

## html5 教程：使用 HTML5 实现拍照上传应用



视频教程: [http://www.jikexueyuan.com/course/lists.html?skey=6---0&hmsr=wenku\\_html5](http://www.jikexueyuan.com/course/lists.html?skey=6---0&hmsr=wenku_html5)

[Swift 教程](#) [安卓开发教程](#) [ios 开发教程](#) [cocos2dx 教程](#) [手机应用开发](#) [游戏开发教程](#)

在 HTML5 规范的支持下，WebApp 在手机上拍照已经成为可能。在下面，我将讲解 Web App 如何用手机进行拍照，显示在页面上并上传到服务器。

## 1、 视频流

HTML5 The Media Capture API 提供了对摄像头的可编程访问，用户可以直接用 getUserMedia 获得摄像头提供的视频流。我们需要做的是添加一个 HTML5 的 Video 标签，并将从摄像头获得视频作为这个标签的输入来源（请注意目前仅 Chrome 和 Opera 支持 getUserMedia。补充：请使用 Opera 支持 HTML5 的新版本，如果是 Chrome，版本需为 Chrome 18.0.1008+，并使用 about:flags 来开启 WebRTC，请看下图。）。

[html][view plaincopy](#)

```
1 <video id="video" autoplay=""></video>
2 <script>
3   var video_element = document.getElementById('video');
4   if (navigator.getUserMedia) { // opera should use opera.getUserMedia now,
chrome use webkitGetUserMedia
5       navigator.getUserMedia('video', success, error);
6   }
7   function success(stream) {
8       video_element.src = stream;
9   }
10 </script>
```



视频流

## 2、 拍照

拍照功能，我们采用 HTML5 的 Canvas 实时捕获 Video 标签的内容，Video 元素能作为 Canvas 图像的输入，这一点很棒。主要代码如下：

[html][view plaincopy](#)

```
11 <script>
12     var canvas =document.createElement(' canvas');
13
14     var ctx = canvas.getContext(' 2d');
15     var cw = vw;
16     var ch = vh;
17     ctx.fillStyle = "#ffffff";
18     ctx.fillRect(0, 0, cw, ch);
19     ctx.drawImage(video_element, 0, 0, vvw, vvh, 0,0, vw, vh);
20     document.body.append(canvas);
21 </script>
```

### 3、 图片获取

下面我们要从 Canvas 获取图片数据，其核心思路是用 canvas 的 toDataURL 将 Canvas 的数据转换为 base64 位编码的 PNG 图像，类似于 “” 的格式。

[html][view plaincopy](#)

```
22 var imgData =canvas.toDataURL("image/png");
```

因为真正图像数据是 base64 编码逗号之后的部分，所以我们实际服务器处理的图像数据应该是这部分，我们可以用两种办法来获取。

第一种：是在前端截取 22 位以后的字符串作为图像数据，例如：

[html][view plaincopy](#)

```
23 var data = imgData.substr(22);
```

如果要在上传前获取图片的大小，可以使用：

[html][view plaincopy](#)

```
24 var length = atob(data).length;// atob decodes a string of data which has been
encoded using base-64 encoding
```

第二种：是在后端获取传输的数据后用后台语言截取22位以后的字符串。例如 PHP 里：

[html][view plaincopy](#)

```
25 $image = base64_decode( str_replace(' data:image/jpeg;base64,', '', $data);
```

#### 4、 图片上传

在前端可以使用 Ajax 将上面获得的图片数据上传到后台脚本。例如使用 jQuery 时：

[html][view plaincopy](#)

```
26 $.post('upload.php', { 'data' : data } );
```

在后台我们用 PHP 脚本接收数据并存储为图片。

[html][view plaincopy](#)

```
27 function convert_data($data) {  
28     $image = base64_decode( str_replace(' data:image/jpeg;base64,',  
29     save_to_file($image);  
30 }  
31 function save_to_file($image) {  
32     $fp = fopen($filename, 'w');  
33     fwrite($fp, $image);  
34     fclose($fp);  
35 }
```

请注意，以上的解决方案不仅能用于 Web App 拍照上传，并且你可以实现把 Canvas 的输出转换为图片上传的功能。这样你可以使用 Canvas 为用户提供图片编辑，例如裁剪、上色、涂鸦的画板功能，然后把用户编辑完的图片保存到服务器上。



## Canvas 涂鸦功能

在HTML5的驱动下，Web App 与 Native App 之间是否还有不可逾越的鸿沟？我将会在3月23日百度开发者大会上现场来解答这个问题，敬请期待。

### 相关规范：

The MediaCapture API: <http://www.w3.org/TR/media-capture-api/>

Canvas: <http://dev.w3.org/html5/2dcontext/>

### 相关 The Media Capture 示例：

[Live Photo booth](#)

[Instant Camera](#)

原文地址: <http://blog.csdn.net/hfahe/article/details/7354912>