

APPMOB - Image Upload

Olivier Liechti & Simon Oulevay
COMEM Applications Mobiles

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Using the Cordova Camera API

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

<http://learn.ionicframework.com/formulas/cordova-camera/>

Add the Plugin

Add the plugin (from your project directory):

```
$> cordova plugin add org.apache.cordova.camera
```

Create a service to wrap the native functionality:

```
.factory("CameraService", function($q) {  
  return {  
    getPicture: function(options) {  
      var deferred = $q.defer();  
  
      navigator.camera.getPicture(function(result) {  
        // do any magic you need  
        deferred.resolve(result);  
      }, function(err) {  
        deferred.reject(err);  
      }, options);  
  
      return deferred.promise;  
    }  
  }  
});
```

Take Pictures

```
.controller("AnyController", function(CameraService) {  
  
    CameraService.getPicture({  
        quality: 75,  
        targetWidth: 400,  
        targetHeight: 300,  
        destinationType: Camera.DestinationType.DATA_URL  
    }).then(function(imageData) {  
        // do something with imageData  
    });  
  
});
```

Remember!

Taking pictures will only work on an actual phone or tablet. It will **not** work in your browser or in an emulator.

Adding Images to Issues

POST

/issues

×

Description

Request

Response

Headers

- **Authentication header** (*string*)
Use this header to specify the authentication.

roles required: any

Example: x-user-id 54d8ae183fd30364605c8401

Body

Type: application/json

Example:

```
{
  "description": "Integer at metus vitae erat porta pellentesque.",
  "lng": "6.651479812689227",
  "lat": "46.77227088657382",
  "imageUrl": "http://www.somewhere.localhost.localdomain",
  "issueTypeId": "54d8ae183fd30364605c81b1"
}
```

The API only accepts an image URL.

So you have to:

- take the picture
- upload it somewhere and get an URL
- create the issue with the image URL

Where to Upload?

Flickr?

Image Shack?

Amazon S3?

Quick Image Service

GitHub Repository

<https://github.com/SoftEng-HEIGVD/qimg/>

API Documentation

<http://softeng-heigvd.github.io/qimg/>

API Endpoint

<https://warm-bastion-3094.herokuapp.com/api>

Images cannot be larger than 2MB.

You cannot store more than 25 images (per user).

Ask for an authentication token by e-mail:

simon.oulevay@heig-vd.ch

You will need to add some configuration:

In:

- config/development.json
- config/production.json

```
{  
  // ...  
  "qimgUrl": "...",  
  "qimgToken": "...",  
}
```

In constants.js (root directory of the app):

```
angular.module("citizen-engagement.constants", [])  
  // ...  
  .constant("qimgUrl", "@qimgUrl@")  
  .constant("qimgToken", "@qimgToken@")  
;
```

In gulpfile.js:

```
// ...  
.pipe(replace(/@qimgUrl@/, config.qimgUrl))  
.pipe(replace(/@qimgToken@/, config.qimgToken))
```

Don't forget to re-run **gulp config-development** or **gulp config-production** to apply the new configuration.

In development, you will need a proxy.

In `config/development.json`, the URL should be:

```
{  
  // ...  
  "qimgUrl": "http://localhost:8100/qimg-proxy",  
}
```

Add a proxy to `ionic.project`:

```
{  
  "name": "...",  
  "proxies": [  
    { ... },  
    {  
      "path": "/qimg-proxy",  
      "proxyUrl": "https://warm-bastion-3094.herokuapp.com/api"  
    }  
  ]  
}
```

Restart `ionic serve` if it's running.

Sample Code

```
.controller("AnyController", function(CameraService, $http, qimgUrl, qimgToken) {

    // take the picture
    CameraService.getPicture({
        quality: 75,
        targetWidth: 400,
        targetHeight: 300,
        // return base64-encoded data instead of a file
        destinationType: Camera.DestinationType.DATA_URL
    }).then(function(imageData) {

        // upload the image
        $http({
            method: "POST",
            url: qimgUrl + "/images",
            headers: {
                Authorization: "Bearer " + qimgToken
            },
            data: {
                data: imageData
            }
        }).success(function(data) {

            var imageUrl = data.url;
            // do something with imageUrl
        });
    });
});
```