

57. Build the logic of the save action for the Add Edit callout. We will need to grab all the data from the fields, determine if it's an edit or a new and then create the alarm action.

1. Open set.dg5

1. Right click on main – headers – newAlarm – newAlarm – callout, select Edit Symbol

1. Fill in all the fields of the callout

1. It is much easier to follow form submission dataflows when there is data passing through.
2. New Alert: "Mem1"
3. Severity: High
4. Metric: Memory Usage
5. Is: >
6. Value: "80"
7. Email: "me@me.com"

2. Right click on addEditCallout – buttons – save, select Dataflow

1. Click on data panel

1. Click on downstream – Alarm Service – metrics

1. Drag and Drop Add Algorithm to dataflow

1. Type: Out of Range
2. Name: Bind from addEditCallout - row1 - name - Symbol Properties - output
3. Right click on Complete, select Pinned
4. Invoke: Bind from addAlarm - Symbol Parameters - invoke
5. Click on invoke, to create our first test Alarm

2. Click on data panel

1. Click on downstream – Alarm Service – metrics – Mem1

1. Drag and Drop Add Watch to dataflow

1. Name: Bind from addEditCallout - row1 - name - Symbol Properties - output
2. Path: Bind from addEditCallout - row1 - metric - Symbol Properties - output
 1. Note: There are 2 path fields in this block, path, the uppermost, and Path, the lowermost. The above directions applies to the lowermost, Path, which is where the alarm will be watching the values to see if they fall out of range of the prescribed value
3. path: (Uppermost one) Bind from addAlarm - Symbol Parameters - sourcePath
4. Invoke: Bind from Add_Algorithm - onComplete

3. Select Add_Algorithm and Add_Watch, right click, select Convert to Symbol

1. Name: "addAlarm"

2. Click Ok
3. When you convert existing bound blocks, any incoming binding is converted to a symbol Parameter.
4. Click on addAlarm
 1. Name1: Right click, select Delete
5. Right click on addAlarm, select Edit Symbol
 1. Right click on addAlarm dataflow, select Edit Properties
 1. Symbol Parameters
 1. Right click Name1, select Delete
 2. Double click on Path
 1. Rename: "watchPath"
 3. Name: Bind to addAlarm dataflow - Add_Algorithim - Name
 4. Name: Bind to addAlarm dataflow - Add_Algorithim - Name
 5. watchPath: Bind to addAlarm dataflow - Add_Algorithim - Path
6. Save symbol
3. On addEditCallout - buttons - save dataflow
 1. Note that the parameter Name1 is still listed as a parameter. This is a dataflow bug. If you delete a data symbol parameter while it is bound, the name of the parameter will be stuck to the block and you will be unable to delete it. Therefore, it is easier to just delete the block and create a new one, or make sure to unbind prior to the parameter deletion.
 2. Right click on addAlarm, select Delete
 3. Click on Symbols panel
 1. Drag and Drop addAlarm
 1. Name: Bind from addEditCallout - row1 - name - Symbol Properties - output
 2. watchPath: Bind from addEditCallout - row1 - metric - Symbol
 4. Drag and Drop Logic - Concatenate
 1. Input 0: "/downstream/alarming/Alarm Service/metrics"
 2. Input 1: "/"
 3. Click + button
 1. Input 2: : Bind from addEditCallout - row1 - name - Symbol Properties - output
 5. Right click on addAlarm, select Edit Symbol
 1. Right click on dataflow, select Edit Properties
 1. Symbol Parameters
 1. Drag and Drop string
 1. Name: "sourcePath"
 6. Save symbol
 7. Click on addAlarm
 1. Right Click on sourcePath, select Pinned
 2. sourcePath: Bind from concat - output
 8. Right click on addAlarm, select Edit Symbol
 1. Right click on dataflow, select Edit Properties
 1. Symbol Parameters
 1. Drag and Drop string
 1. Name: "operation"
 2. Right click, select Pinned

2. Drag and Drop string
 1. Name: "severity"
 2. Right click, select Pinned
 3. Drag and Drop string
 1. Name: "metric"
 2. Right click, select Pinned
 4. Drag and Drop string
 1. Name: "email"
 2. Right click, select Pinned
 5. Drag and Drop number
 1. Name: "value"
 2. Right click, select Pinned
 6. Drag and Drop trigger
 1. Name: "invoke"
 2. Right click, select Pinned
 7. Drag and Drop trigger
 1. Name: "close"
 2. Right click, select Pinned
 8. Severity: Bind from addEditCallout - row1 - severity - Symbol Properties - output
 9. Metric: Bind from addEditCallout - row2 - metric- Symbol Properties - output
 10. Operation: Bind from addEditCallout - row2 - operation - Symbol Properties - output
 11. Value: Bind from addEditCallout - row1 - value - Symbol Properties - output
 12. Email: Bind from addEditCallout - row1 - email - Symbol Properties - output
 13. Invoke: bind from addEditCallout - Actions - Advanced, onClick
2. On dataflow
 1. Drag and Drop Logic - If block
 1. Input 0: Bind from addAlarm - Symbol Parameters - operation
 2. Input 1: ">"
 3. Then: Bind from addAlarm - Symbol Parameters - value
 4. Else: "-99999"
 5. Duplicate block with (Cntrl + D)
 2. If1
 1. Input 1 : "<"
 2. Else: "99999"
 3. Drag and Drop String - Concatenate block
 1. Input 0: Bind from addAlarm - Symbol Parameters - sourcePath
 2. Input 1: "/Max Value"
 3. Duplicate block with (Cntrl + D)
 4. concat1
 1. Input 1: "/Min Value"
 5. Click on Data tab

1. Click on downstream – alarming – Alarm Service – metrics – Mem1
2. In Metrics tab
3. Right click on Max Value
4. Drag and Drop @set to addAlarm dataflow
5. Right click on Min Value
6. Drag and Drop @set to addAlarm dataflow
6. set
 1. Name: “setMin”
 2. path: Bind from concat1 - output
 3. value: Bind from if - output
 4. Right click on onComplete, select Pinned
 5. Invoke: Bind from Add_Watch - onComplete
7. set1
 1. Name: “setMax”
 2. path: Bind from concat - output
 3. value: Bind from if1 - output
 4. Right click on onComplete, select Pinned
 5. Invoke: Bind from SetMin - onComplete
8. Click on Data tab
 1. Click on downstream – alarming – Alarm Service – metrics – Mem1
 2. Drag and Drop @addAttribute to dataflow
9. addAttribute
 1. path: Bind from addAlarm – System Parameters – sourcePath
 2. Right click on onComplete, select Pinned
 3. Duplicate block with (Cntrl + D) 3 times
 4. Rename: “severity”
 5. Name: “severity”
 6. Value: Bind from addAlarm – System Parameters – severity
 7. Invoke: Bind from setMin - onComplete
10. addAttribute1: Rename: “email”
 1. Rename: “email”
 2. Name: “email”
 3. Value: Bind from addAlarm – System Parameters – email
 4. Invoke: Bind from severity - onComplete
11. addAttribute2: Rename: “operation”
 1. Rename: “operation”
 2. Name: “operation”
 3. Value: Bind from addAlarm – System Parameters – operation
 4. Invoke: Bind from email - onComplete
12. addAttribute3: Rename: “metric”
 1. Rename: “metric”
 2. Name: “metric”
 3. Invoke: Bind from operation - onComplete

13. Drag and Drop String Operations - Split
 1. Separator: "/"
 2. Input: Bind from addAlarm - System Parameters - watchPath
 14. Drag and Drop Table Operations - Table Aggregation block
 1. Input: Bind from split - output
 2. Column: "value"
 3. Method: Last
 4. Output: Bind to metric - output
 15. Drag and Drop Logic - Delay block
 1. Input 0: Bind from metrics - onComplete
 2. Delay: 0.3
 3. Output 0: Bind to addAlarm - System Parameters - close
9. Save symbol
 4. Click on addEditCallout - buttons - save - dataflow
 1. addAlarm
 1. Close: Double click on blue binding dot to hang binding window
 5. Click on addEditCallout - buttons - cancel - dataflow
 1. Hub
 1. Input 1: Bind from addAlarm.close hanging binding block
 6. Close hanging binding block
 2. Save Symbol
2. Save page
 3. Test the Add Alarm functionality and make sure everything works.

From:

<https://wiki.dglogik.com/> - **DGLogik**

Permanent link:

<https://wiki.dglogik.com/dglearning:level3:57?rev=1556322301>

Last update: **2021/09/20 14:40**

