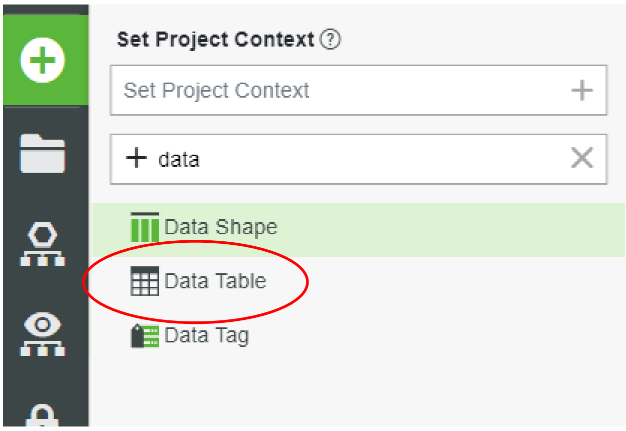
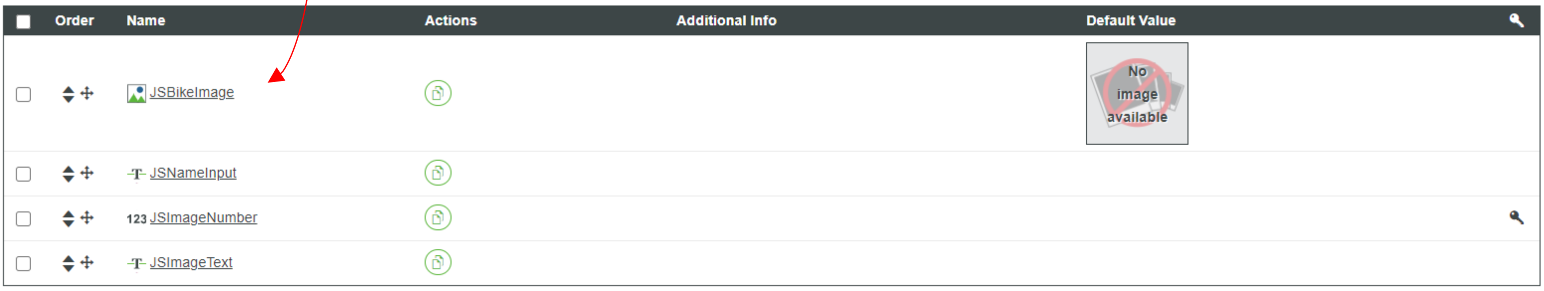
Sending Vuforia Studio data to a Thingworx Data Table

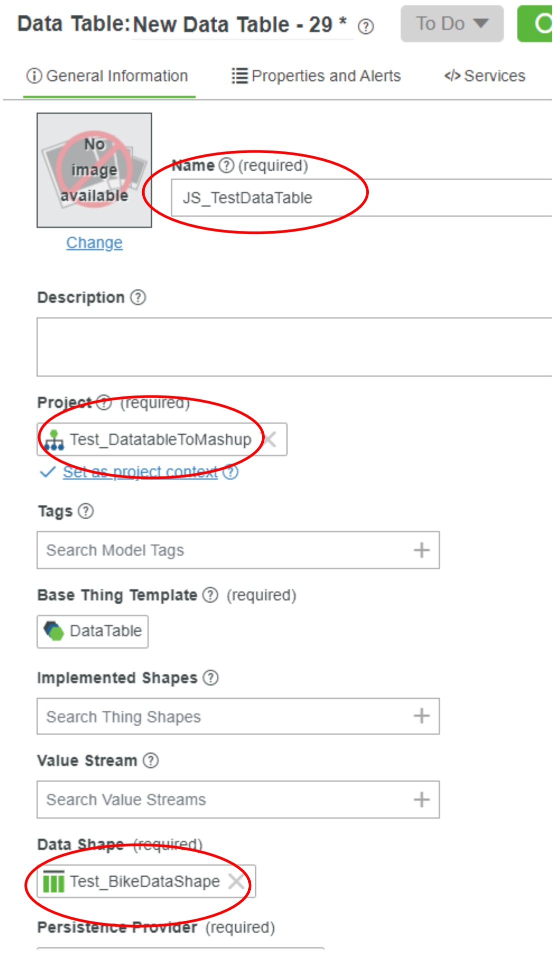
1. Open Thingworx and create a new DataTable → DataTable



   a. Name = FirstInitials_TestDataTable (ex. JS_TestDataTable)
   b. Add a Project and a DataShape. (In this example I use a previously made project: Test_DatatableToMashup and DataShape: Test_BikeDataShape). You can create your own Project or use PTCDefaultProject. Your DataShape should have 4 field definitions:
      i. An Image (name it InitialsBikeImage: ex. JSBikeImate
      ii. A String named InitialsNameInput: ex. JSNameInput.
      iii. An Integer named InitialsImageNumber: ex. JSImageNumber
      iv. A String named InitialsImageText: ex. JSImageText
   c.



■	Order	Name	Actions	Additional Info	Default Value
□	⬍ ✢	JSBikeImage	⎘		No image available
□	⬍ ✢	JSNameInput	⎘		
□	⬍ ✢	123 JSImageNumber	⎘		
□	⬍ ✢	JSImageText	⎘		



Data Table: New Data Table - 29 * ⓘ To Do ▼

ⓘ General Information ☰ Properties and Alerts </> Services

Change

Name ⓘ (required): JS_TestDataTable

Description ⓘ

Project ⓘ (required): Test_DatatableToMashup

✓ Set as project context ⓘ

Tags ⓘ: Search Model Tags +

Base Thing Template ⓘ (required): DataTable

Implemented Shapes ⓘ: Search Thing Shapes +

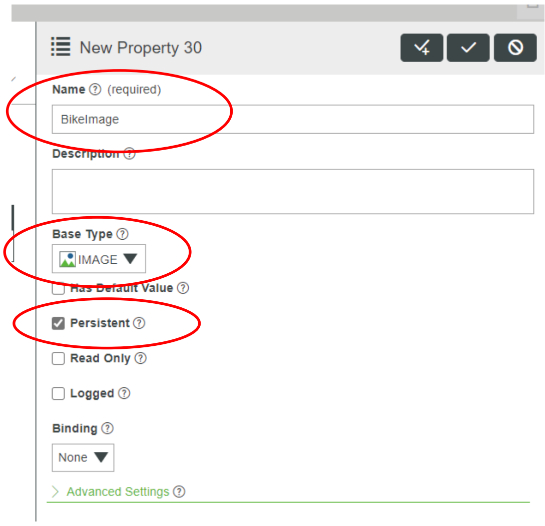
Value Stream ⓘ: Search Value Streams +

Data Shape (required): Test_BikeDataShape

Persistence Provider (required)

2. Go to Properties and Alerts, add 4 properties
    a. Name = BikeImage → BaseType = Image → check box Persistent
    b. Name = ImageText → BaseType = String →check box Persistent
    c. Name = NameInput → BaseType = String → check box Persistent

d. Name = PictureNumber → BaseType = integer → check box Persistent



New Property 30

Name (required)

BikeImage

Description

Base Type

IMAGE

☐ Has Default Value

☑ Persistent

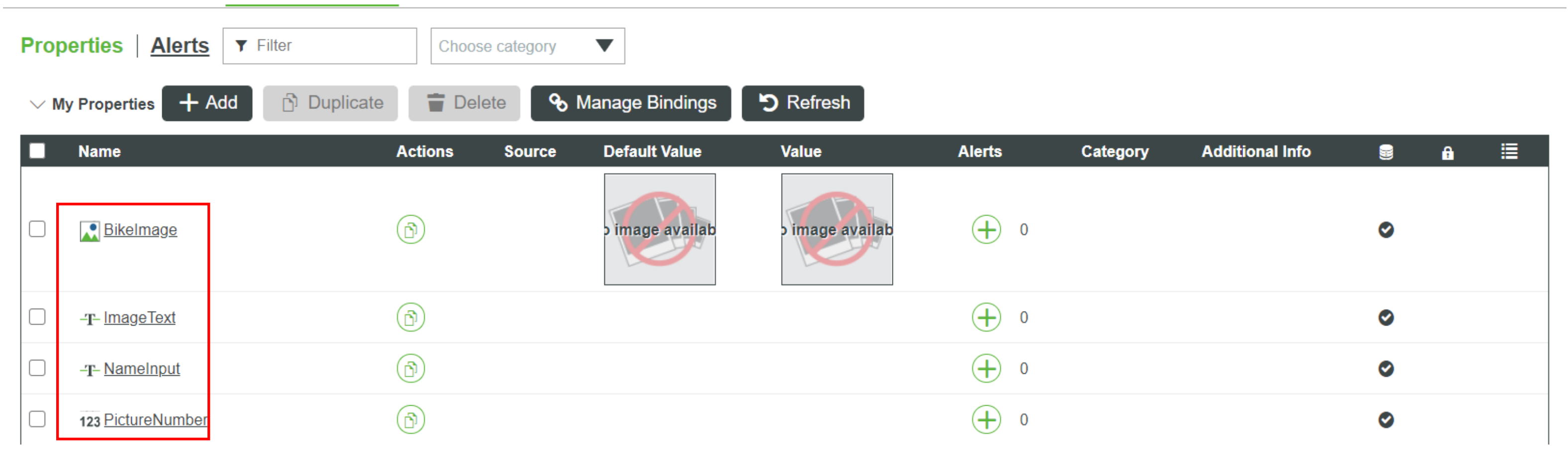
☐ Read Only

☐ Logged

Binding

None

Advanced Settings

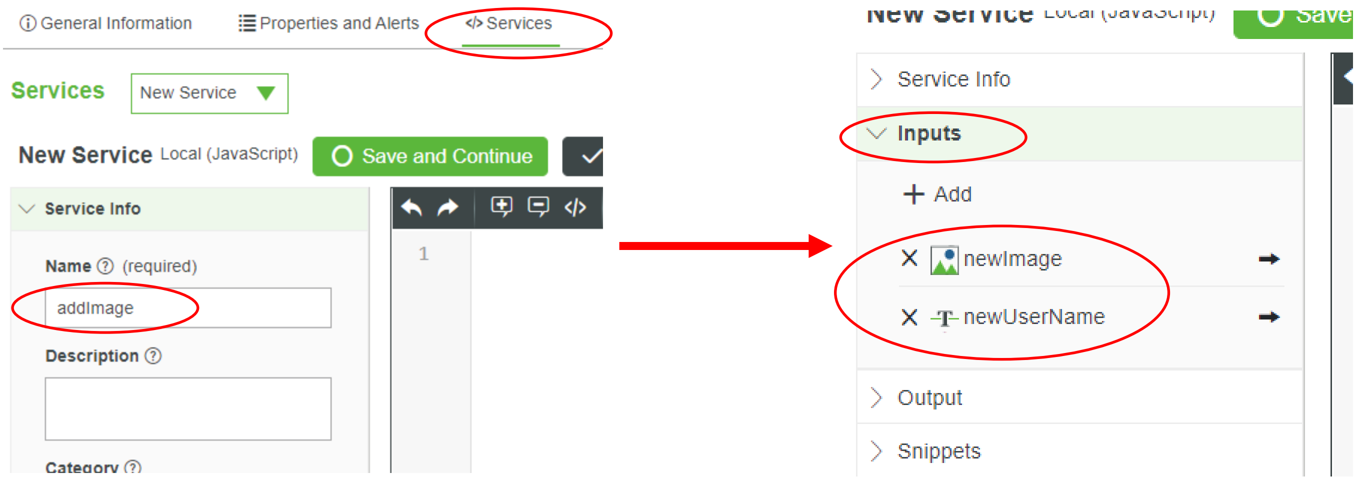


Properties | Alerts | Filter | Choose category

My Properties | + Add | Duplicate | Delete | Manage Bindings | Refresh

☐	Name	Actions	Source	Default Value	Value	Alerts	Category	Additional Info			
☐	BikeImage			image availab	image availab	0			✓		
☐	ImageText					0			✓		
☐	NameInput					0			✓		
☐	123 PictureNumber					0			✓		

3. Go to Services, press ADD → Local JavaScript
   a. Name = AddImage
   b. Under the Inputs tab, click +Add
      i. Name = newImage → BaseType = Image
      ii. Name = newUserName → BaseType = String



General Information | Properties and Alerts | Services

Services | New Service

New Service Local (JavaScript) | Save and Continue

Service Info

Name (required)

addImage

Description

Category

New Service Local (JavaScript) | Save

Service Info

Inputs

+ Add

newImage

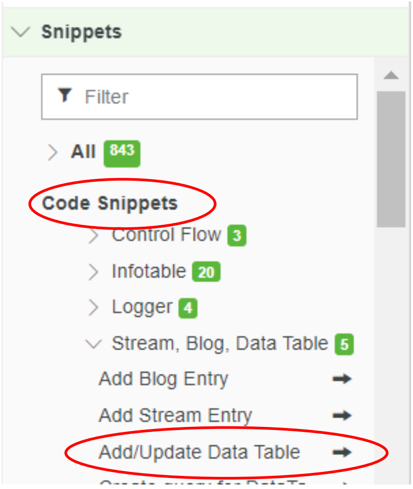
newUserName

Output

Snippets

c. Under Snippets tab, expand Stream, Blog, Data Table

   i. Click the arrow “→” next to “Add/Update Data Table”

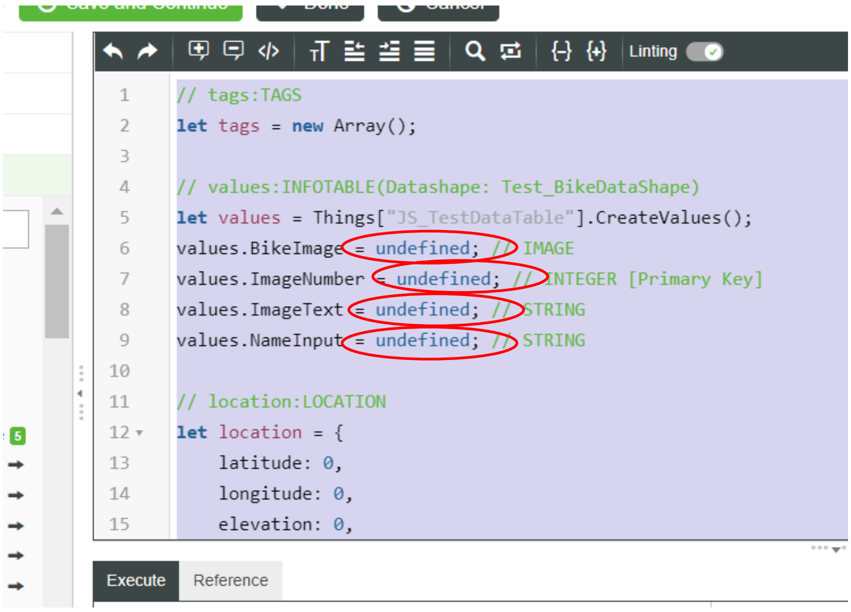


   ii. Pick the DataTable you just made. This will populate your text area.

d. On lines 4-9 it should give you code similar to this:

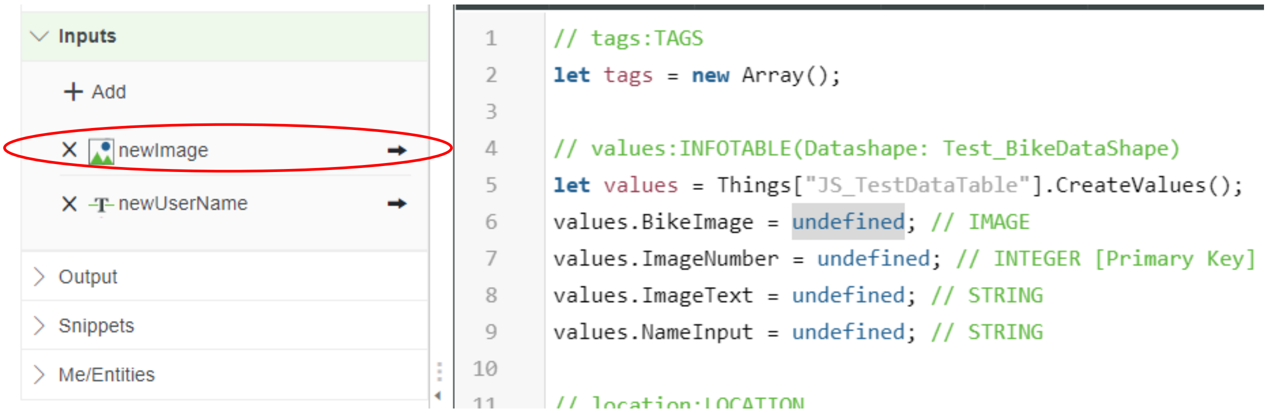
```
// values:INFOTABLE(Datashape: Test_BikeDataShape)
let values = Things["JS_TestDataTable"].CreateValues();
values.BikeImage = undefined; // IMAGE
values.ImageNumber = undefined; // INTEGER [Primary Key]
values.ImageText = undefined; // STRING
```

values.NameInput = undefined; // STRING



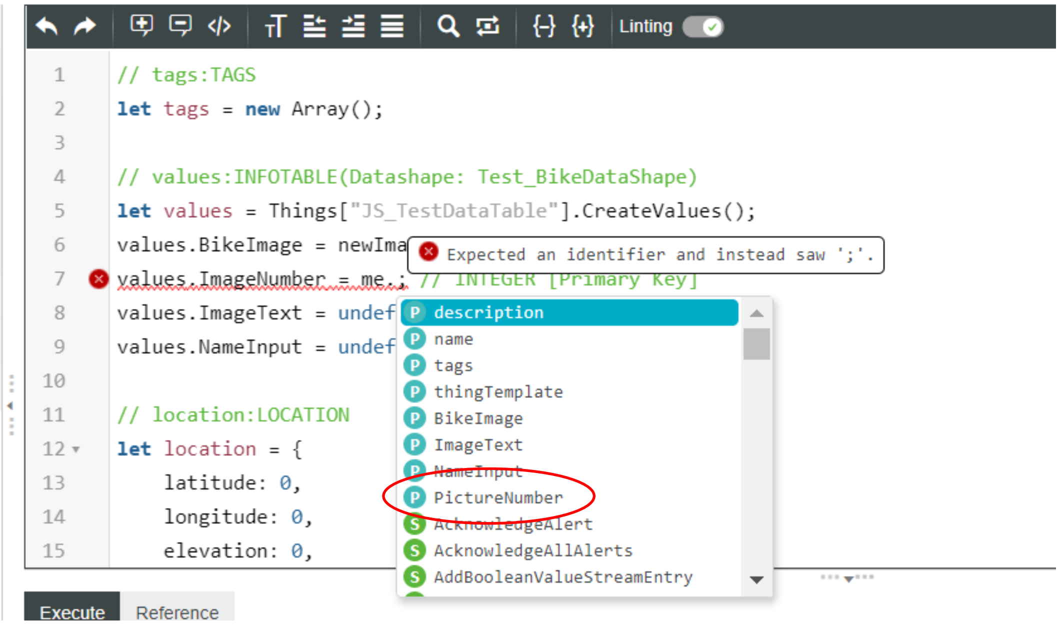
```
1   // tags:TAGS
2   let tags = new Array();
3
4   // values:INFOTABLE(Datashape: Test_BikeDataShape)
5   let values = Things["JS_TestDataTable"].CreateValues();
6   values.BikeImage = undefined; // IMAGE
7   values.ImageNumber = undefined; // INTEGER [Primary Key]
8   values.ImageText = undefined; // STRING
9   values.NameInput = undefined; // STRING
10
11  // location:LOCATION
12  let location = {
13      latitude: 0,
14      longitude: 0,
15      elevation: 0,
```

e. Repopulate all the “undefined” values:
   i. Double click on undefined on line 6. While “undefined” is highlighted, go over to the inputs on the left side of the screen and push the “→” next to newImage. This should repopulate “undefined” with “newImage”



```
1   // tags:TAGS
2   let tags = new Array();
3
4   // values:INFOTABLE(Datashape: Test_BikeDataShape)
5   let values = Things["JS_TestDataTable"].CreateValues();
6   values.BikeImage = undefined; // IMAGE
7   values.ImageNumber = undefined; // INTEGER [Primary Key]
8   values.ImageText = undefined; // STRING
9   values.NameInput = undefined; // STRING
10
11  // location:LOCATION
```

ii. Double click on undefined on line 7. Type “me.” This should give you options of properties to pick. Choose “PictureNumber”. End the line with a “;”



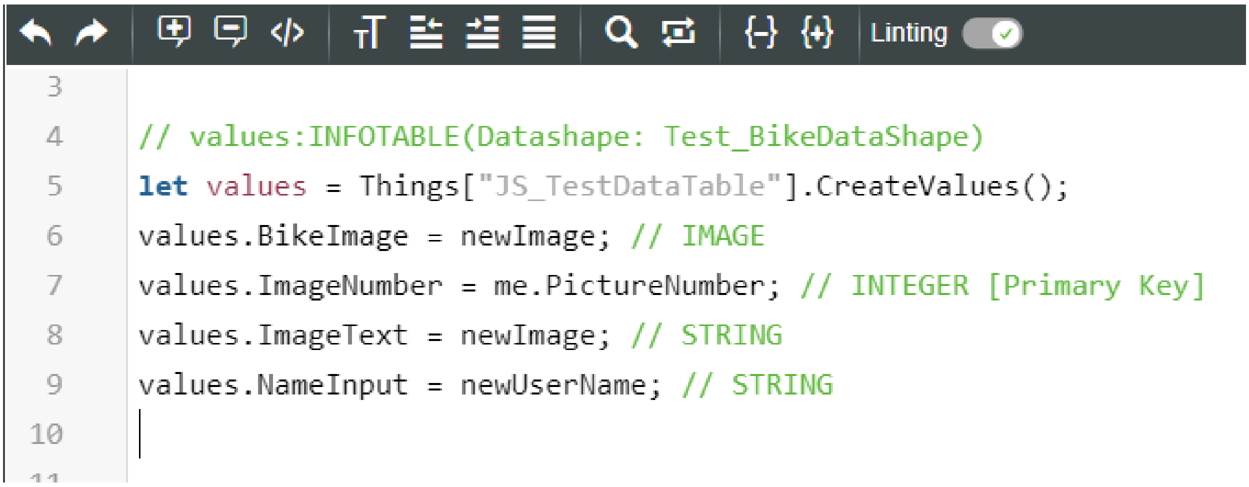
```
1   // tags:TAGS
2   let tags = new Array();
3
4   // values:INFOTABLE(Datashape: Test_BikeDataShape)
5   let values = Things["JS_TestDataTable"].CreateValues();
6   values.BikeImage = newIma
7   values.ImageNumber = me.; // INTEGER [Primary Key]
8   values.ImageText = undef
9   values.NameInput = undef
10
11  // location:LOCATION
12  let location = {
13      latitude: 0,
14      longitude: 0,
15      elevation: 0,
```

Expected an identifier and instead saw ';'.

iii. Double click on undefined on line 8. Repeat Line 6 and repopulate “undefined” with input “newImage”

iv. Double click on undefined on line 9. Using the newUserName input you made, replace “undefined on line 9. This should repopulate “undefined” with “newUserName”

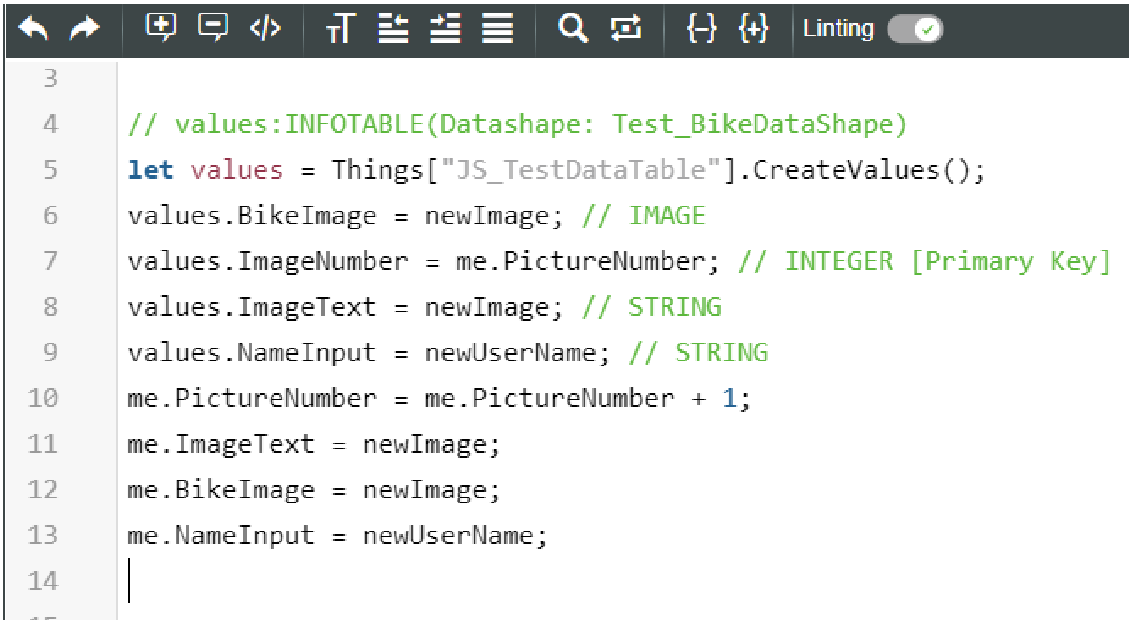
v. Your code should now match this:
// values:INFOTABLE(Datashape: Test_BikeDataShape)
let values = Things["JS_TestDataTable"].CreateValues();
values.BikeImage = newImage; // IMAGE
values.ImageNumber = me.PictureNumber; // INTEGER [Primary Key]
values.ImageText = newImage; // STRING
values.NameInput = newUserName; // STRING



```
3
4   // values:INFOTABLE(Datashape: Test_BikeDataShape)
5   let values = Things["JS_TestDataTable"].CreateValues();
6   values.BikeImage = newImage; // IMAGE
7   values.ImageNumber = me.PictureNumber; // INTEGER [Primary Key]
8   values.ImageText = newImage; // STRING
9   values.NameInput = newUserName; // STRING
10
```

f. Copy and paste this code after line 9

me.PictureNumber = me.PictureNumber + 1;
me.ImageText = newImage;
me.BikeImage = newImage;
me.NameInput = newUserName;

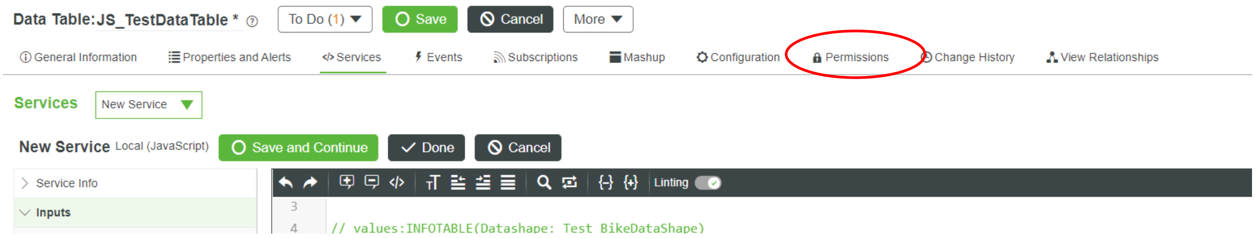


```
3
4   // values:INFOTABLE(Datashape: Test_BikeDataShape)
5   let values = Things["JS_TestDataTable"].CreateValues();
6   values.BikeImage = newImage; // IMAGE
7   values.ImageNumber = me.PictureNumber; // INTEGER [Primary Key]
8   values.ImageText = newImage; // STRING
9   values.NameInput = newUserName; // STRING
10  me.PictureNumber = me.PictureNumber + 1;
11  me.ImageText = newImage;
12  me.BikeImage = newImage;
13  me.NameInput = newUserName;
14
```

g. Click “Save and Continue” and then “Done”

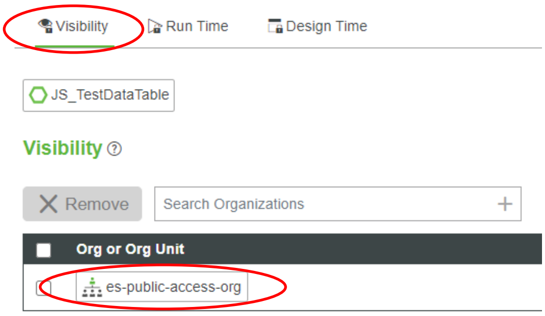
4. Change Permissions in Data Table

a. Go to Permission tab



```
// values:INFOTABLE(Datashape: Test BikeDataShape)
```

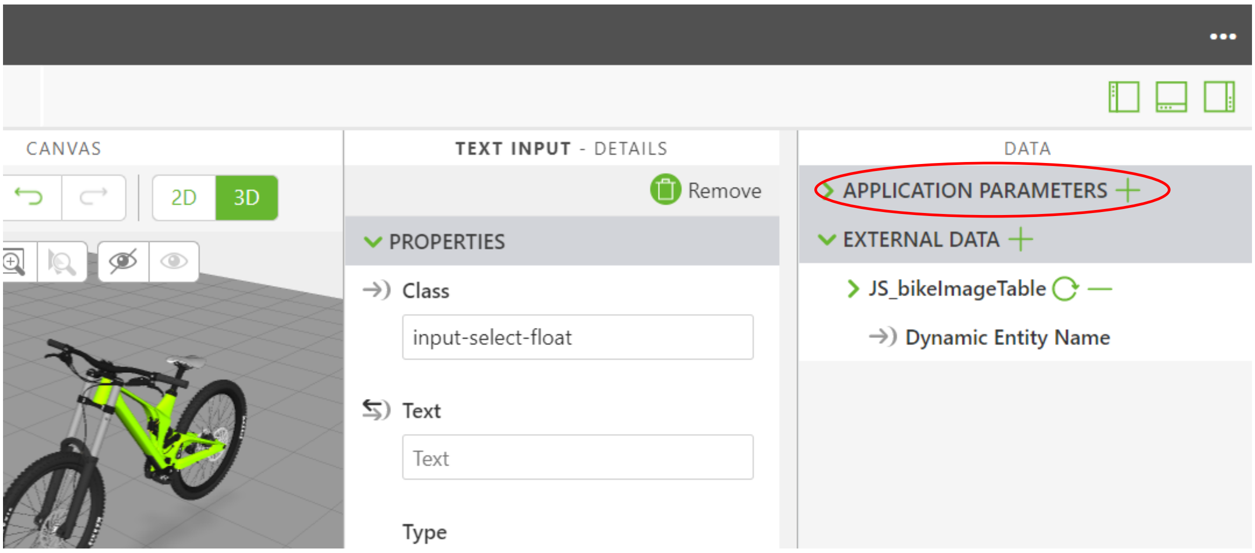
b. Under Visibility, search and select “es-public-access-org”



c. Under Run Time, search and select “es-public-access-org” and then check each box

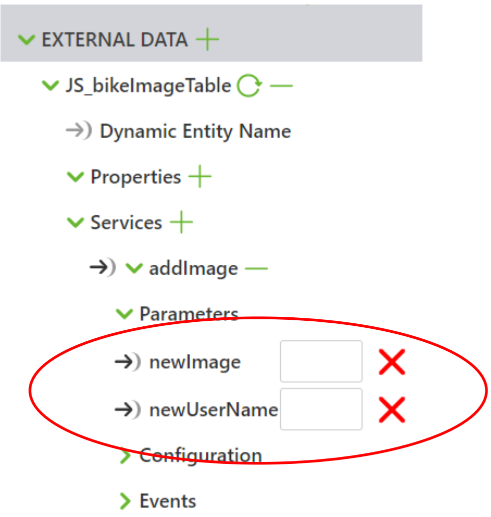
5. Open Vuforia Studio → Bike Example

a. On the far right column under external data, click on the + on “external data +”

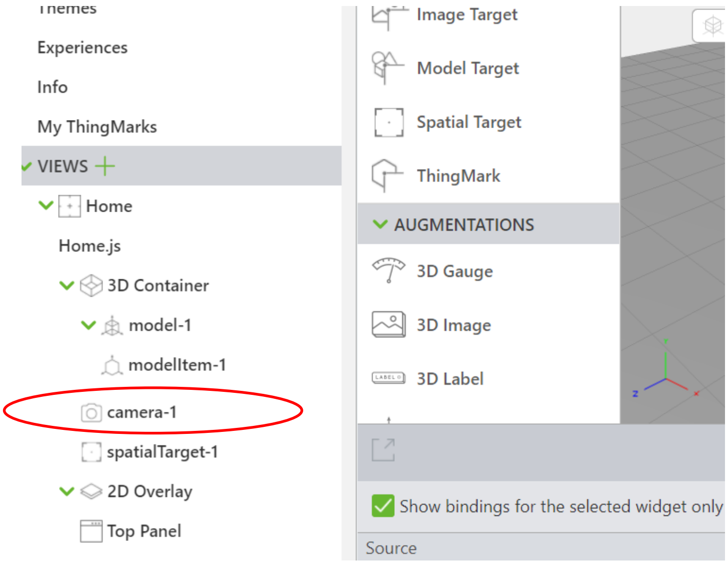


   i. On the left tab of the popup, type in your initials and select your DataTable
   ii. On the right tab of the popup, click the middle tab "Services"
   iii. Search + for "addImage"

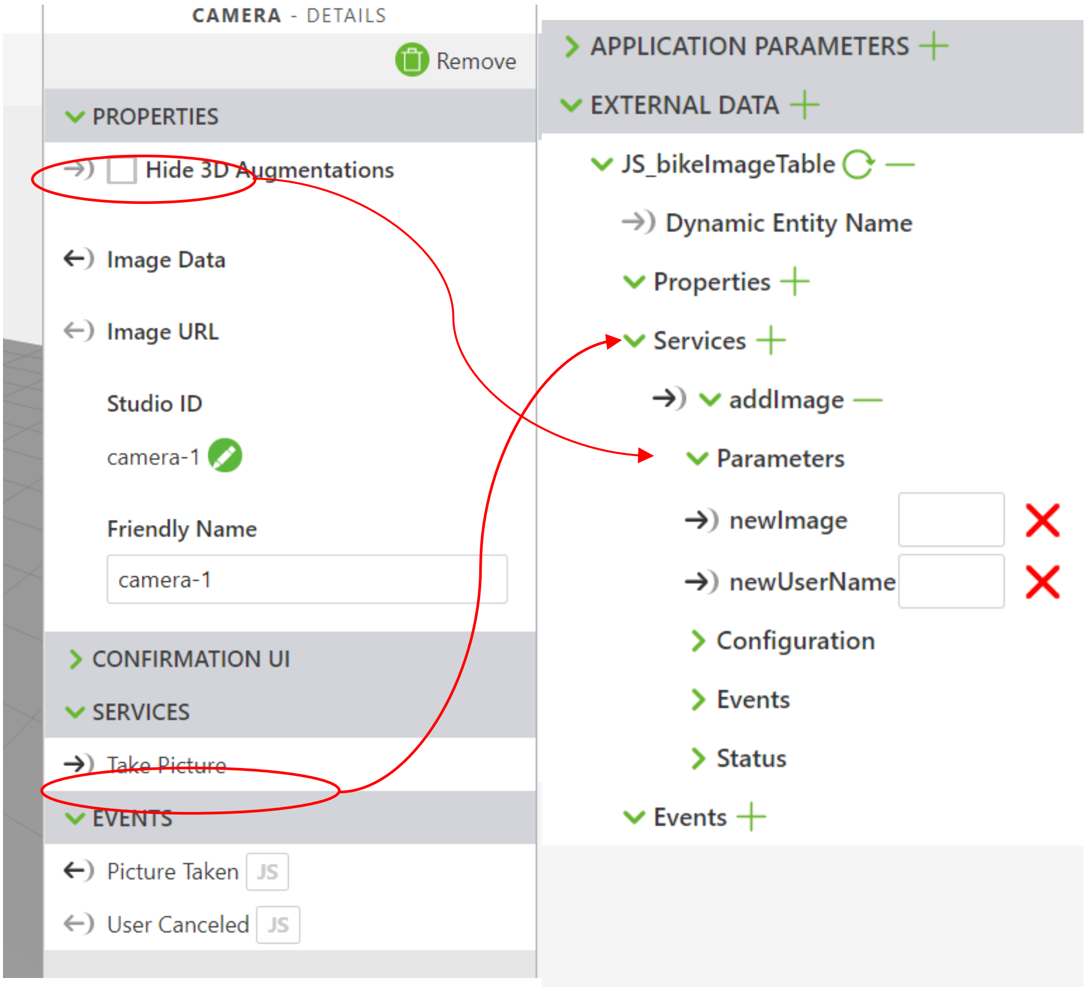
b. Under External Data, extent your Data Table → Services → addImage → Parameters, until you see NewImage and NewUserName



6. Bind Data to Thingworx:
   a. On the far left column find and select camera-1

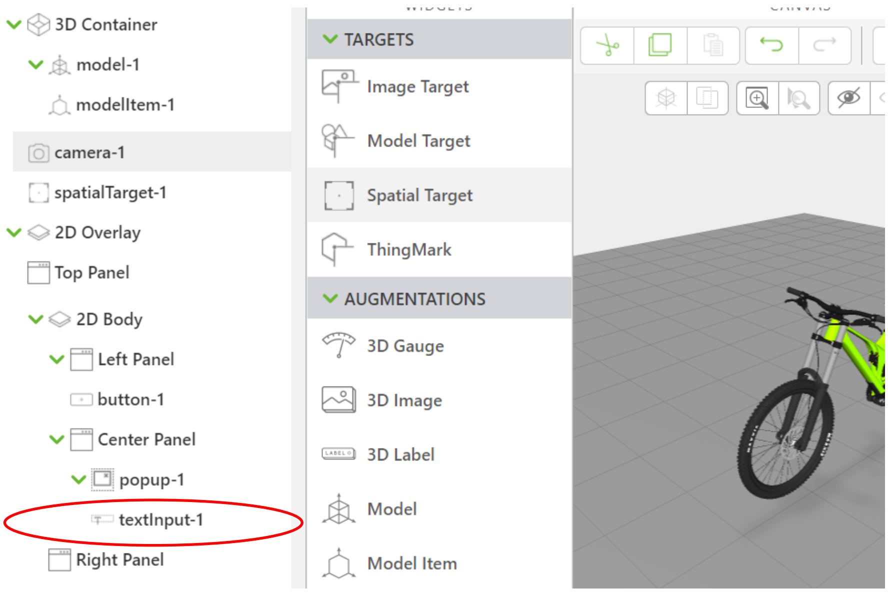


b. On the middle right panel, find “Picture Take” under Events, click and drop the “←” Picture Taken to “addImage” on the far right column

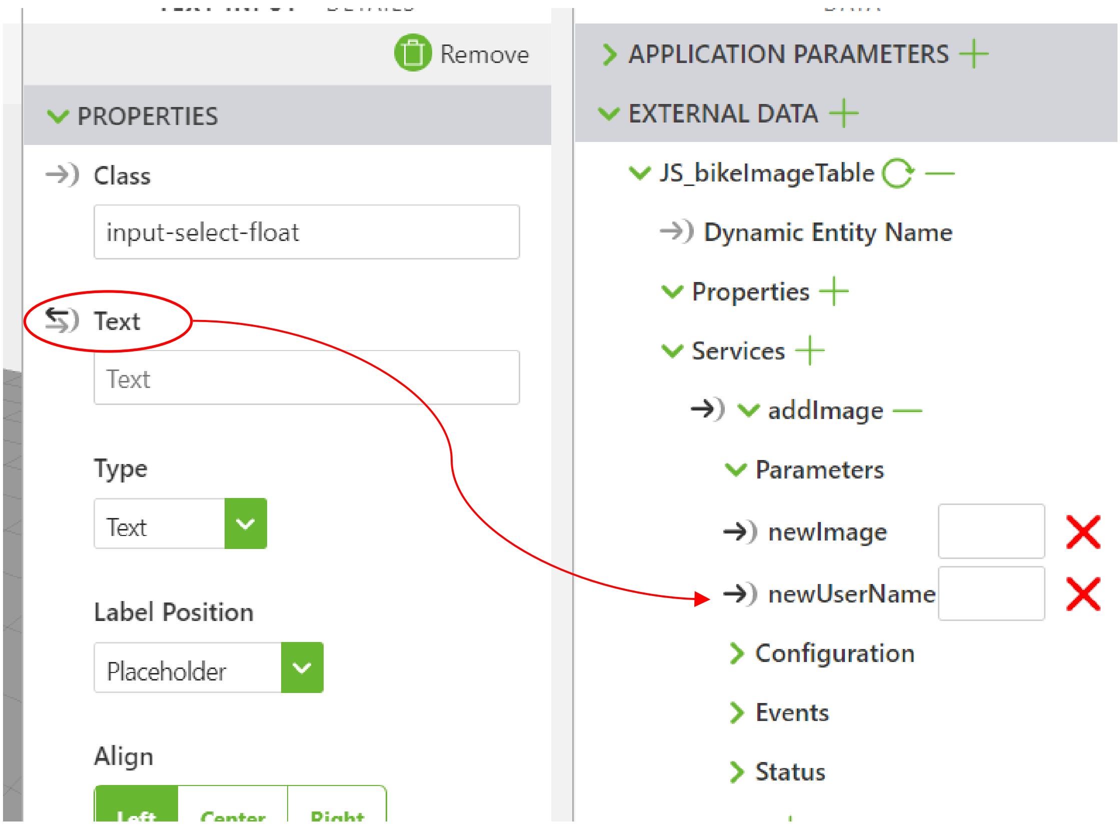


c. On the same middle-right panel, drag and drop the “←” from Image Data to “New Image” on the far right column

d. On the far left column, find and select "textInput-1"



e. Click and drag the arrow from "text" to "newUserName" on the far right column



Class: input-select-float

Text: Text

Type: Text

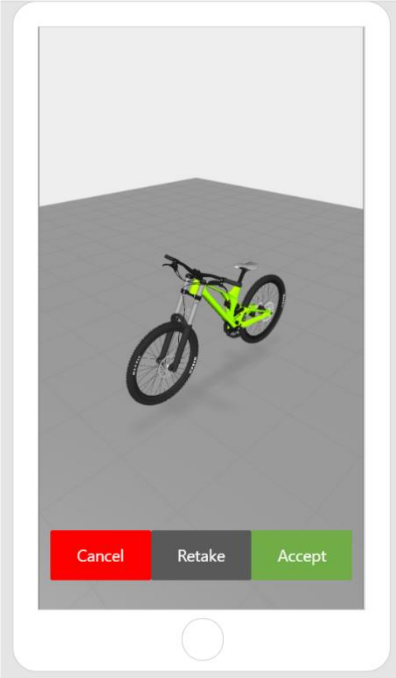
Label Position: Placeholder

Align: Left Center Right

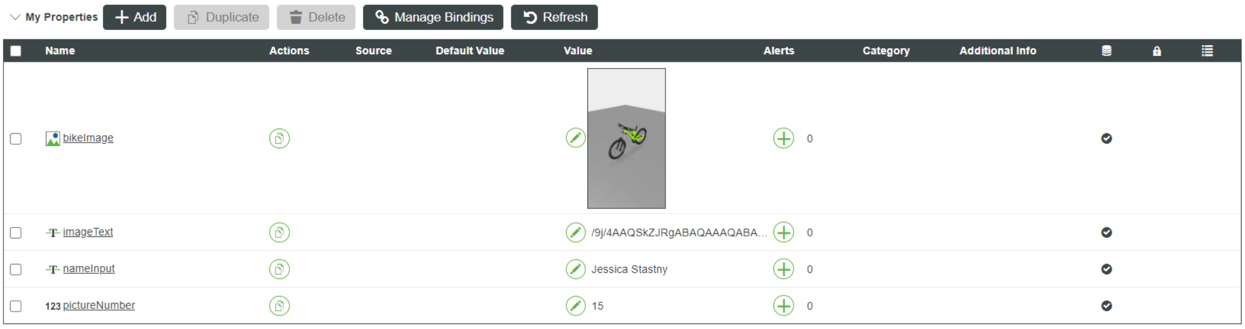
f. Click Save

7. Click Preview
   a. Type you Name and then press enter.
   b. Select the camera button.

c. Press Accept



8. Go back into Thingworx to your Data Table and press refresh. In the Properties tab you should see the image you just took in Vuforia along with your name



☐	Name	Actions	Source	Default Value	Value	Alerts	Category	Additional Info			
☐	bikeImage					0			✓		
☐	imageText				/9j/4AAQSkZJRgABAQAAAQABA...	0			✓		
☐	nameInput				Jessica Stastny	0			✓		
☐	123 pictureNumber				15	0			✓		