

Mãos na Massa: Novo curso no Tray

Vamos agora fazer com que o nosso campo seja funcional. O primeiro passo é detectar um click no botão de +.

1- Em seu `renderer.js` adicione um evento de click, que captura os dados do campo e substitui o valor do curso atual pelo o do campo. Além disso, precisamos zerar o tempo atual, afinal se é um curso novo, seu tempo deve começar do zero.

```
//renderer.js

let botaoAdicionar = document.querySelector('.botao-adicionar');
let campoAdicionar = document.querySelector('.campo-adicionar');

...

botaoAdicionar.addEventListener('click', function(){
  let novoCurso = campoAdicionar.value;
  curso.textContent = novoCurso;
  tempo.textContent = '00:00:00';
  campoAdicionar.textContent = '';
})
```

Agora quando você clicar no botão de + o nome do novo curso deve surgir na tela e o tempo deve ser zerado.

2- Agora precisamos alterar o nosso Traybar para que o novo curso surja já também como uma opção. Para isto, vamos enviar um evento para o processo principal notificando que um novo curso foi adicionado:

```
//renderer.js

...

botaoAdicionar.addEventListener('click', function(){
  let novoCurso = campoAdicionar.value;
  curso.textContent = novoCurso;
  tempo.textContent = '00:00:00';
  campoAdicionar.textContent = '';
  ipcRenderer.send('curso-adicionado', novoCurso);
})
```

3- Vamos agora escutar o evento de 'curso-adicionado' em nosso processo principal:

```
//main.js
ipcMain.on('curso-adicionado', (event, novoCurso)=>{

});
```

4- Neste novo evento, temos que adicionar o `novoCurso` no template de nosso Menu. Para isto, vamos criar uma função `adicionaCursoNoTray()` em nosso template generator, que vai ser responsável de retornar um novo template de Menu, com o nome do curso lá. No `template.js` faça:

```
//template.js

module.exports = {
  ...

  adicionaCursoNoTray(curso){

  }
}
```

5- Como precisamos adicionar algo no template que já existe, precisamos guardar uma referência para ele quando ele é criado. Crie a variável `templateInical` que inicia com `null` no módulo `template.js` e ao final da função `geraTrayTemplate`, atribua um novo valor a ela.

```
//template.js

module.exports = {
  templateInical : null,
  geraTrayTemplate(win){
    ...
    this.templateInical = template;
    return template;
  },
  adicionaCursoNoTray(curso){

  }
}
```

6- Agora sim, com a referência ao `templateInical`, podemos modificá-lo. Basta adicionar um novo curso ao array do `templateInical`. Vale lembrar também que precisamos habilitar o evento de click em nossa função `adicionaCursoNoTray`, então além de receber o nome do curso, também precisamos da janela para enviar o click.

```
//template.js

module.exports = {
  templateInical : null,
  geraTrayTemplate(win){
    ...
  },
  adicionaCursoNoTray(curso,win){
    this.templateInical.push({
      label: curso,
      type: 'radio',
      checked: true,
      click: () => {
        win.send('curso-trocado', curso);
      }
    })

    return this.templateInical;
  }
}
```

7- Agora em nosso `main.js`, basta fazer uso da nossa nova função para colocar o novo Menu de Tray, já modificado com o curso novo, no local do antigo. Se a declaração da variável da sua `mainWindow` ainda estiver dentro do evento de `app.on('ready')`, mova-o para fora:

```
let mainWindow = null;
app.on('ready', ()=>{
  mainWindow = new BrowserWindow({
    width: 600,
    height: 400
  })
})
```

E no evento de curso `curso-adicionado` faça:

```
//main.js
ipcMain.on('curso-adicionado', (event, novoCurso)=>{
  let novoTemplate = templateGenerator.adicionaCursoNoTray( novoCurso, mainWindow );
  let novoTrayTemplate = Menu.buildFromTemplate(novoTemplate);
  tray.setContextMenu(novoTrayTemplate);
});
```

Agora você deve poder alterar entre os cursos novos adicionados e o seus antigos.