How to Create Mashup using Data from a Remote Sensor

Learn how to create a Mashup that displays a temperature and humidity value which is reported to ThingWorx using a Raspberry Pi microprocessor and an AM2302 temperature/humidity sensor
In order to begin building you’re Mashup, first complete the tutorial “How to Bind Data”

1. Once you have completed the “How to Bind Data” tutorial, you will be able to utilize your input data in a Mashup. The first step to creating this Mashup is by locating the search field in the top left corner of your screen:

   ![Search Field](image1)

   **In the search, type “+Mashup” and click the option that matches. This will bring you to a screen that looks like this:**

   ![Mashup Type Selection](image2)

   **New Mashup**

   **Mashup Type**

   ![Mashup Types](image3)

   - Page
   - Thing Template
   - Thing Shape

   **Layout Options**

   ![Layout Options](image4)

   - Responsive
   - Static

   2. Select “Page” for the Mashup type, and “Responsive” for the Layout Options, then click “Done”. You will now be brought to another page that looks like this:
3. Next we’re going to save the project and give it a name so we can retain our work throughout this tutorial. To do that, locate and click the button towards the top-middle part of the page; doing so will bring you here:

![General Information](image)

Make sure to add a name that follows the naming conventions outlined in the previous tutorial (if you don’t remember click here). I named my Mashup “Am2302Mashup” so I could easily locate it at any time. Also, it is **very** important that you tag this Mashup with the same tag you used for the ThingShape, ThingTemplate, and Thing in the “How to Bind Data” tutorial; if you don’t do this you will be unable to transfer your project to any other servers.

To complete this step, click , and then click to get back to the screen in step 2.

4. Once you’ve reached this page, go to the right part of the screen where there are three tabs labeled “Data”, “Session” and “User”. Under the “Data” tab, click , doing so will bring you to this screen:

*click here to go back to step 11
5. Pictured above, there is a checkbox called “Dynamic” that is circled, click that box to check it. The reason you want to check this box is because you want to add an entity with data that changes or updates, hence dynamic.

6. Next, select the dropdown [All] to get the following menu:

7. Select “Thing Templates” and then click the wand:

8. Selecting the wand will allow you to search for the Template we made in the last tutorial “How to bind data”. Find the template we made (I named mine Am2302Template) and select it

*Click [here] to go back to step 12
9. Selecting the template will give you a list of services that you can implement. Services are basically actions that the entity will perform. For this tutorial, we want to perform an action that retrieves the numerical values for temperature and humidity. To do this, we add a service called “GetProperties”. In order to add “GetProperties” as a service, type “GetProperties” into the “Select Services” field and click the : 

10. For now, ignore the box that says “Mashup Loaded” we’ll get to that later. When this is completed, go ahead and click also make sure to your work!

11. Next, we’re going to add a new entity with a service called “GetImplementingThings”. This entity will allow us to begin implementing data from the sensor in our Mashup. Begin by repeating step 4, but this time don’t check the “Dynamic” box

12. Repeat steps 6, 7 and 8

13. In the “Select Services” field, type “GetImplementingThings” and then click

14. This time, we are going to check the box that says “Mashup Loaded?” . What this means is that we are telling the Mashup “perform this service when you start (load) up” . If we did not check this box, we would have to identify another trigger (i.e. clicking):

15. When you have this completed, go ahead and click , you may also want to click !
16. Verify that the information under your “Data” tab looks like the image below to the left:

NOTE: I created my entity with the “GetImplementingThings” service before I created my entity with the “GetProperties” service (the reverse of what you did). Disregard this since order does not matter here.

17. Click “GetImplementingThings” (the red bar) as I have done here. The black dashed line around the bar indicates it is selected.

18. Click and hold so that you can drag and drop it on top of “GetProperties”:

19. Notice how the bottom pane changes:

This pane provides users (us) with a visual representation of what we’re telling ThingWorx to do, it’s a good idea to check this every once in a while throughout the project. In this pane for instance we can see that ThingWorx is saying we have taken information from the Am2302Template (which houses a Am2302Thing bound to data from our sensor) and called a service which implements a new service (GetProperties) that is triggered by a row value changing. From there it says “get the new data from the service that calls upon the properties (data) of the Am2302Thing (the temperature and humidity values).

20. Now we can begin the fun part – using widgets! To begin, locate the “Widgets” tab on the
left part of your screen:

21. Where it says “Filter Widgets” type “LED” to get a widget that simulates an LED display (like a digital alarm clock):

22. Click, hold and drag the “LED Display” onto the white space located to the right of the widgets pane:

23. When you do this, you should get a message from ThingWorx:

![Add a Panel](image)

Click “Yes”
24. Go back to the “Data” tab on the right part of the screen and select “GetProperties” and then click, hold and drag “Prop_Humidity” onto the LED Display:

25. This next display will be:

Click “#Data”, doing this binds selected data (the humidity value) to a specific action that the widget performs; in this example, the LED performs the action of displaying numerical data.

26. Repeat steps 20 through 25, but this time for Prop_Temperature. When you’re done click Save and then click View Mashup to see your results!
Reference

Naming conventions:
1) Names must begin with a letter, the dollar sign "$", or the underscore character "_"
2) No white space in the name i.e. “Temperature Sensor” is not allowed
3) After the first character, letters, digits, dollar signs and underscore characters are allowed
4) Good naming practices include:
   o making one word names all lowercase letters
   o for multi-word names, type the first word in lowercase and capitalize the first letter of each subsequent word
   o “temperatureSensorThingShape” is an example of an appropriate name

*Click [here](#) to go back to step 3