Variable Avatar Images

Below are three methods for creating variable Avatar images within the Display Card for a given entity.

# Simple Variable Replacement

Typical Avatar image address will grab a file from the image library, and will look something like this:

images://<libraryName>/<image.png>

If you have a set of images for something like Categories, or Types, then name each image after the category or type that it matches up to, and concatenate it all together, like this:

‘images://<libraryName>/’ || <Attribute> || ‘<.png>’

A couple of notes about the above:

* Replace anything within < > with your specific data, the < > should not be there when you’re done
* The filetype may or may not be PNG. Update that according to the image file you’re using

# Complicated Variable Replacement

Typical Avatar image address will grab a file from the image library, and will look something like this:

images://<libraryName>/<image.png>

If you have a set of images for something like Categories, or Types that DO NOT match up with the attribute values, then a more complex Case statement will need to be created and used:

case

when <Attribute> = ‘<imageName1>’ then

‘images://<libraryName>/<imageName1.png>’

when <Attribute> = ‘<imageName2>’ then

*‘images://<libraryName>/<imageName2.png>’*

else null

end

A couple of notes about the above:

* Replace anything within < > with your specific data, the < > should not be there when you’re done
* The filetype may or may not be PNG. Update that according to the image file you’re using
* You will need to create a “when” line for each of the different attribute types/images, so this can become quite a large snippet of code, quickly

# Third Party Logo Enrichment

Typical Avatar image address will grab a file from the image library, and will look something like this:

images://<libraryName>/<image.png>

If you are looking for specific logos for your Customer/Company data, there is a third party API set that is available to us called Clearbit. To use it, we simply need to use a URL in the following syntax:

https://logo.clearbit.com/<companyWebsite>

And example of this would be <https://logo.clearbit.com/goamplifi.com>

As you can see when clicking on this link, the Amplifi logo will appear in the browser. We can utilize that within the Semarchy Application by drawing upon existing Website fields, and then doing a bit of Regular Expression work in order to remove extra characters from the website. I’ll show an example here below, and then explain a few steps:

case

when (lookup first VirtualAddress from VirtualAddresses

where FDN\_VirtualAddressType ='Website'

Order By UpdateDate DESC end) is not null then

'https://logo.clearbit.com/' ||

RTRIM(REGEXP\_REPLACE(LOWER(lookup first

VirtualAddress from VirtualAddresses where

FDN\_VirtualAddressType ='Website' Order By

UpdateDate DESC end),

'^(https?://)?(www\.)?',''), '/')

when BrandLogoURL is not null then BrandLogoURL

else 'images://<libraryName>/<imageName.png>'

end

A couple of notes about the above:

* Replace anything within < > with your specific data, the < > should not be there when you’re done
* The filetype may or may not be PNG. Update that according to the image file you’re using

## How it works

* The first part is a lookup against an Entity called VirtualAddresses. It might be where the Website information is being provided in our Data Model. Website might also be a static field. However, the point is that if Website “is not null” then we are going to take our baseline Clearbit URL, and concatenate a value onto it.
* We need to do some work to our Website however, because it may have https://www still contained within it, and *https://logo.clearbit.com/https://www.goamplifi.com/* doesn’t quite work.
* We utilize a Regular Expression and a RTRIM to help us remove those extra characters from the website URL, leaving us with our clean example from the top of this page.
* If we don’t have any Websites in our website field, the BrandLogoURL shows an example where a specific link to an image on the web, such as <https://www.goamplifi.com/wp-content/uploads/2021/06/Amplifi-Lines-WhitePink.png> can be used as a backup
* Lastly, if neither of the above options works, then we make an ELSE statement that uses our traditional pattern grabbing an image from the image library as our fallback image.