

Calling a RESTful service and processing the JSON or XML_HTTP response

Part 1: Deployment of the service package using Amazon's Elastic Beanstalk

Part 2: Building a mobile app to call the service

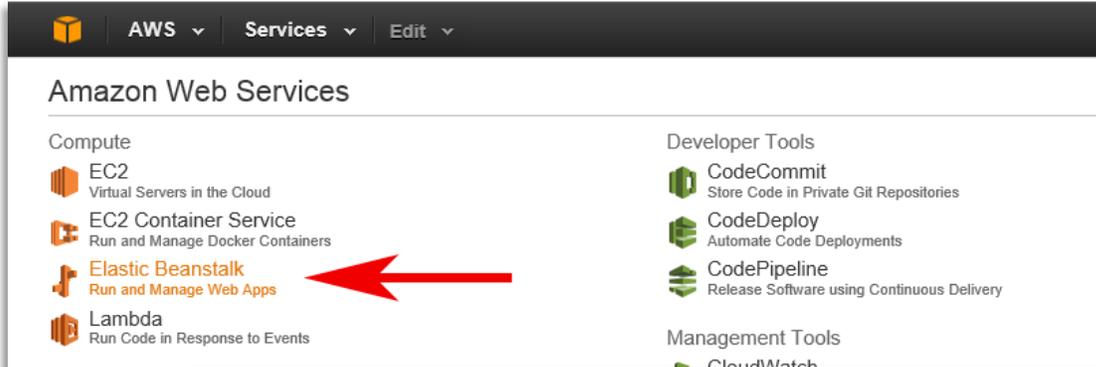
Part 3: Running the app on a mobile device

Part 4: Final tips on using the PROCE55 Player iOS application

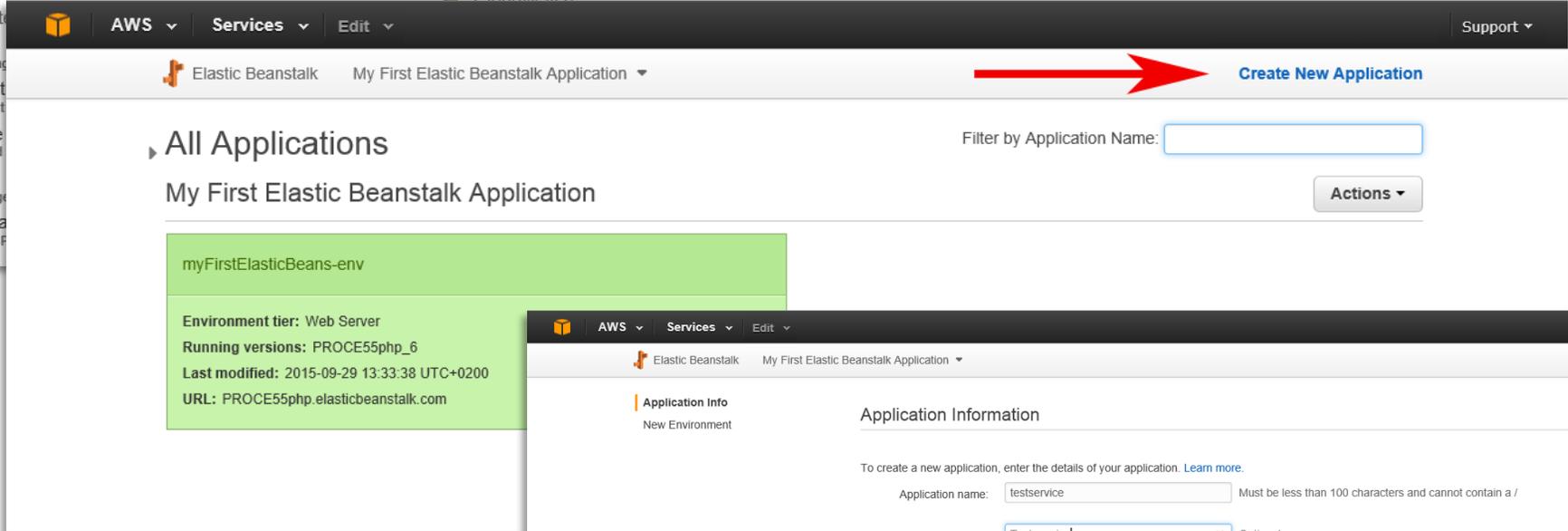
Calling RESTful service and processing the JSON or XML_HTTP response

Part 1: Deployment of the service package using Amazon's Elastic Beanstalk

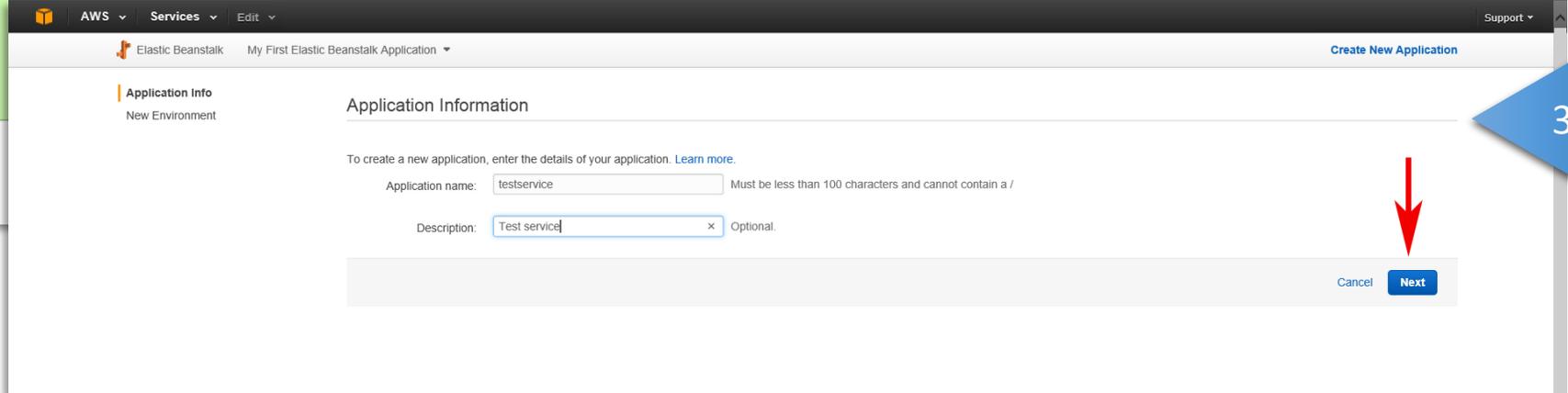
Creating and deployment of a testing AWS Elastic Beanstalk application



1



2

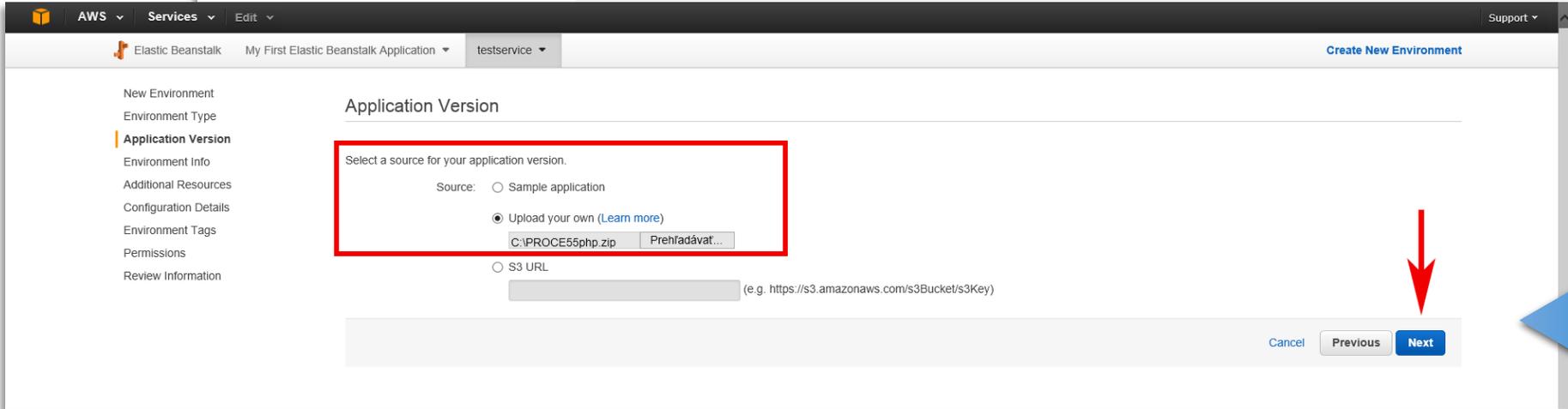
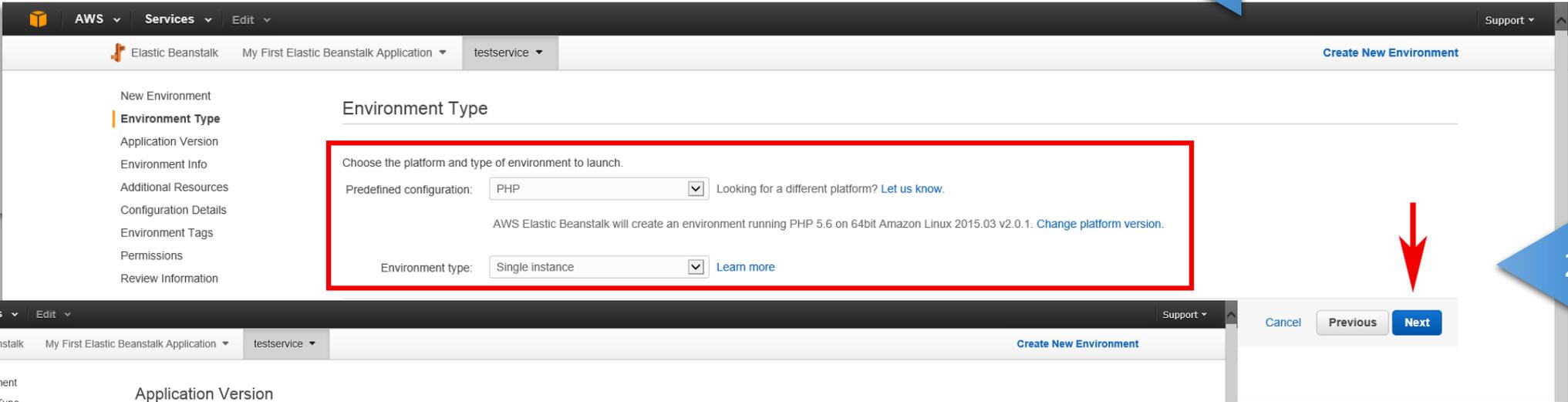
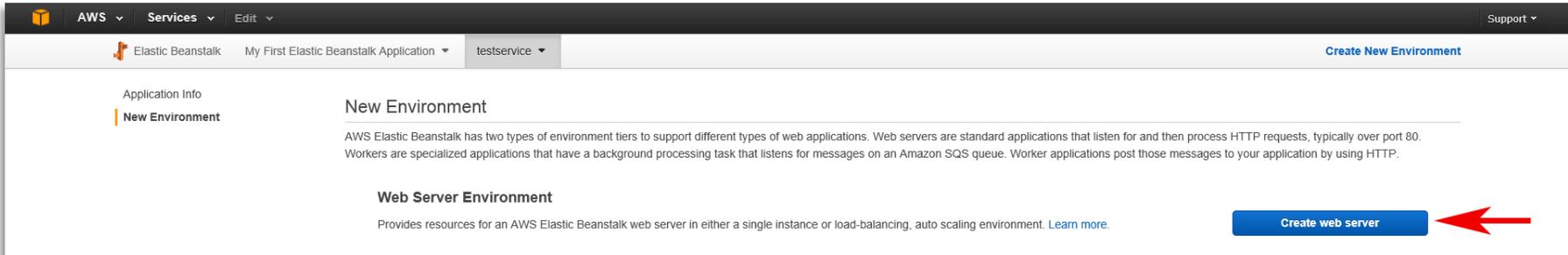


3

Calling RESTful service and processing the JSON or XML_HTTP response

Part 1: Deployment of the service package using Amazon's Elastic Beanstalk

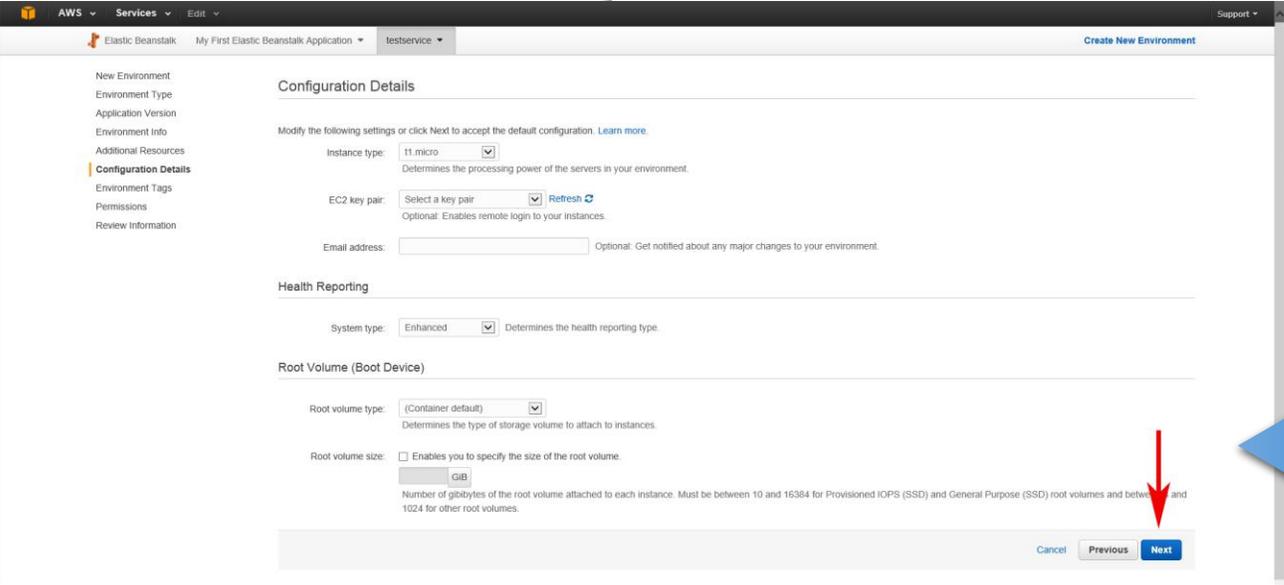
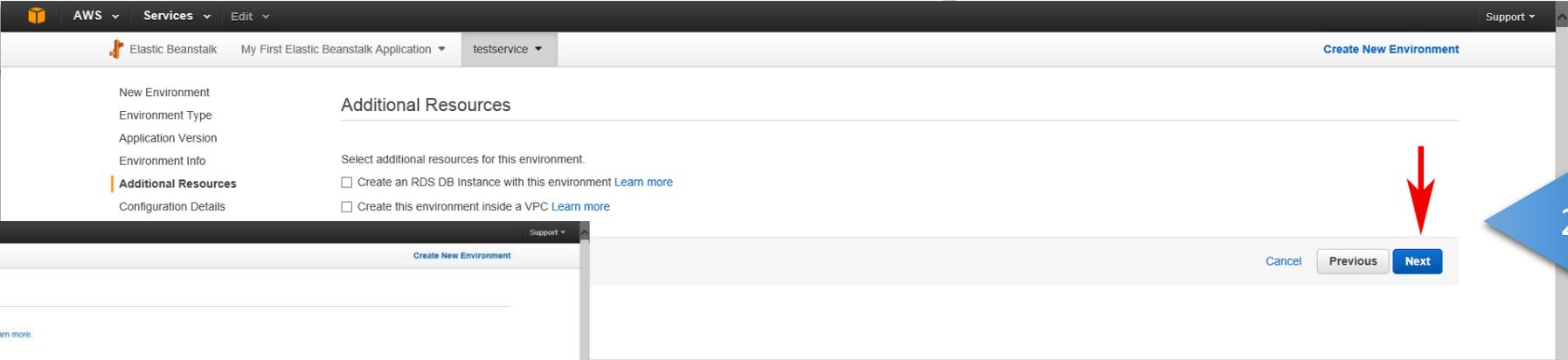
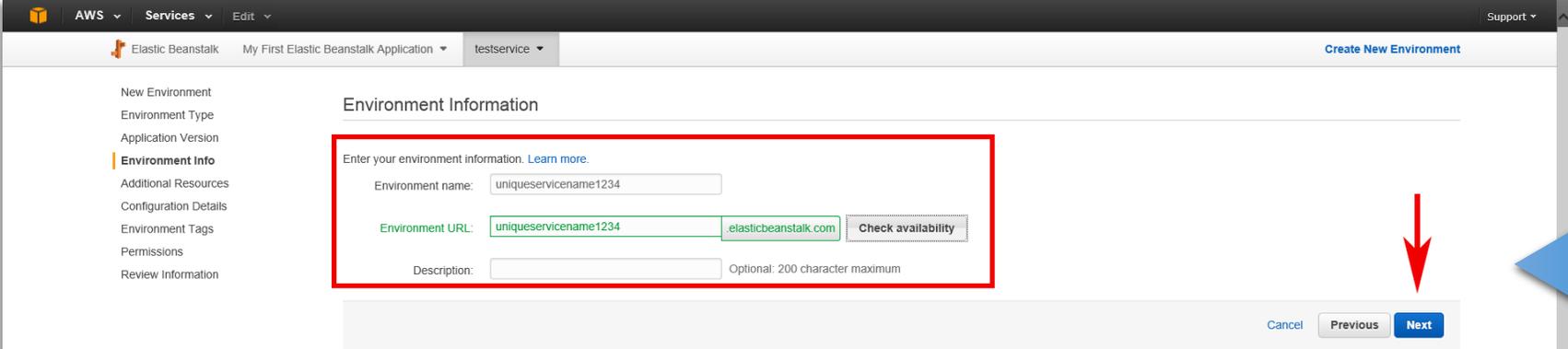
Creating and deployment of a testing AWS Elastic Beanstalk application



Calling RESTful service and processing the JSON or XML_HTTP response

Part 1: Deployment of the service package using Amazon's Elastic Beanstalk

Creating and deployment of a testing AWS Elastic Beanstalk application



Calling RESTful service and processing the JSON or XML_HTTP response

Part 1: Deployment of the service package using Amazon's Elastic Beanstalk

Creating and deployment of a testing AWS Elastic Beanstalk application

This screenshot shows the 'Environment Tags' step in the AWS Elastic Beanstalk console. The page title is 'Create New Environment'. On the left, a navigation menu lists: New Environment, Environment Type, Application Version, Environment Info, Additional Resources, Configuration Details, Environment Tags (selected), Permissions, and Review Information. The main content area is titled 'Environment Tags' and includes the instruction: 'You can specify tags (key-value pairs) for your Environment. You can add up to 7 unique key-value pairs for each Environment.' Below this is a table with two columns: 'Key (128 characters maximum)' and 'Value (256 characters maximum)'. A single row is visible with the number '1.' in the first column and empty input fields in the second. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons. A red arrow points to the 'Next' button, and a blue triangle with the number '1' is positioned to the right.

This screenshot shows the 'Permissions' step in the AWS Elastic Beanstalk console. The page title is 'Create New Environment'. The navigation menu on the left is the same as in the previous step. The main content area is titled 'Permissions' and includes the instruction: 'Select an instance profile and service role for your AWS Elastic Beanstalk environment.' Below this, there are two paragraphs of explanatory text. The 'Instance profile' dropdown menu is set to 'aws-elasticbeanstalk-ec2-role' and the 'Service role' dropdown menu is set to 'aws-elasticbeanstalk-service-'. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons. A red arrow points to the 'Next' button, and a blue triangle with the number '2' is positioned to the right.

This screenshot shows the 'Review' step in the AWS Elastic Beanstalk console. The page title is 'Create New Environment'. The navigation menu on the left is the same as in the previous steps. The main content area is divided into several sections: 'Application source' (C:\takepath\PROCE55php.zip), 'Environment Info' (Environment name: uniqueservicename1234, Environment URL: http://uniqueservicename1234.elasticbeanstalk.com), 'Configuration Details' (Instance type: t1.micro, Key pair, Email address, Root volume type, Root volume size, Root volume IOPS, Application health check URL), 'Environment Tags' (No settings provided), and 'Permissions' (Service role: aws-elasticbeanstalk-service-role, Instance profile: aws-elasticbeanstalk-ec2-role). At the bottom right, there are 'Cancel', 'Previous', and 'Launch' buttons. A red arrow points to the 'Launch' button, and a blue triangle with the number '3' is positioned to the right.

Calling RESTful service and processing the JSON or XML_HTTP response

Part 1: Deployment of the service package using Amazon's Elastic Beanstalk

Creating and deployment of a testing AWS Elastic Beanstalk application

The screenshot shows the AWS Elastic Beanstalk console interface. At the top, there is a navigation bar with 'AWS', 'Services', and 'Edit' menus. Below it, the breadcrumb path is 'Elastic Beanstalk > My First Elastic Beanstalk Application > testservice'. A blue information box at the top center states: 'Elastic Beanstalk is now creating your environment. When it has finished it will be running uniqueservicename1234Version.' Below this, the application name 'testservice' is followed by 'uniqueservicename1234' and its URL '(uniqueservicename1234.elasticbeanstalk.com)'. A red arrow points to this URL. The left sidebar contains navigation options: Dashboard, Configuration, Logs, Monitoring, Alarms, Events, and Tags. The main content area is titled 'Overview' and includes a 'Refresh' button. It features four key metrics: Health (Ok), Running Version (uniqueservicename1234Version), Configuration (64bit Amazon Linux 2015.03 v2.0.1 running PHP 5.6), and a 'php' logo. Below these are buttons for 'Causes', 'Upload and Deploy', and 'Change'. A 'Recent Events' section at the bottom contains a table with columns for Time, Type, and Details.

Time	Type	Details
2015-09-29 16:43:22 UTC+0200	INFO	Successfully launched environment: uniqueservicename1234
2015-09-29 16:42:51 UTC+0200	INFO	Environment health has transitioned from Pending to Ok.
2015-09-29 16:41:51 UTC+0200	INFO	Added instance [i-6a1b0aaf] to your environment.
2015-09-29 16:41:38 UTC+0200	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
2015-09-29 16:40:51 UTC+0200	INFO	Environment health has transitioned to Pending. There are no instances.

Calling RESTful service and processing the JSON or XML_HTTP response

Part 1: Deployment of the service package using Amazon's Elastic Beanstalk

Check if your service application is available using a web browser. As you can see, based on the 'mode' parameter in the URL the service returns either an XML_HTTP (default) or a JSON response.

```
http://uniqueservicename1234.elasticbeanstalk.com/

<?xml version="1.0"?>
- <XML_HTTP>
  - <XML_HTTP_RESPONSE>
    - <RETURN_PARAMETERS>
      <MATNR>0987654321</MATNR>
      <TEXT>Test material text</TEXT>
      <PHPVER>PHP 5.6.10 (1443537887)</PHPVER>
    </RETURN_PARAMETERS>
  - <TABLES>
    - <TABLE1>
      <ID>1111</ID>
      <MATNR>4687513202</MATNR>
      <TEXT>Material 1</TEXT>
    </TABLE1>
    - <TABLE1>
      <ID>2222</ID>
      <MATNR>9157821211</MATNR>
      <TEXT>Material 2</TEXT>
    </TABLE1>
  </TABLES>
</XML_HTTP_RESPONSE>
</XML_HTTP>
```

```
http://uniqueservicename1234.elasticbeanstalk.com/test?mode=JSON

1 {
2   "TABLE1": [{
3     "ID": "111",
4     "MATNR": "9157821211",
5     "TEXT": "Test material 111"
6   },
7   {
8     "ID": "222",
9     "MATNR": "9157821212",
10    "TEXT": "Test material 222"
11  },
12  {
13    "ID": "333",
14    "MATNR": "9157821213",
15    "TEXT": "Test material 333"
16  }
17  ],
18  "scalars": [{
19    "MATNR": "0987654321",
20    "TEXT": "Test material text",
21    "PHPVER": "PHP 5.6"
22  }
23  ]
24 }
```

```
http://uniqueservicename1234.elasticbeanstalk.com/test?mode=XML_HTTP

<?xml version="1.0"?>
- <XML_HTTP>
  - <XML_HTTP_RESPONSE>
    - <RETURN_PARAMETERS>
      <MATNR>0987654321</MATNR>
      <TEXT>Test material text</TEXT>
      <PHPVER>PHP 5.6.10 (1443537920)</PHPVER>
    </RETURN_PARAMETERS>
  - <TABLES>
    - <TABLE1>
      <ID>1111</ID>
      <MATNR>4687513202</MATNR>
      <TEXT>Material 1</TEXT>
    </TABLE1>
    - <TABLE1>
      <ID>2222</ID>
      <MATNR>9157821211</MATNR>
      <TEXT>Material 2</TEXT>
    </TABLE1>
  </TABLES>
</XML_HTTP_RESPONSE>
</XML_HTTP>
```

Calling RESTful service and processing the JSON or XML_HTTP response

Part 2: Building a mobile app to call the service

The screenshot shows the PROCES5 Mobile Modeler application window. The title bar reads "PROCES5 Mobile Modeler [C:\Users\boris\AppData\Roaming\EastGate\Mobile Modeler\Mobile Process\PHP_Service_AWS_ElasticBeanstalk.p55m]". The interface includes a menu bar (File, View, Support) and a control panel with the following fields:

- Screen: X s1 +
- Element: X Label +
- Process name: PHP Service Demo
- Version: 24
- Origin ID: EG_DEFAULT
- Description: RESTful PHP Amazon Elast
- Password: [empty]
- Buttons: Deploy to Portal, Build App QR Code

The PROCES5 logo is visible in the top right corner. On the left, a mobile phone mockup displays a screen with the following elements:

- Label s1_01: Choose the service execution mode:
- Table s1_t1: A 2x4 grid table.
- Button s1_b2: Call the service!
- Button s1_b1: Quit

A blue call-to-action box on the right contains the text: "Open the 'PHP Service Demo' app using the PROCES5 Mobile Modeler (file name: PHP_Service_AWS_ElasticBeanstalk.p55m). You can download both from PROCES5.com".

Calling RESTful service and processing the JSON or XML_HTTP response

Part 2: Building a mobile app to call the service

The screenshot shows the PROCES5 Mobile Modeler interface. On the left, a mobile phone mockup displays a screen 's1' with a label 'Choose the service execution mode:', a table 's1_t1', and two buttons: 'Call the service!' and 'Quit'. A blue callout bubble points to the table in the mockup. On the right, the configuration panel for element 's1_t1' is shown. It includes a 'Hidden' checkbox, an 'Apply' button, and a 'Clear table definition' button. Below these are buttons for '+ Add column', 'Remove column', and 'Remove row'. A table defines the columns:

Call mode	Info	Response type
XML_HTTP	Service will be called using XML_HTTP mode	1
JSON	Service will be called in JSON mode	2

Select the first table on the screen s1 (s1_t1) and examine the table column definition. Table columns have the visible Header text (shown to the user) and the internal system name used as a parameter in action sequences (e.g. 'Response type' / resp_type).

Calling RESTful service and processing the JSON or XML_HTTP response

Part 2: Building a mobile app to call the service

The screenshot shows the PROCES5 Mobile Modeler interface. On the left, a mobile app design for screen 's1' is shown, featuring a label 'Choose the service execution mode:', a table 's1_t1', and two buttons: 'Call the service!' (s1_b2) and 'Quit' (s1_b1). A blue callout bubble points to the 'Call the service!' button with the text: 'Select the 's1_b2' button and inspect the action sequence'.

The main workspace shows the configuration for element 's1_b2'. The 'Button text' is 'Call the service!'. The 'Target screen' is 's2'. The 'Action SEQUENCES' list contains two items:

- Condition: $\#{s1_t1[resp_type]}=1$
- Service: PHP_AWS
- Condition: $\#{s1_t1[resp_type]}=2$
- Service: PHP_AWS

Three blue callout bubbles provide additional context:

- Top bubble: 'Once the action sequence is completed, we will end up on the target screen (s2 in this case).' (Points to the 'Target screen' dropdown)
- Middle bubble: 'This service will only be called if the enclosing condition is true' (Points to the first 'Service' row)
- Bottom bubble: 'Similarly, the other service will be called if we choose the second option from the s1_t1 table (resp_type=2, which is the JSON response type in this case)' (Points to the second 'Service' row)

The interface also shows a 'Condition' field with the value $\#{s1_t1[resp_type]}=1$ and a '< Save' button.

Calling RESTful service and processing the JSON or XML_HTTP response

Part 2: Building a mobile app to call the service

The screenshot shows the PROCES5 Mobile Modeler interface. On the left, a mobile app design is shown with a screen labeled 's1' containing a label 'Choose the service execution mode:', a table 's1_t1', and two buttons: 'Call the service!' (s1_b2) and 'Quit' (s1_b1). The main workspace shows the configuration for the 'Call the service!' button. The 'Service' is set to 'PHP_AWS' with 'XML_HTTP' as the 'Service type'. The 'URL' is 'http://proce55php.elasticbeanstalk.com/test'. The 'Parameters' table is as follows:

Type	Name	Value
Import	mode	#{s1_t1[mode]}
Export	MATNR	s2_o2
Export	TEXT	s2_o4
Export	PHPVER	s2_o5
Export	TABLE1	s2_t1

The 'Action Sequences' section shows two conditions and two service calls. The first condition is 'Condition #{s1_t1[resp_type]}=1' followed by 'Service PHP_AWS'. The second condition is 'Condition #{s1_t1[resp_type]}=2' followed by 'Service PHP_AWS'. A blue callout bubble points to the 'Service type' dropdown with the text: 'The service request and response will be processed using the XML_HTTP format'. Another blue callout bubble points to the 'Parameters' table with the text: 'The service call request (POST body) will contain one parameter (mode) and the response should contain three scalar values (MATNR, TEXT, PHPVER) and a table (TABLE1)'.

Calling RESTful service and processing the JSON or XML_HTTP response

Part 2: Building a mobile app to call the service

The screenshot shows the PROCES5 Mobile Modeler interface. On the left is a mobile app preview for screen 's1' with a label 'Choose the service execution mode:', a table 's1_t1', and two buttons 'Call the service!' and 'Quit'. The main workspace shows the configuration for element 's1_b2' (Button text: 'Call the service!'). The 'ACTION SEQUENCES' panel shows two paths: a default path with condition `#{s1_t1[resp_type]}=1` and an alternative path with condition `#{s1_t1[resp_type]}=2`, both leading to the 'PHP_AWS' service. The service configuration on the right is for 'PHP_AWS' using 'JSON_GET' with URL 'http://proce55php.elasticbeanstalk.com/test'. A parameters table is shown below:

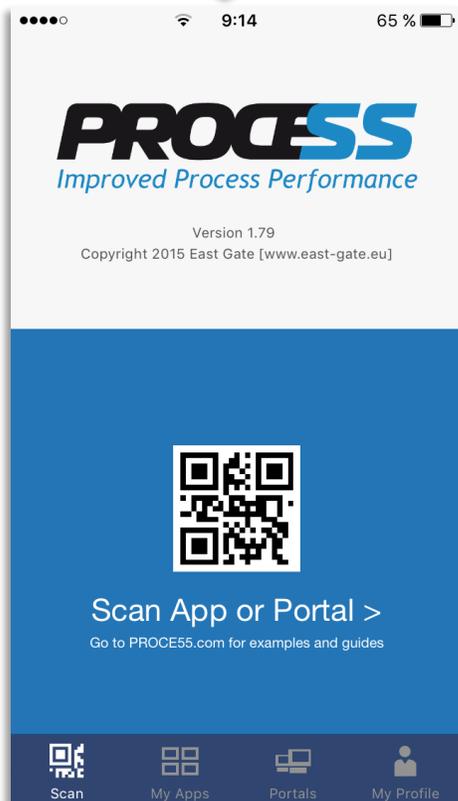
Type	Name	Value
Import	mode	#{s1_t1[mode]}
Export	MATNR	s2_o2
Export	TEXT	s2_o4
Export	PHPVER	s2_o5
Export	TABLE1	s2_t1

Two blue callout boxes provide additional information: 'This service request will be called using the HTTP GET' and 'The service call response should be in JSON format'.

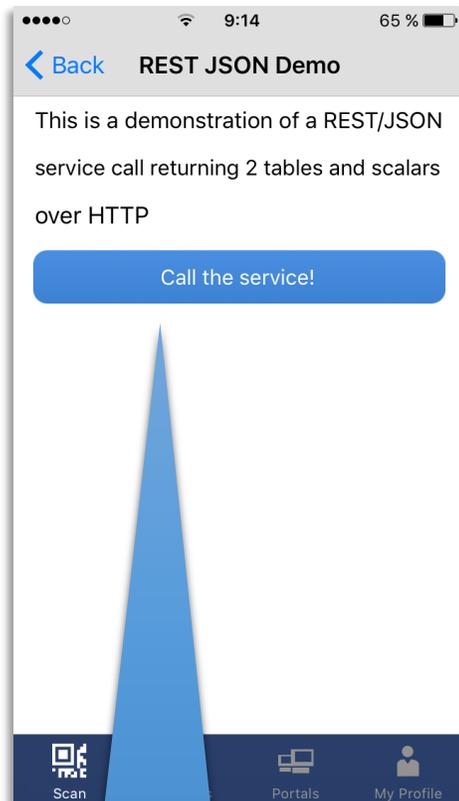
Calling RESTful service and processing the JSON or XML_HTTP response

Part 3: Running the app on a mobile device

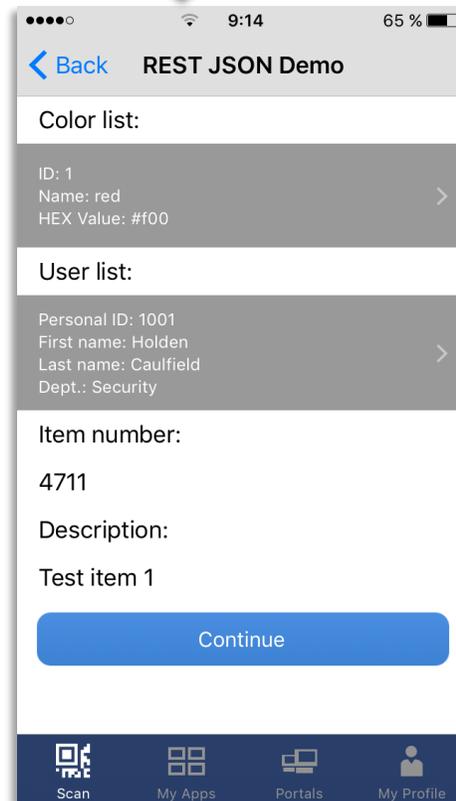
1. Scan the app QR code Using the 'PROCE55 Player' application on your mobile device



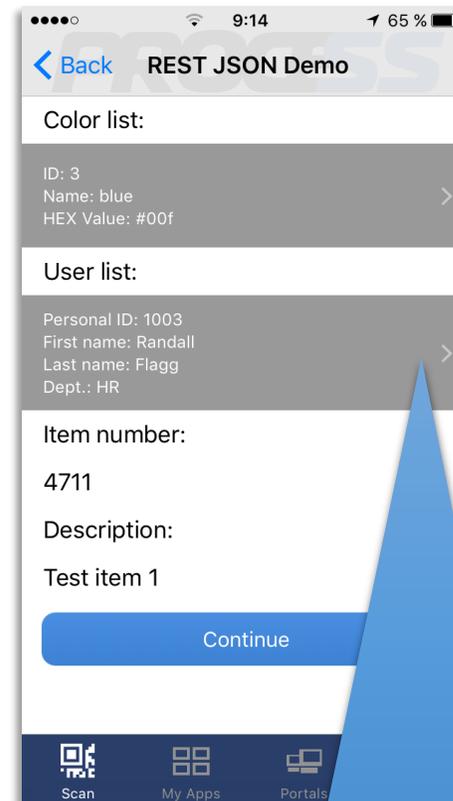
2. Tap the button to call the service



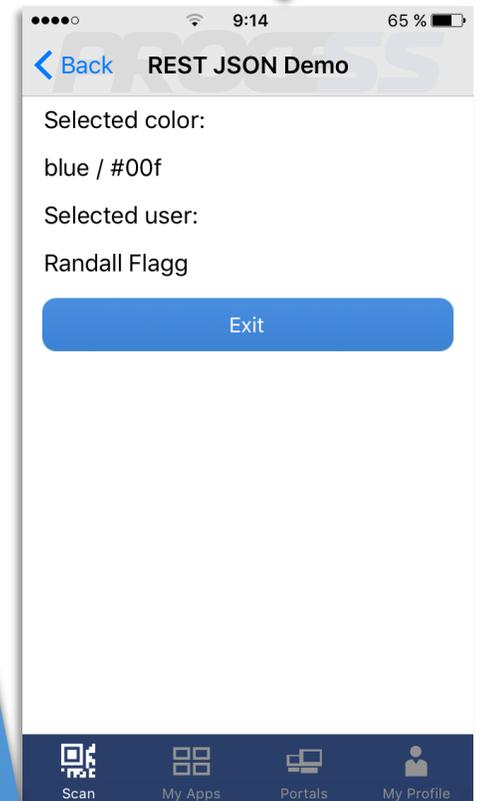
3. The service has returned two tables and two scalar values



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

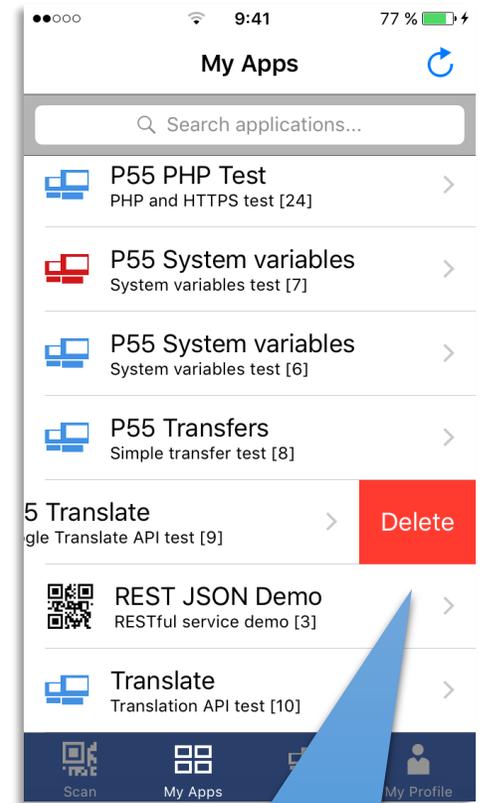
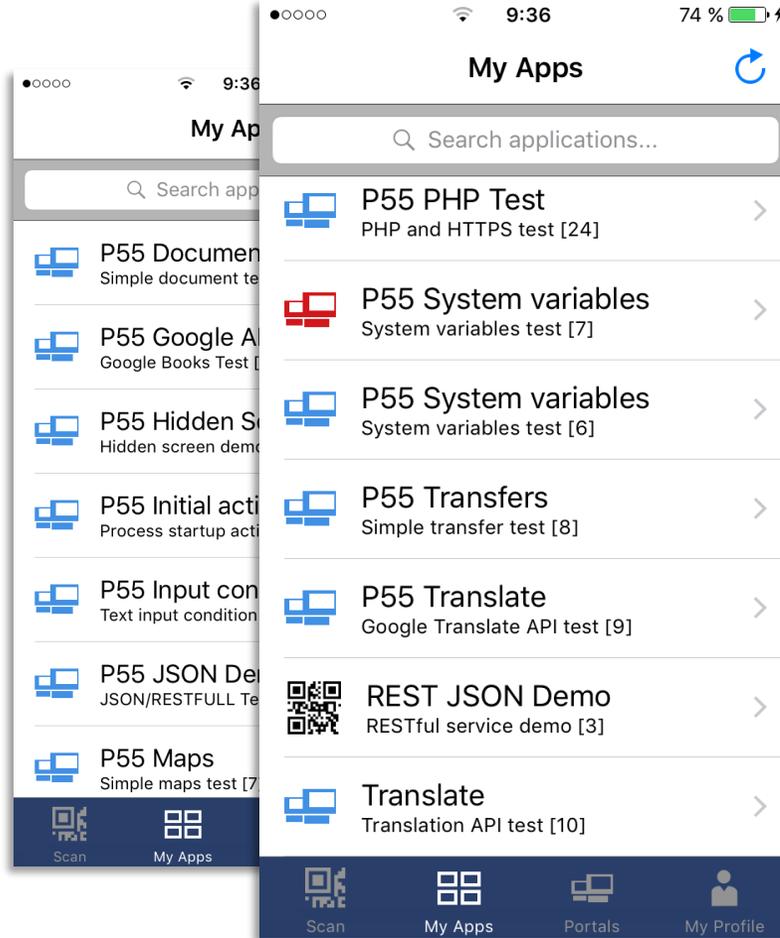
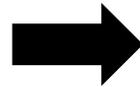
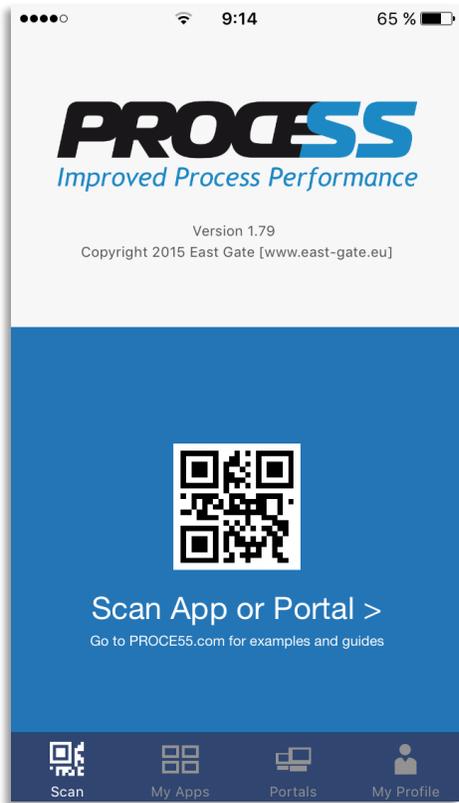


5. We have now transferred the selected table values to the last screen (s3). Tap the 'Exit' button.



Calling RESTful service and processing the JSON or XML_HTTP response

Part 4: Final tips on using the PROCE55 Player iOS application



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