Electron está lista para comenzar.
- Llama a `createWindow()` para crear la ventana principal.
- Configura un manejador de eventos para el caso en que la aplicación se active sin ventanas abiertas.

```
// Espera a que la aplicación esté lista
app.whenReady().then(() => {
  createWindow();

  // Maneja el evento 'activate' para
  app.on('activate', function () { ...
  });
});
```

Evento `'window-all-closed'`

- Maneja el evento `'window-all-closed'` que se activa cuando todas las ventanas están cerradas.
- Comprueba si el sistema operativo es diferente de macOS (darwin) y, si es así, cierra la aplicación.

```
// Maneja el evento 'window-all-closed' para salir
app.on('window-all-closed', function () {
  if (process.platform !== 'darwin') app.quit();
});
```

Proceso de Renderizado (`preload.js`)

Función `contextBridge.exposeInMainWorld()`

- Permite exponer funciones del proceso principal al proceso de renderizado de manera segura.
- Crea un objeto llamado `'electronAPI'` que contiene funciones específicas que pueden ser llamadas desde el proceso de renderizado.

```
// Expone funciones al mundo del proceso de renderizado
contextBridge.exposeInMainWorld('electronAPI', {
  // Función para abrir una ventana modal
  openModal: () => { ...
  },

  // Función para enviar datos del formulario al
  sendFormData: (formData) => { ...
  },

  // Función para realizar una acción relacionada
  profilesFunction: () => { ...
  },
});
```

Proceso de Renderizado (renderer.js)

Evento 'DOMContentLoaded'

- Escucha el evento 'DOMContentLoaded' que se dispara cuando el documento HTML se ha cargado completamente.
- Agrega un evento de envío de formulario al elemento con el ID 'subscriptionForm', evitando el envío predeterminado del formulario.
- Obtiene los datos del formulario, los convierte en un objeto y los envía al proceso principal cuando se envía el formulario.
- Obtiene referencias a varios elementos del formulario y agrega un evento de cambio al selector de usuario.

```
// Obtén el elemento del botón con el id 'openModalButton' en el documento
> document.getElementById('openModalButton').addEventListener('click', () => { ...
});
```

Evento 'click' en el Botón

- Agrega un evento de clic al botón con el ID 'openModalButton'.
- Cuando se hace clic en este botón, se llama a la función 'openModal' del objeto 'electronAPI' expuesto por el proceso de precarga. Esto abre una ventana modal en la aplicación.

Proceso de Renderizado Modal (modalRenderer.js)

Evento 'DOMContentLoaded'

- Este evento se dispara cuando el documento HTML de la ventana modal se ha cargado completamente.
- En este contexto, se utiliza para configurar interacciones y operaciones una vez que la ventana modal está lista.

```
document.addEventListener('DOMContentLoaded', () => {
  // Agrega un evento de envío de formulario al elemento con el ID 'subscriptionForm'
  document.getElementById('subscriptionForm').addEventListener('submit', (event) => { ...
});
```

Evento `click` en un Elemento

- Dentro del evento **'DOMContentLoaded'**, se pueden configurar manejadores de eventos de clic para elementos específicos en la ventana modal.
- Por ejemplo, podrías definir acciones adicionales cuando un botón se hace clic o cuando un elemento de formulario se selecciona.

```
// Agrega un evento de envío de formulario al elemento con el ID 'subscriptionForm'  
document.getElementById('subscriptionForm').addEventListener('submit', (event) => {...  
});  
  
// Obtiene referencias a varios elementos del formulario  
const userIdSelect = document.getElementById('userId'); // Se cambió a 'userId'  
const nameInput = document.getElementById('name');  
const emailInput = document.getElementById('email');  
const genderSelect = document.getElementById('gender'); // Se cambió a 'gender'  
const messageTextarea = document.getElementById('message');  
  
// Agrega un evento de cambio al elemento 'userIdSelect'  
userIdSelect.addEventListener('change', (event) => {...  
});
```