05

Utilizando Web API's em seu app

Transcrição

Por último, gostaríamos de saber quando o *timer* foi iniciado, pois podemos utilizar o atalho **CmdOrCtrl+Shift+S** e não sabemos se o *timer* foi iniciado, nem se a aplicação está aberta e o atalho funcionou.

Para notificar o usuário quando o atalho for executado, vamos utilizar as notificações do próprio sistema operacional. Essas notificações são como Web API's, notificações da web, mas sem precisarmos pedir permissão. Normalmente, só podemos exibir uma notificação em um site, quando o usuário der permissão, assim como o acesso ao microfone, vídeo, entre outros.

Mas quando estamos trabalhando com aplicativos desktop, supõe-se que o após instalá-lo, o usuário já deu essas permissões. Podemos exibir uma notificação do seguinte modo:

```
new Notification('Vai!!!');
```

Assim exibimos a notificação do sistema operacional para o usuário, inclusive com o ícone do aplicativo.

Exibindo notificações para o usuário

Para exibir a notificação quando o *timer* for iniciado ou parado, devemos adicionar o código de notificação no **renderer.js**, onde estamos escutando o evento `click` do botão. Adicionaremos uma notificação com o título **Alura Timer**:

```
// renderer.js
// restante do código omitido

botaoPlay.addEventListener('click', function () {
    if(play) {
        timer.parar(curso.textContent);
        play = false;
        new Notification('Alura Timer');
    } else {
        timer.iniciar(tempo);
        play = true;
        new Notification('Alura Timer');
    }
    imgs = imgs.reverse();
    botaoPlay.src = imgs[0];
});
```

Além do título, podemos configurar outras coisas na notificação. Podemos passar um objeto para mesma, e configurar o corpo de texto da notificação através da propriedade `body` :

```
// renderer.js
// restante do código omitido
```

```
botaoPlay.addEventListener('click', function () {
    if(play) {
        timer.parar(curso.textContent);
        play = false;
        new Notification('Alura Timer',{
            body: `O curso ${curso.textContent} foi parado!!`
        });
    } else {
        timer.iniciar(tempo);
        play = true;
        new Notification('Alura Timer',{
            body: `O curso ${curso.textContent} foi iniciado!!`
        });
    }
    imgs = imgs.reverse();
    botaoPlay.src = imgs[0];
});
```

Do mesmo jeito, podemos adicionar um ícone ao lado da notificação, através da propriedade `icon`. Os ícones que vamos utilizar podem ser baixados aqui (https://s3.amazonaws.com/caelum-online-public/electron/cap08/stages/download-cap08.zip) e devem ser colocados dentro da pasta **alura-timer/app/img**:

```
// renderer.js
// restante do código omitido

botaoPlay.addEventListener('click', function () {
    if(play) {
        timer.parar(curso.textContent);
        play = false;
        new Notification('Alura Timer',{
            body: `O curso ${curso.textContent} foi parado!!`,
            icon: 'img/stop-button.png'
        });
    } else {
        timer.iniciar(tempo);
        play = true;
        new Notification('Alura Timer',{
            body: `O curso ${curso.textContent} foi iniciado!!`,
            icon: 'img/play-button.png'
        });
    }
    imgs = imgs.reverse();
    botaoPlay.src = imgs[0];
});
```

Agora, sempre que a aplicação for iniciada ou parada, o usuário é notificado por uma notificação nativa do sistema operacional. Mas além das notificações, como estamos trabalhando com Electron, podemos utilizar todas as Web API's, sem pedir permissão do usuário, deixando o nosso aplicativo com uma usabilidade melhor.