

Screen: s1 +

Element: X Label +

Process name: New process

Version: 1 ^

Origin ID: EG_DEFAULT

Deploy to Portal

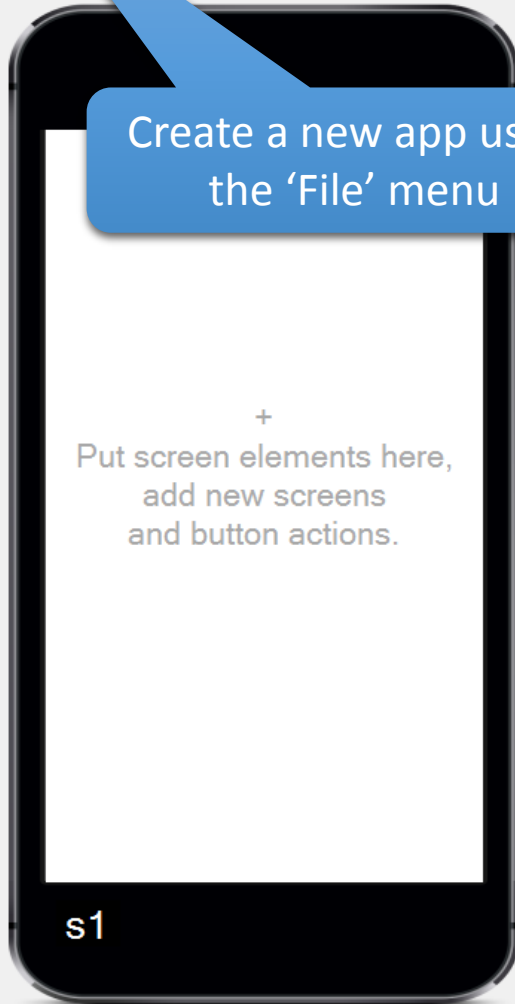
Build App QR Code



Description: New process description

Password:

Create a new app using the 'File' menu



+
Put screen elements here,
add new screens
and button actions.

s1

1. Add two more screens, so that we have s1, s2, and s3

2. Give your app some name and description

3. Add some labels and a button

The screenshot shows the Mobile Modeler application window. The title bar indicates the file path: [C:\Users\boris\AppData\Roaming\EastGate\Mobile Modeler\Mobile Process\REST_JSON_Service.p55m]. The menu bar includes 'File', 'View', and 'Support'. The toolbar contains a 'Screen' dropdown menu (currently showing 's1' with a '+' icon), an 'Element' dropdown menu (currently showing 'Label' with a '+' icon), and several input fields: 'Process name: REST JSON Demo', 'Version: 3', 'Origin ID: EG_DEFAULT', 'Description: RESTful service demo', and a 'Password' field. There are also buttons for 'Deploy to Portal' and 'Build App QR Code', and the 'PROCESS' logo.

The main workspace displays a mobile app design for screen 's1'. It features a list of UI elements:

Label s1_o1	This is a demonstration of a REST/JSON
Label s1_o2	service call returning 2 tables and scalars
Label s1_o3	over HTTP
Button s1_b1	Call the service!

The screen 's1' is also labeled at the bottom left of the mobile device preview.

(this is the actual JSON response we are going to receive)

```
{
  "colorsArray": [
    {
      "colorID": "1",
      "colorName": "red",
      "hexValue": "#f00"
    },
    {
      "colorID": "2",
      "colorName": "green",
      "hexValue": "#0f0"
    },
    {
      "colorID": "3",
      "colorName": "blue",
      "hexValue": "#00f"
    },
    {
      "colorID": "4",
      "colorName": "cyan",
      "hexValue": "#0ff"
    },
    {
      "colorID": "5",
      "colorName": "magenta",
      "hexValue": "#f0f"
    },
    {
      "colorID": "6",
      "colorName": "yellow",
      "hexValue": "#ff0"
    },
    {
      "colorID": "7",
      "colorName": "black",
      "hexValue": "#000"
    }
  ],
  "usersList": [
    {
      "personID": "1001",
      "nameFirst": "Holden",
      "nameLast": "Caulfield",
      "department": "Security"
    },
    {
      "personID": "1002",
      "nameFirst": "Jane",
      "nameLast": "Eyre",
      "department": "PR"
    },
    {
      "personID": "1003",
      "nameFirst": "Randall",
      "nameLast": "Flagg",
      "department": "HR"
    },
    {
      "personID": "1004",
      "nameFirst": "Milo",
      "nameLast": "Minderbinder",
      "department": "Logistics"
    },
    {
      "personID": "1005",
      "nameFirst": "Ichabod",
      "nameLast": "Crane",
      "department": "Logistics"
    },
    {
      "personID": "1006",
      "nameFirst": "Jay",
      "nameLast": "Gatsby",
      "department": "Logistics"
    }
  ],
  "scalars": [
    {
      "recordID": "10001",
      "itemNumber": "4711",
      "itemDescription": "Test item 1",
      "itemCreated": "2016-02-14 12:00:00",
      "itemCount": "125"
    }
  ]
}
```

The screenshot shows the PROCES5 Mobile Modeler interface. On the left, a mobile app design is visible with a screen labeled 's1' containing three labels (s1_01, s1_02, s1_03) and a button (s1_b1) labeled 'Call the service!'. The main workspace shows the configuration for the button's action sequence. The 'Element variable name' is 's1_b1' and the 'Button text' is 'Call the service!'. The 'ACTION SEQUENCES' panel shows a 'Service' action named 'Test' selected, with 'Initialize' checked and 'Transfer' selected as the action type. The 'Service name' is 'Test', 'Service type' is 'JSON_GET', and the 'URL' is 'http://www.proce55.com/test/values.json'. The 'Parameters' table is as follows:

Type	Name	Value
Export	colorsArray	s2_t1
Export	usersList	s2_t2
Export	itemNumber	s2_o4
Export	itemDescription	s2_o6

Other configuration options include 'Service name: Test', 'Service type: JSON_GET', 'Encoding: UTF-8', 'URL: http://www.proce55.com/test/values.json', and 'Ignore certificate errors (HTTPS/TLS)'. A 'Save' button is visible.

1. Select the button...

2. Set the action sequence target to the screen s2

3. Add a 'Service' action and set the service parameter. Then press the blinking Save/Apply buttons

1. Switch to the screen s2 and add six labels, two tables and a button

ProcessMaker [C:\Users\doris\AppData\Roaming\Eastgate\Mobile Modeler\Mobile Process\REST_JSON_Service.p55m]

Screen: **s2** Element: **Label**

Process name: REST JSON Demo Version: 3 Origin ID: EG_DEFAULT
Description: RESTful service demo Password: Deploy to Portal Build App QR Code

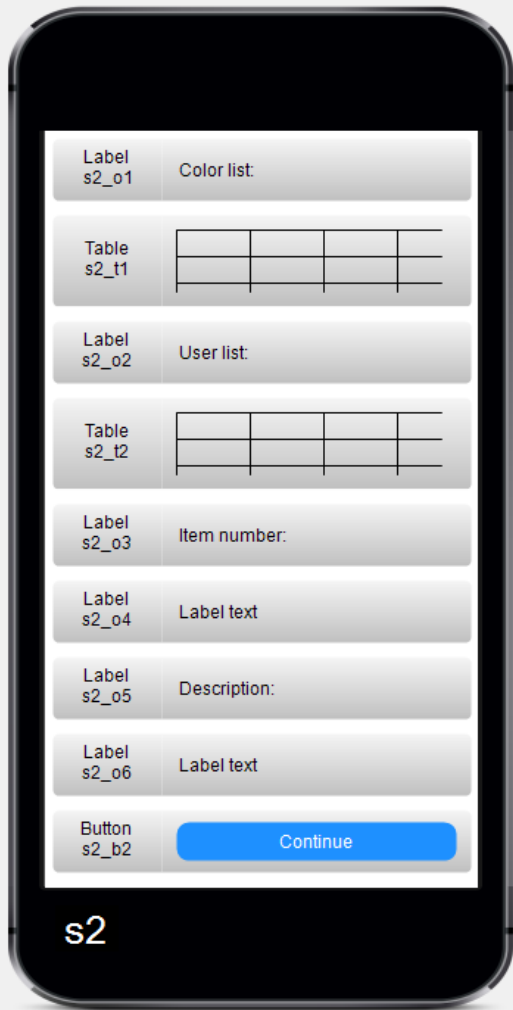
Element variable name: **s2_t1**

Hidden

Header text: ID System name: colorID Hidden column

ID	Name	HEX Value
11	Test	FFF

2. Add three table columns having exact system names (colorID, colorName and hexValue) as seen on the following slides...



Element variable name: **s2_t1**

Hidden

< Apply

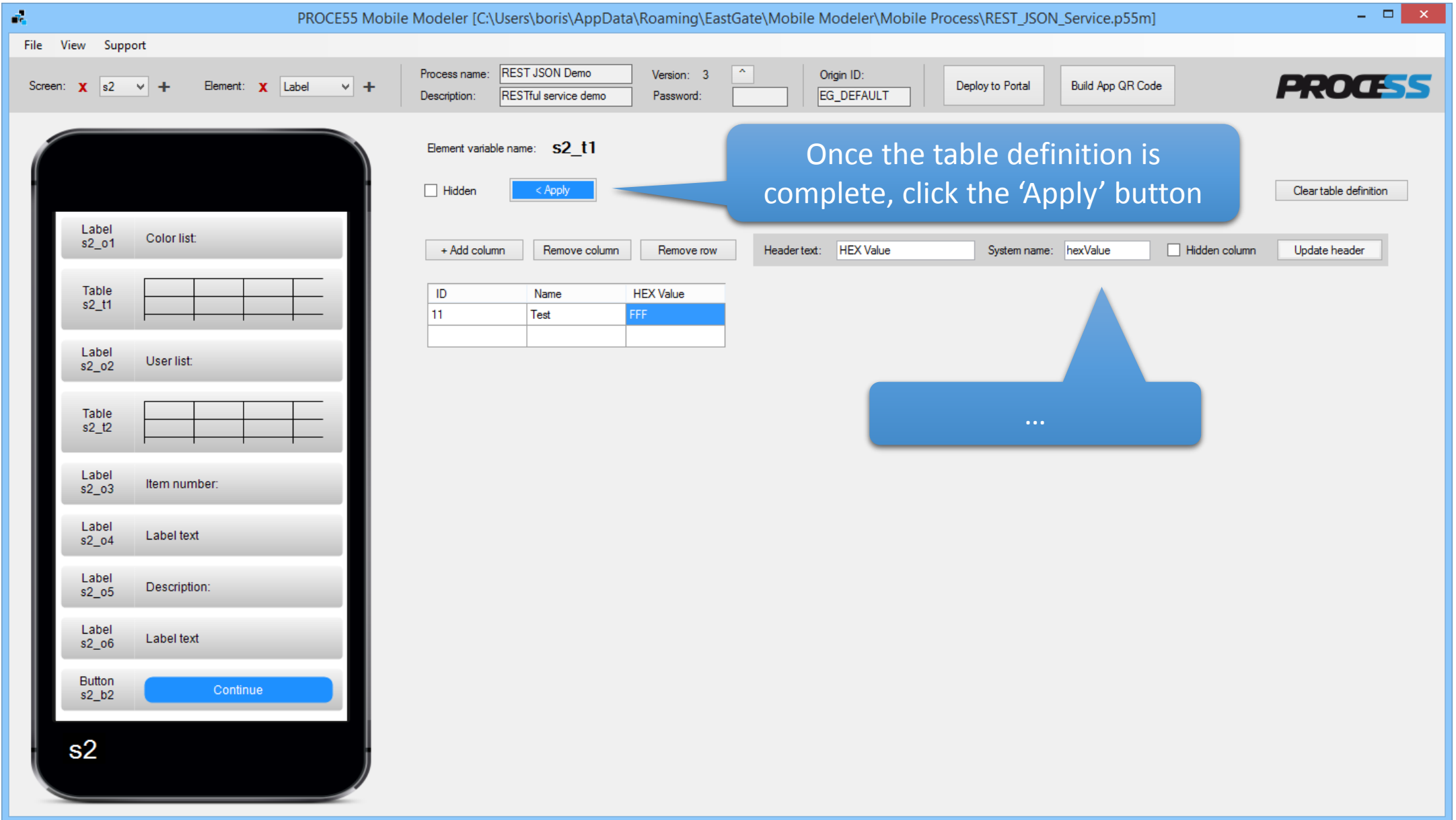
Clear table definition

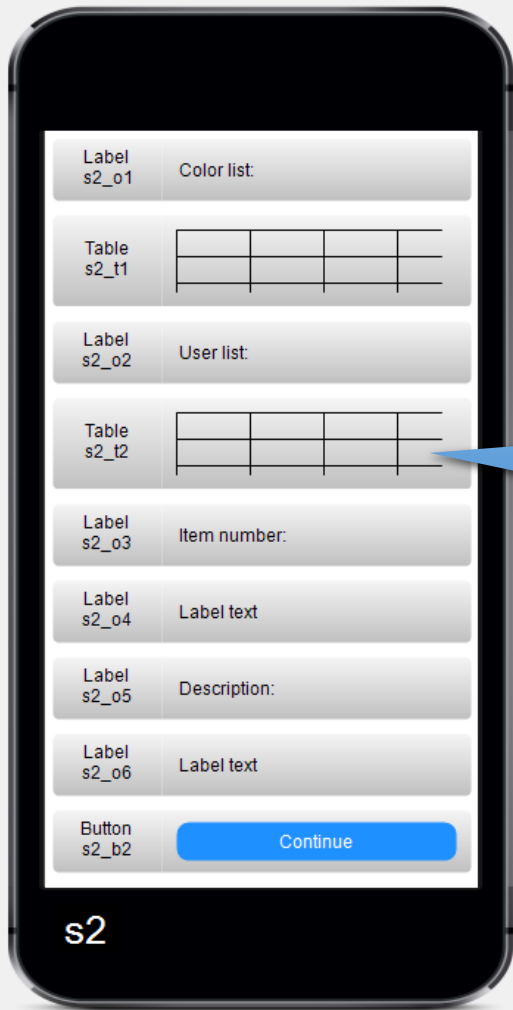
+ Add column Remove column Remove row

Header text: Name System name: colorName Hidden column Update header

ID	Name	HEX Value
11	Test	FFF







Element variable name: **s2_t2**

Hidden

< Apply

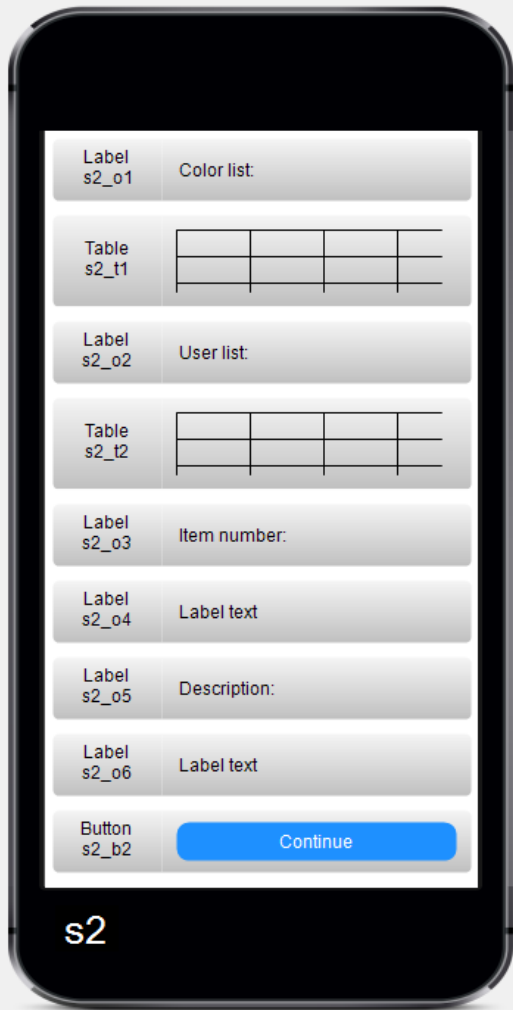
Clear table definition

+ Add column Remove column Remove row Header text: Personal ID System name: personID Hidden column Update header

Personal ID	First name	Last name	Dept.
11	22	3	4



Define the other table similarly, adding four table columns: personID,nameFirst, nameLast and department



Element variable name: **s2_t2**

Hidden

< Apply

Clear table definition

+ Add column

Remove column

Remove row

Header text: First name

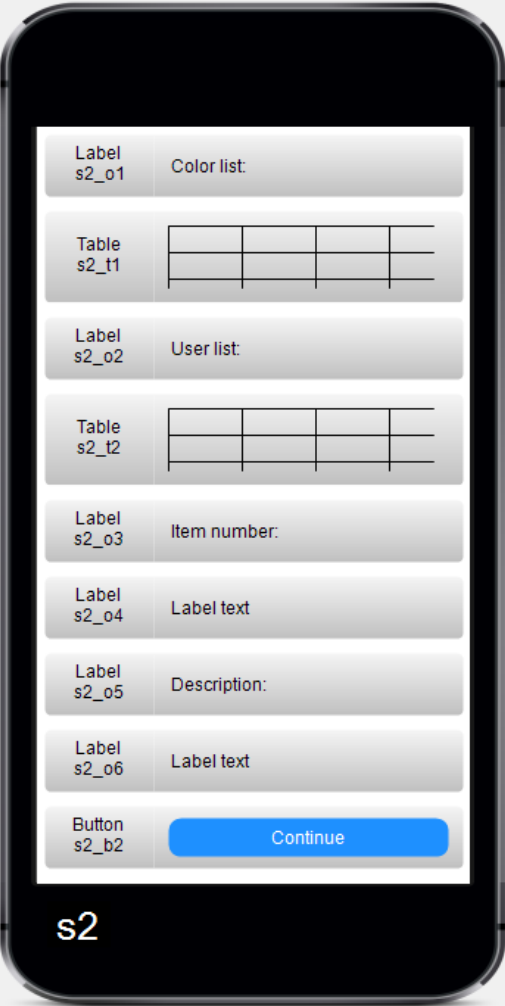
System name: nameFirst

Hidden column

Update header

Personal ID	First name	Last name	Dept.
11	22	3	4



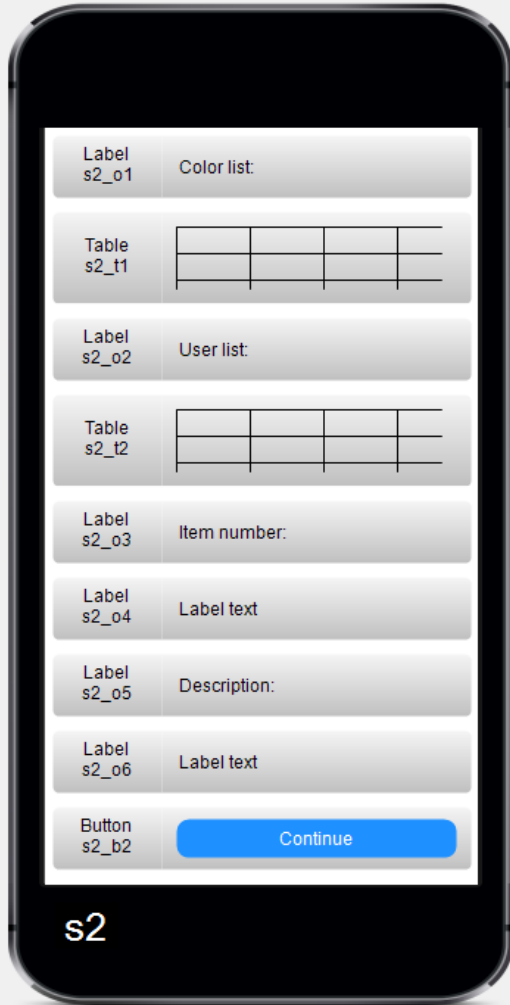


Element variable name: **s2_t2**
 Hidden < Apply Clear table definition

+ Add column Remove column Remove row Header text: Last name System name: nameLast Hidden column Update header

Personal ID	First name	Last name	Dept.
11	22	3	4





Element variable name: **s2_t2**

Hidden

< Apply

Once the table definition is complete, click the 'Apply' button

Clear table definition

+ Add column

Remove column

Remove row

Header text: Dept.

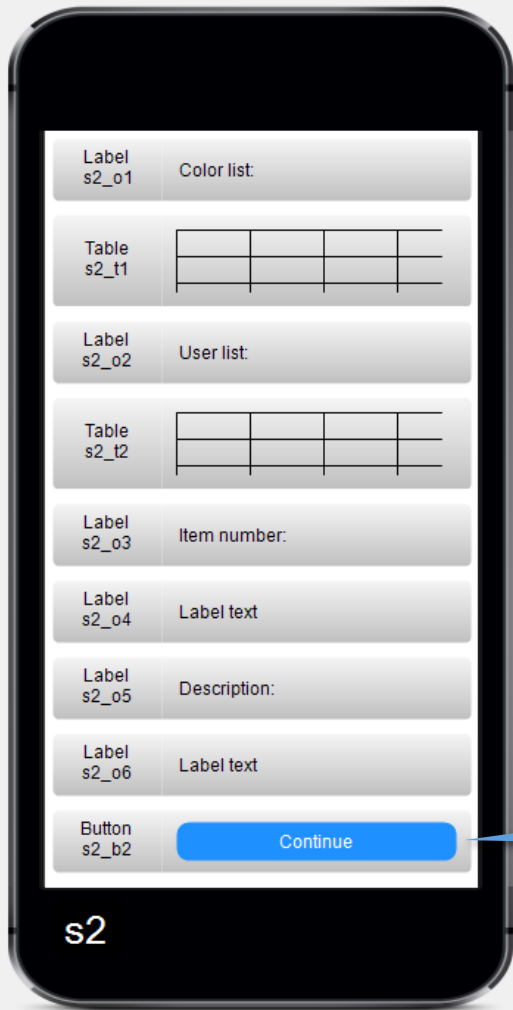
System name: department

Hidden column

Update header

Personal ID	First name	Last name	Dept.
11	22	3	4





Element variable name: **s2_b2**

Button text:

Automatic (jump to the target screen)

Default (SYS_RESULT == "") Alternative (SYS_RESULT != "")

Target screen: **s3** Initialize Transfer

ACTION SEQUENCES

- Transfer `${s2_t1[colorName]} / ${s2_t1[hexValue]}` ->
- Transfer `${s2_t2[nameFirst]} ${s2_t2[nameLast]}` -> **s3**

2. Set the sequence target screen to s3

3. Add two 'Transfer' actions using which we will transfer the selected table values from the screen s2 to screen s3:

```

${s2_t1[colorName]} / ${s2_t1[hexValue]} ==> s3_o2
${s2_t2[nameFirst]} ${s2_t2[nameLast]} ==> s3_o4

```

1. Select the button

PROCE55 Mobile Modeler [C:\Users\boris\AppData\Roaming\EastGate\Mobile Modeler\Mobile Process\REST_JSON_Service.p55m]

File View Support

Screen: **X** s2 + Element: **X** Label +

Process name: REST JSON Demo Version: 3 ^

Description: RESTful service demo Password:

Origin ID: EG_DEFAULT

Deploy to Portal Build App QR Code

PROCE55

Element variable name: **s2_b2**

Button text: < Apply

Automatic (jump to the target screen)

Default (SYS_RESULT == "") Alternative (SYS_RESULT != "")

Target screen: s3 Initialize **X** Transfer +

Transfer `${s2_t1[colorName]} / ${s2_t1[hexValue]} ->`

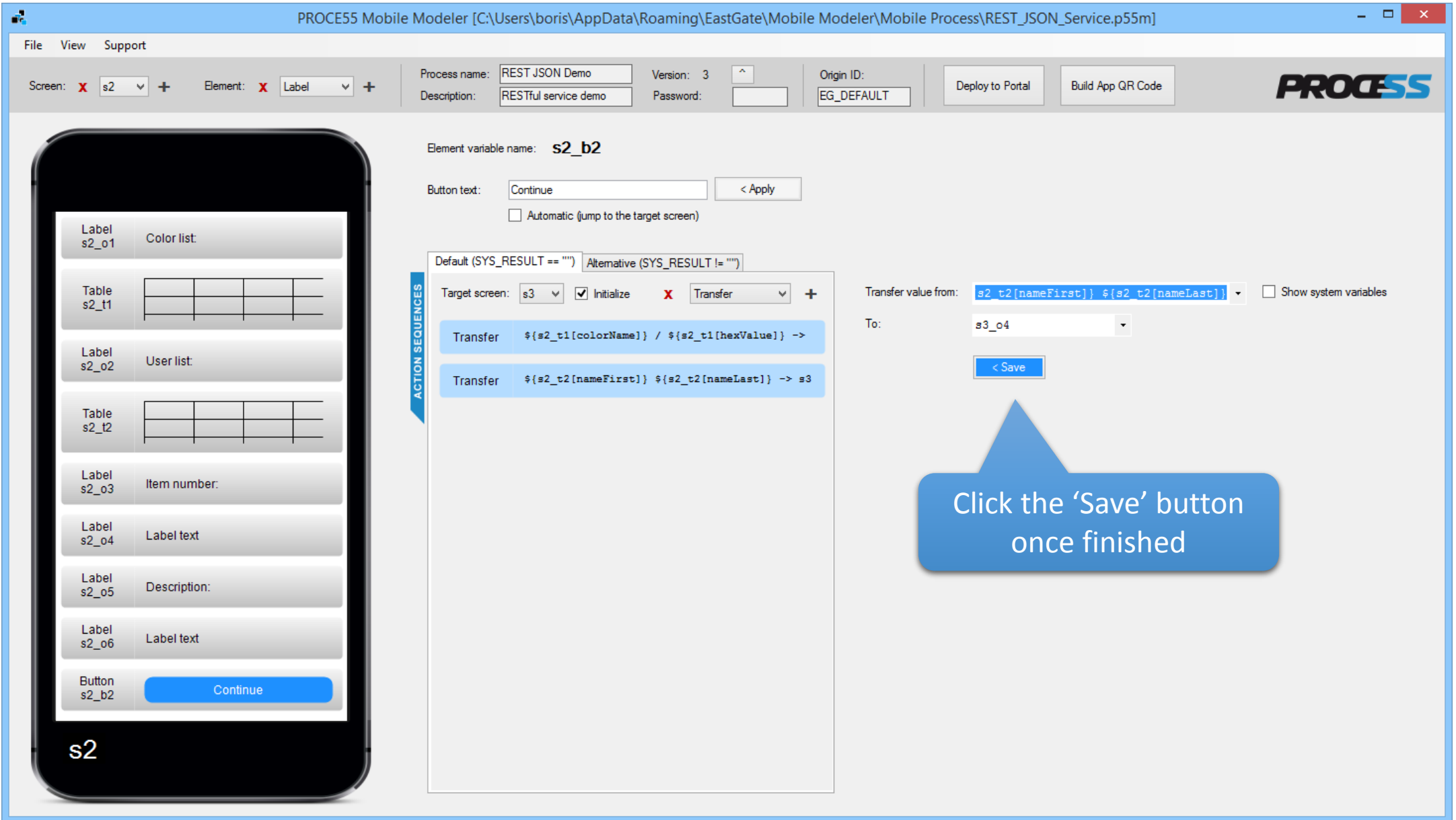
Transfer `${s2_t2[nameFirst]} ${s2_t2[nameLast]} -> s3`

Transfer value from: `s2_t2[nameFirst]} ${s2_t2[nameLast]}` Show system variables

To: s3_o4

< Save

Click the 'Save' button once finished



The screenshot displays the PROCE55 Mobile Modeler interface. On the left, a mobile phone mockup shows screen 's2' with several UI elements: a label 's2_o1' with text 'Color list', a table 's2_t1', a label 's2_o2' with text 'User list', another table 's2_t2', a label 's2_o3' with text 'Item number:', a label 's2_o4' with text 'Label text', a label 's2_o5' with text 'Description:', a label 's2_o6' with text 'Label text', and a button 's2_b2' with text 'Continue'. The main configuration area on the right shows the element variable name 's2_b2' and the button text 'Continue'. Below this, there are two 'Transfer' actions in the 'ACTION SEQUENCES' section. The first action is for the 'Default' state (SYS_RESULT == "") and the second is for the 'Alternative' state (SYS_RESULT != ""). Both actions are set to 'Transfer' and target screen 's3'. The first action transfers the values of 'colorName' and 'hexValue' from 's2_t1'. The second action transfers the values of 'nameFirst' and 'nameLast' from 's2_t2'. A 'Save' button is visible below the configuration, and a blue callout box points to it with the text 'Click the 'Save' button once finished'.

1. Switch to the last screen (s3)

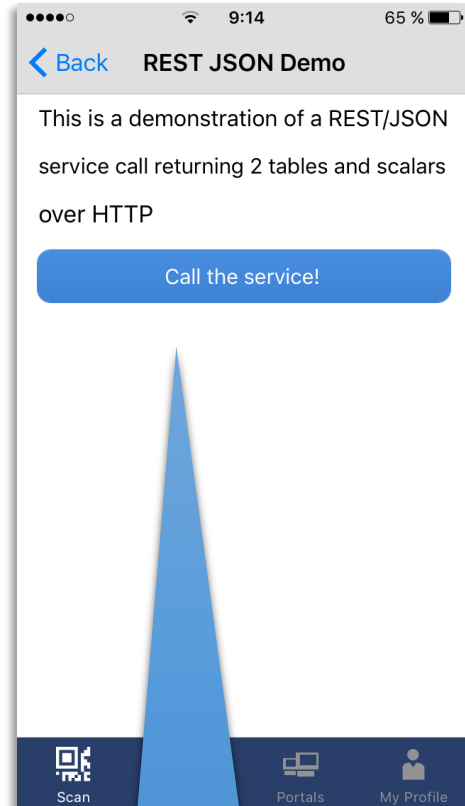
The screenshot shows the ProcessMaker Mobile Modeler interface. The top menu bar includes 'File', 'View', and 'Support'. The main toolbar contains fields for 'Screen: s3', 'Element: Label', 'Process name: REST JSON Demo', 'Version: 3', 'Origin ID: EG_DEFAULT', and buttons for 'Deploy to Portal' and 'Build App QR Code'. The 'PROCESSESS' logo is in the top right corner.

The central workspace is titled 'Element variable name: s3_b1'. It includes a 'Button text: Exit' field with an '< Apply' button and an unchecked checkbox for 'Automatic (jump to the target screen)'. Below this is the 'ACTION SEQUENCES' section, which has two tabs: 'Default (SYS_RESULT == "")' and 'Alternative (SYS_RESULT != "")'. The 'Default' tab is active, showing a table with one row: 'System' | 'Exit'. To the right of the table are 'Target screen: s1', a checked 'Initialize' checkbox, an unchecked 'Transfer' checkbox, and a '+', 'System: Exit', and '< Save' buttons.

On the left, a mobile device preview shows screen 's3' with five elements: 'Label s3_o1 Selected color:', 'Label s3_o2 Label text', 'Label s3_o3 Selected user:', 'Label s3_o4 Label text', and 'Button s3_b1 Exit'.

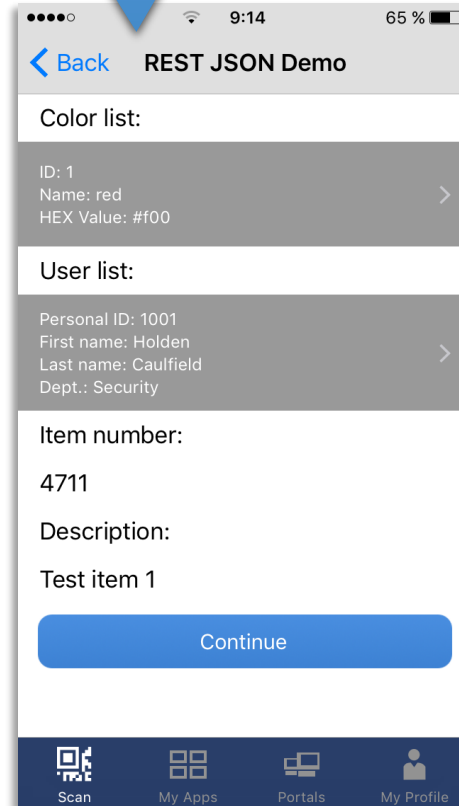
2. Add the system 'Exit' action and click the 'Save' button

1. Open the process on your mobile device

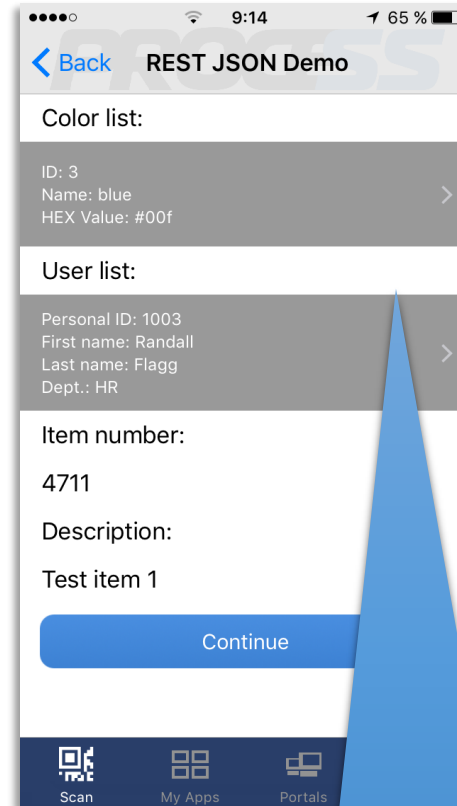


2. Tap the button to call the service

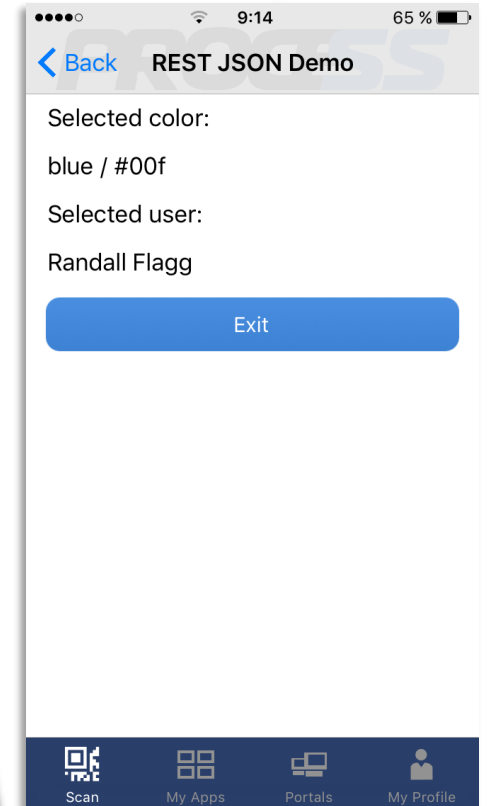
3. The service has returned two tables and two scalar values



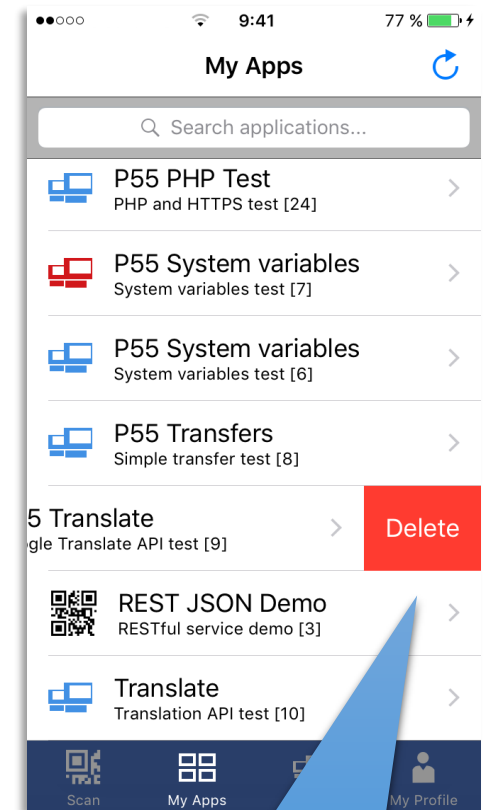
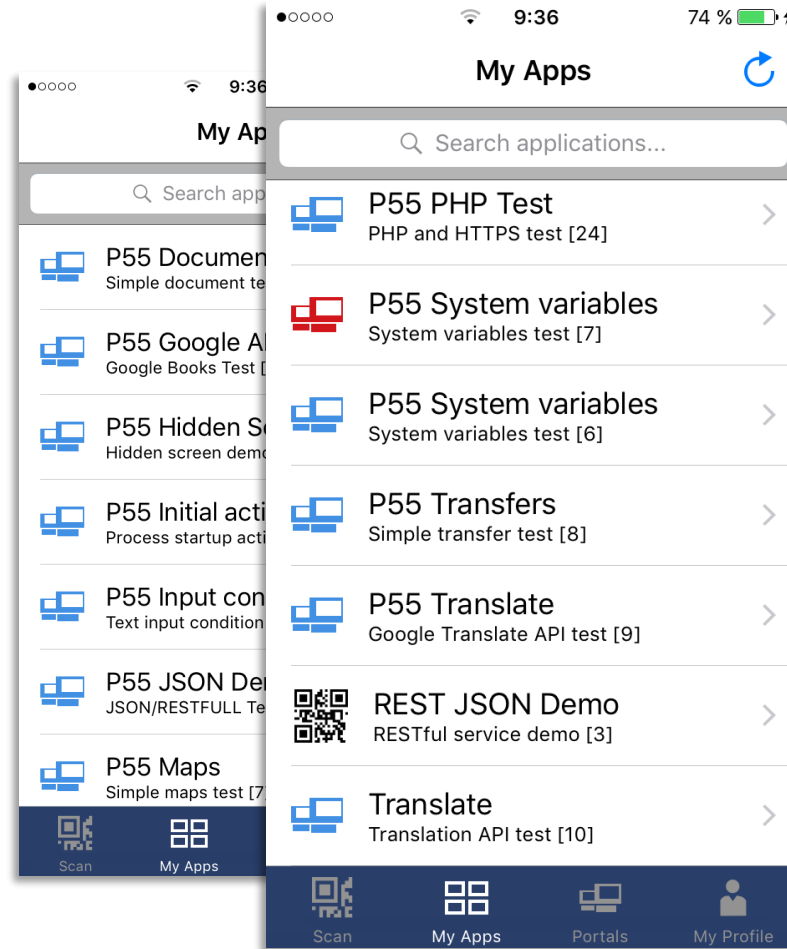
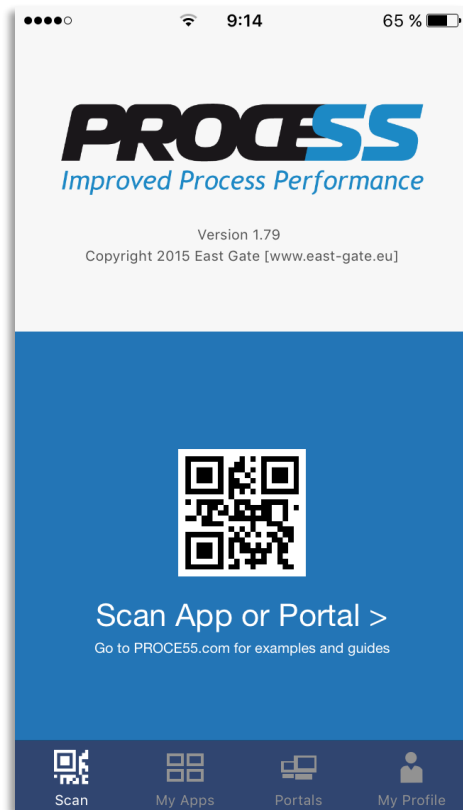
4. Change the preselected table rows by tapping the table elements



5. We have now transferred the selected table values to the last screen (s3). Tap the 'Exit' button.



Final notes:



You can access all the apps you have imported from portals or QR codes using the 'My Apps' tab

You can also remove apps you no longer need in the list