Image Architecture for Posts

Images are input to the system by selecting the image from a file selection tool. The collected image has a shrunken version placed in the icons field of a Meteor Post object, which is sent to the Meteor server as base64 encoded string.

The client also sends the shrunken object, along with the original image to the Meteor server using Meteor methods. The meteor server takes the images and places them on an Amazon S3 service, and replaces the appropriate icons or images field with the http address of the image on the server.

The POST document structure would have a URL list added to it such as:

**const** values = {  
 [FIELDS.EMAIL]: emailAddress,  
 [FIELDS.COMMENTS]: [],

[FIELDS.IMAGES]: images,  
 [FIELDS.ICONS]: [], 🡨 new field, indicates the image URLs of the icons,

starts out as base64 encoded image string.

[FIELDS.LIKERS]: [],  
 [FIELDS.LIKES]: 0,  
 [FIELDS.TEXT]: text,  
 [FIELDS.TOPIC]: topic,  
 [FIELDS.CREATED\_AT]: **new** Date()  
};

Image sequence:

* Select image for Post from phone file system.
* Put shrunken version of encoded image data into the POST ICONS field
* Send shrunken and original images to server using Meteor.method() where:
  + Creates a filename https address for the image
  + Sends the image to the image server
  + Updates the IMAGES or ICONS field of the Post with the URL to fetch the image
* Retrieve images using the URL

Meteor

Method

Image Service

Client

Image

PostObj

Server

PostObj

Image

Icon

Full

Image

Images

Display

Image

Set URL(s)

Base64

Encoded

Retrieve

Image

Video data

Video will be stored on uTube. It will be up to the administrator building html content to embed the link to the video in the content. Such as:

<**iframe width="854" height="480" src="https://www.youtube.com/embed/fxIgkgBbpQg" frameborder="0" allowfullscreen**></**iframe**>

They will have to be careful to grab the ‘embed’ URL for the video.