APPMOB - Image Upload

Olivier Liechti & Simon Oulevay COMEM Applications Mobiles



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

Using the Cordova Camera API



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



Add the plugin (from your project directory):

\$> cordova plugin add cordova-plugin-camera

Create a service to wrap the native functionality:

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
   });
});
```

Native Functionality

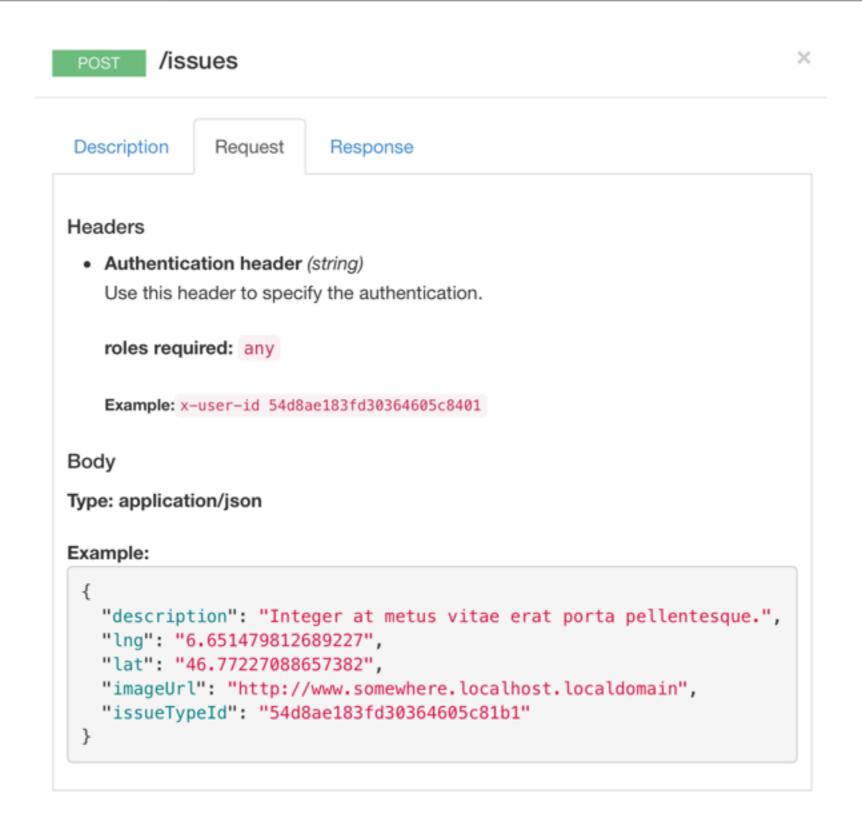


Remember!

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







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

qimg Limitations



Images cannot be larger than 2MB.

You cannot store more than 50 images (per user).

qimg API Credentials



Ask for an authentication token by e-mail:

simon.oulevay@heig-vd.ch

qimg Configuration



You will need to add some configuration:

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:

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
   });
 });
```