Pau Casesnoves
Centre Integrat de Formació Professional

C/Joan Miró, 22 07300 Inca
Tel. 971 881711
secretaria@paucasesnovescifp.cat

Electron IPC and Events

Curs	23/24	**Grup**	**S2P**	**Data lliurament**	25/10 - 23:55
Mòdul	Interfaces Development				
Títol	Electron IPC and Events				

Type	individual
Instructions	

Electron IPC and Events
To study the use of events and IPC (Inter-Process Communication) in Electron you will implement a simple application that uses a form to send data from a renderer process to the main process.

Requirements
- The form options will be populated with data from a public API.
- The form will be launched in a new modal window triggered by a button in the main window.
- All the form data has to be displayed in the main process's console.

Instructions
In this practice, you will explore Electron's events and IPC capabilities. You will create a simple application that consists of two main parts: a main process and a renderer process. The renderer process will launch a modal window containing a form for data entry.

Task 1: Set Up Your Electron Application
- Initialize a new Electron project using Electron's documentation here.
- Create two HTML files, one for the main window and one for the modal form window.
- Ensure that you use HTML5 semantic elements like <header>, <footer>, <article>, and <section>. Learn about HTML5 semantic elements here.

Task 2: Create the Main Window
- In your main window HTML, create a button that, when clicked, will open the modal form. Use the <button> element with an id attribute to select it easily in JavaScript.
- Implement an event listener for the button's click event. Refer to Electron's documentation on handling events in the renderer process here.

Pau Casesnoves
Centre Integrat de Formació Professional
C/Joan Miró, 22 07300 Inca
Tel. 971 881711
secretaria@paucasesnovescifp.cat

Task 3: Create the Modal Form

- In the modal form HTML, design a form containing different input elements (text fields, radio buttons, selectors, etc.). Make sure you use HTML5 form elements and provide appropriate labels.
- Create an event listener to capture the form submission. Refer to MDN Web Docs for HTML forms here.
- Once the form is submitted, gather the data and use IPC to send it to the main process. Refer to Electron's documentation on Inter-Process Communication here.

Task 4: Populate Form Options from a Public API

- Choose a public API to fetch data from. For example, you can use the JSONPlaceholder API (https://jsonplaceholder.typicode.com) to fetch sample data.
- Create a function in the renderer process that uses fetch or another suitable method to retrieve data from the API.
- Populate a dropdown (<select>) or other form elements with the data obtained from the API.

Task 5: Communicate with the Main Process

- After gathering form data in the renderer process, send this data to the main process using IPC.
- In the main process, set up an IPC event listener to receive and display the data on the console. Use Electron's IPC communication methods for this purpose.

Task 6: Testing and Debugging

- Test your application by clicking the button in the main window to open the modal form. Fill out the form, submit it, and verify that the data is displayed in the main process's console.

Deliverables

Submit your Electron project folder containing all the code and HTML files, along with a document explaining the process, your implementation, and any issues you encountered during the practice. Be sure to provide comments in your code for clarity.
Remember to consult the provided documentation links for further guidance.

Qualification criteria

This activity corresponds to 10% of the practical part.

Criterion 1: Application Setup (10%)

- 0 points: Application is not set up, or major errors exist.
- 1 points: Application setup is partially complete with several issues.
- 2 points: Application is set up correctly, but with minor issues or omissions.
- 3 points: Application is well-structured and fully functional.

C/Joan Miró, 22 07300 Inca
Tel. 971 881711
secretaria@paucasesnovescifp.cat

Criterion 2: Main Window and Event Handling (20%)
- 0 points: No main window or events implemented.
- 1 points: Main window and events are partially implemented with significant issues.
- 2 points: Main window and event handling are correctly implemented, but with minor issues.
- 3 points: Main window and event handling are correctly and efficiently implemented.

Criterion 3: Modal Form Creation (10%)
- 0 points: Modal form is not created, or major issues exist.
- 1 points: Modal form is partially created with significant issues.
- 2 points: Modal form is correctly created but with minor issues.
- 3 points: Modal form is well-designed and fully functional.

Criterion 4: Form Data Submission and IPC (20%)
- 0 points: Form data submission and IPC communication are not implemented.
- 1 points: Data submission and IPC are partially implemented with major issues.
- 2 points: Data submission and IPC are correctly implemented but with minor issues.
- 3 points: Data submission and IPC are correctly and efficiently implemented.

Criterion 5: API Data Retrieval and Integration (30%)
- 0 points: No API data retrieval or integration.
- 1 points: API data retrieval is partially implemented with significant issues.
- 2 points: API data retrieval and integration are correctly implemented but with minor issues.
- 3 points: API data retrieval and integration are well-implemented and functional.

Criterion 6: Code Quality and Comments (10%)
- 0 points: Code is messy, unreadable, and lacks comments.
- 1 points: Code is somewhat organized, with minimal comments.
- 2 points: Code is well-organized, but comments are sparse.
- 3 points: Code is well-structured, organized, and includes helpful comments for clarity.