Image Gallery Configuration

Image gallery screens allow you to showcase images in a compelling, device-native way. There are four different ways to provide the images for a gallery.

1: From URL's you manually enter in your buzztouch control panel. One URL for each image.

2: From a URL that points to a file on your website. This file may or may not be dynamically generated. It provides the list of images for the gallery.

3: From a Flickr™ photo feed. The photo-feed provides the list of images for the gallery.

4: From image files you add to your project after downloading the source-code. We call these “embedded images.”

How To

1: To manually enter images using URL's, simply enter the URL to each image in your buzztouch control panel. Add as many images as you need.

2: To feed your image gallery from a feed on your website, create a plain-text file, or a server-side-script that outputs data in JSON format. The text file must contain one parent objects called “childItems” and from zero to many image objects. Each image object has these properties (omit the image title if you don't have one).

“itemId”:”unique id for the image”
“itemType”:”BT_imageItem”
“imageTitle:”Johnny and Mary in the mountains”
“imageName”:”the name of the image in the project”
“imageURL”:”the URL to the online image”

Property names are case-sensitive. A sample output with three images would look like this:

```
{"childItems":[

        {"itemId":"12341234", "itemType":"BT_imageItem", “imageTitle”:”title 1”, "imageName":"myCoolImage.png", "imageURL":"" },
        {"itemId":"CEFB3C3", "itemType":"BT_imageItem", ”imageTitle:”title 2”, "imageName":"", "imageURL":"http://....neatimage1.png" },
        {"itemId":"92224499", "itemType":"BT_imageItem", “imageTitle”:”title 3”, "imageName":"", "imageURL":"http://....neatimage2.png" }
]}
```

The first image is included in the project and the other two are downloaded from the imageURL. Images can be .JPG or .PNG and the size is flexible. Generally, it's best to provide reasonably sized images that don't use huge amounts of storage on the users device. If you're using images that are 2500 X 3000 you should re-consider your plan. Large images consume large amounts of memory and storage space on small devices.

Note: If you are providing the images from your own URL, you may want to use merge fields in the URL. Merge fields allow you to serve different content for different reasons. For example, you may want to learn the latitude and longitude of the device before outputting the list of images. Available merge fields are:

userId: The id of the logged in user if you use a login screen.
userEmail: The email of the logged in user, if you use a login screen.

deviceLatitude: The latitude location of the device in decimal format.
deviceLongitude: The longitude of the device in decimal format.
deviceId: The unique id of the device.
deviceModel: The model of the device as provided by the manufacturer.

To use one or more merge fields, surround the name of the merge field with square brackets like this:

http://www.mywebsite.com/images.php?deviceId=[deviceId]&deviceLatitude=[deviceLatitude]

3: **F**lickr™ photo feeds are very flexible. When using Flickr™ feeds, you must set the output to JSON. A sample feed that pulls 25 public images associated with the keyword “Golf, Tennis, Bowling” looks like this:

http: //api.flickr.com/services/rest/?&method=flickr.photos.search&api_key=API-KEY-GOES-HERE&format=json&tags=Golf, Tennis, Bowling&privacy_filter=1&content_type=1&per_page=25

Enter the URL for the feed in the dataURL property in the Gallery Advanced Settings. Use merge fields to send latitude / longitude info in the URL:

http: //api.flickr.com/services/rest/?&method=flickr.photos.search&api_key=API-KEY-GOES-HERE&format=json&tags=Golf, Tennis, Bowling&privacy_filter=1&content_type=1&per_page=25&lat=[deviceLatitude]&lon=[deviceLongitude]

There are hundreds of ways to use Flickr™ and their powerful API. You will need to obtain a Flickr™ API key before pulling images from their backend. API keys are free and simple to setup. See http://www.flickr.com/services/api/ for more information.

4: Embedding images inside your downloaded project may be appropriate. Generally, it's best to use images from URL's if you can because it is far more flexible. Also, embedding images in your project increases it's size. Large projects tend to take longer to download from the App Store and some won't download at all without a Wi-Fi connection. This means that if you compile your project with 50 high-quality images included, it will be very very large and take a very very long time for end-users to download.

To embed images:

a) Create a folder on your computer and add some images to it.

b) Drag this folder into your Xcode or Eclipse project. Confirm if a dialogue opens and asks you if you want to “Add these files to your project.” It's important that you add them to your project environment, it's not enough to only add them to the downloaded source-code folder.

c) On your buzztouch control panel, enter the image name for each image you included in your project. Do not enter a name AND a URL for an image – it's a name OR a URL, not both.

Refreshing Images: If you decide to show a refresh button on the Image Gallery screen, it wont' have any effect unless you have provided a data-URL (your website, Flickr™, etc).