

ngCordova e recurso de câmera

Transcrição

Agora que já conseguimos emular a nossa aplicação diretamente no nosso dispositivo, nós conseguimos acessar recursos nativos do nosso celular - como a câmera ou vibração. Nós conseguimos emular essas funcionalidades só no celular. Como faremos para ter acesso a esses recursos nativos?

Vamos dar uma olhada no projeto **ngCordova**. Teremos que instalar um plugin feito para o Cordova. O ngCordova encapsula plugins do Cordova e o Angular para facilitar a vida do desenvolvedor.

Mais especificamente, queremos usar os recursos da câmera. Então, além de instalar o Cordova, iremos instalar o plugin da câmera. Primeiramente, vamos instalar o **ngCordova** (<http://ngcordova.com/>). Nós conseguiremos instalar a `lib` com o `bower`. Você encontrará mais instruções para a instalação na [documentação](http://ngcordova.com/docs/install/) (<http://ngcordova.com/docs/install/>).

Vamos rodar o comando no Terminal:

```
AluraCar bower install ngCordova
```

A `lib` será instalada. Para encontrarmos o nosso projeto, teremos que seguir o seguinte caminho especificado na documentação: `www -> lib -> ngCordova -> ng-cordova.js`. Encontraremos também o `ng-cordova.min.js`, que é mais fácil de carregar. Nós iremos adicioná-lo no `index.html`, antes do `cordova.js`.


```
<!-- cordova script (this will be a 404 during development) -->
<script src="lib/ngCordova/dist/ng-cordova.min.js" ></script>
<script src="cordova.js"></script>
```

Em seguida, iremos injetar o `ngCordova` como dependência no `app.js`. Já temos o `ionic`, `idf.br-filters` e vamos instalar a base que é o `ngCordova`.

```
angular.module('starter', ['ionic', 'idf.br-filters', 'ngCordova'])
```

Agora, vamos instalar um plugin, veremos no site como fazer isso. As instruções podem ser encontradas na [documentação](http://ngcordova.com/docs/plugins/camera/) (<http://ngcordova.com/docs/plugins/camera/>):

New! Check out [Ionic Native](#). It's like ngCordova but for ES6 and TypeScript.

 ngCordova

[DOCS](#) [CUSTOM BUILD](#) [IONIC](#) [GITHUB](#)




OVERVIEW

INSTALL

COMMON ISSUES

PLUGINS

- 3D Touch
- Action Sheet
- AdMob
- App Availability
- App Rate
- App Version
- Badge
- Background Geoloc
- Battery Status

\$cordovaCamera    [Source](#) [Official Docs](#)

This service makes it easy to use the [cordova-plugin-camera](#) plugin to take pictures and video from a device.

NOTE: The camera API only works on a real device, and not in the emulator.

```
cordova plugin add cordova-plugin-camera
```

Methods

[getPicture\(options\)](#)

[View Camera Options](#)

Param	Type	Detail
options	Object	Camera options

Options **Type** **Detail**

Usaremos o seguinte comando no Terminal:

```
cordova plugin add cordova-plugin-camera
```

Depois, usaremos as instruções da documentação.

Adicionaremos o botão da câmera na tela de perfil, mas ele também seguirá a regra do botão "Editar", ou seja, só aparecerá quando estivermos editando o perfil. Faremos as alterações no `perfil.html`, dentro da `class="item item-avatar"`:

```
<button ng-show="estaEditando" class="button button-full button-positive" ngclick="tirarFoto()"></button>
```

Ao clicarmos no botão "Editar", iremos disparar a ação como o `ng-click="tirarFoto()"`. Precisaremos criar a função na `controller` da `View`. No arquivo, `controllers.js`, adicionaremos o `$scope.tirarFoto`:

```
angular.module('starter')
.controller('PerfilController', function($rootScope, $scope, $cordovaCamera){
  $scope.estaEditando = false;
  $scope.textoBotao = 'Editar';

  $scope.usuarioLogado = $rootScope.usuario;

  $scope.tirarFoto = function(){

    var opcoes = {

    }

    $cordovaCamera.getPicture(opcoes).then(function(foto){
    }, function (erro){

    })

  }
})
```

```
$scope.acaoBotao = function(){
```

```
$scope.estaEditando = true;  
}  
})
```

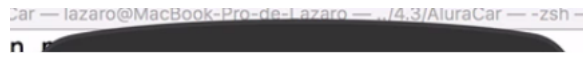
Chamamos a câmera injetando no `function` o `$cordovaCamera`. Depois, usamos o `$cordovaCamera` e função `getPicture`, conforme foi descrito na documentação. Para o caso de que dê erro, adicionamos o `function(erro)`. Após salvarmos as alterações em todos os arquivos, faremos o teste. Mas não será possível fazê-lo no Ionic Lab, teremos que fazer diretamente no dispositivo.

No Terminal usaremos o seguinte comando:

```
AluraCar ionic run android --device
```

Lembrando que o celular deve estar no modo desenvolvedor e deve estar conectado ao computador.

A instalação será feita com sucesso. Após clicarmos no botão "Editar", surgirá o botão "Tirar Foto".



A câmera será aberta e nós poderemos tirar a foto.

tion see https://docs.gradle.org/current/userguide/build_environment.html

Seguiremos com nossos estudos.