Using JavaScript to parse input fields for Nested Attributes  
  
First screen shot shows the process (step 7), the second one shows how you use the output of the JavaScript (the array of data) as the Data Pill for the Nested Attribute you are posting into (step 8).   
  
The required JavaScript is here (the sample schemas are below the screen shots)

exports.main = ({ inputNames }) => {

var aNames = inputNames.split(";")

const rows = []

aNames.forEach(thisName => {

row = {}

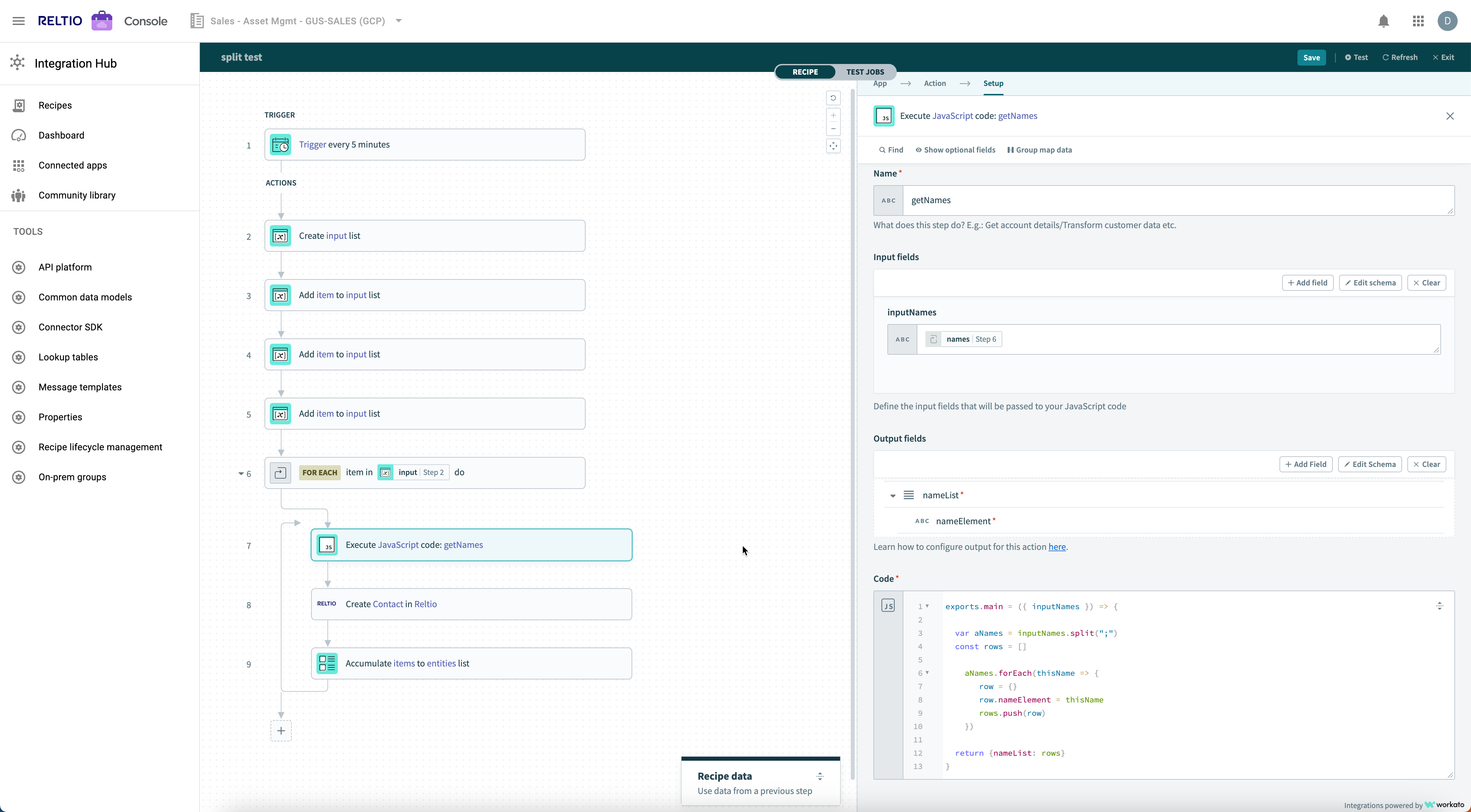
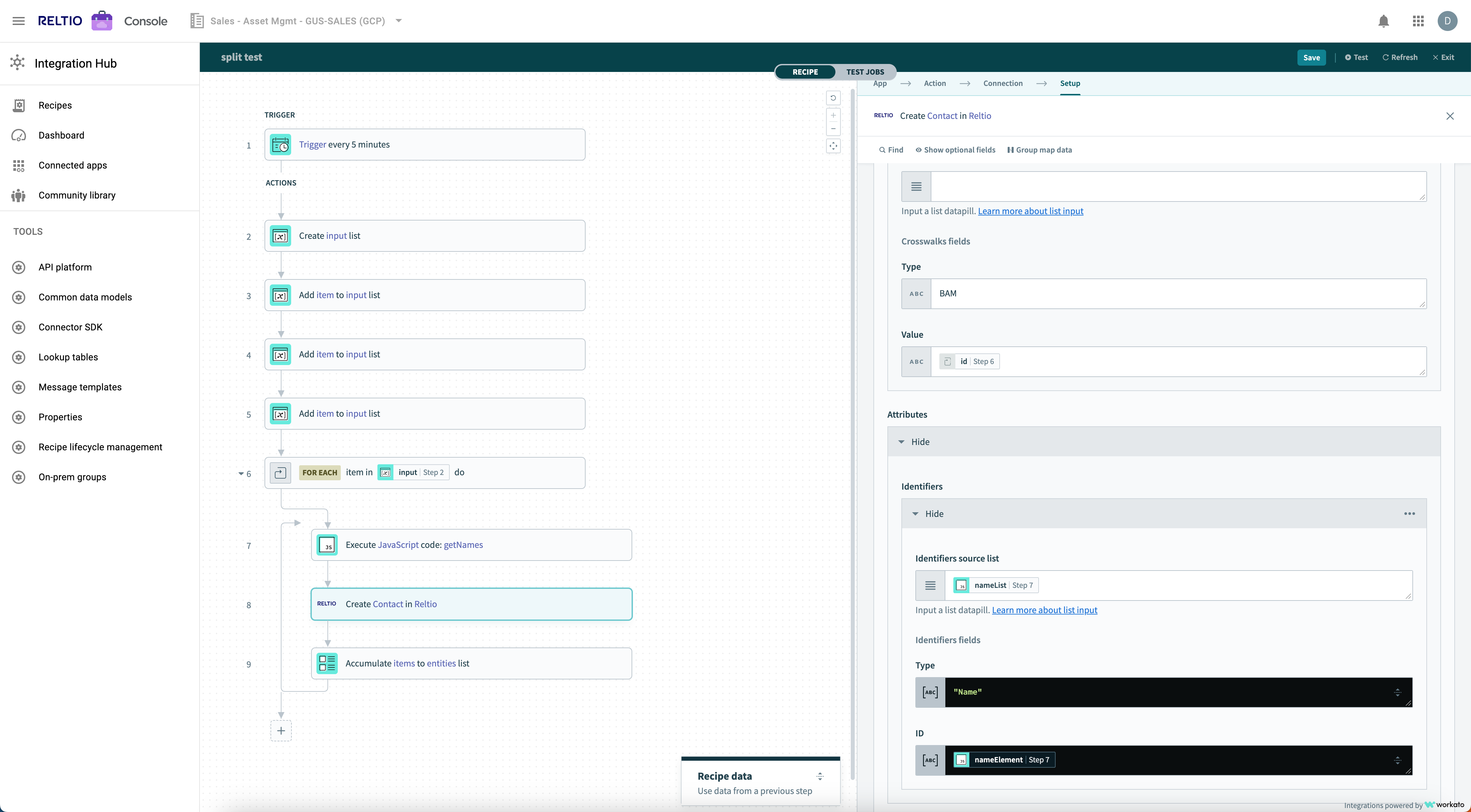
row.nameElement = thisName

rows.push(row)

})

return {nameList: rows}

}

Input Schema:  
[

{

"control\_type": "text",

"label": "inputNames",

"name": "inputNames",

"type": "string",

"optional": true

}

]

Output Schema:  
[

{

"name": "nameList",

"type": "array",

"of": "object",

"label": "nameList",

"properties": [

{

"control\_type": "text",

"label": "nameElement",

"type": "string",

"name": "nameElement"

}

]

}

]