

Ejercicio 1 - Build a Dinamic List – Orlando Sánchez Acuña

Part 1 – Create the Getting Started Objects

Create and Populate the Address Database Table

The screenshot shows two instances of MySQL Workbench. The top window displays the SQL editor with the creation of the 'address' table:

```
1 CREATE TABLE `address`(
2     `address_id` INT(11) NOT NULL,
3     `shipping_address` VARCHAR(100),
4     `unit_number` INT(5),
5     `city` VARCHAR(100),
6     `state_or_province` VARCHAR(2),
7     `postal_code` INT(5)
8 );
9 ALTER TABLE `address`
10 ADD PRIMARY KEY(`address_id`);
11 ALTER TABLE `address`
12 MODIFY `address_id` INT(11) NOT NULL AUTO_INCREMENT;
```

The bottom window shows the SQL editor with the insertion of data into the 'address' table:

```
1 INSERT INTO `address`(`shipping_address`, `unit_number`,
2 `state_or_province`, `postal_code`, `city`) VALUES ('47 New
3 Saddle Ave.', NULL, 'SC', 29445, 'Goose Creek');
4 INSERT INTO `address`(`shipping_address`, `unit_number`,
5 `state_or_province`, `postal_code`, `city`) VALUES ('7208 Oakland
6 Drive', 50, 'OH', 44266, 'Revenna');
7 INSERT INTO `address`(`shipping_address`, `unit_number`,
8 `state_or_province`, `postal_code`, `city`) VALUES ('358 Mayfair
9 Dr.', 23, 'MA', 01902, 'Lynn');
```

Create Groups, a CDT, and a Data Store

Create Group

Name ***Description****Parent Group** **Group Members****Group Type** **Visibility** 

Create Group

Name ***Description****Parent Group** ADV All Users  **Group Members****Group Type** **Visibility**

Create Data Type

- Create from scratch
- Duplicate existing data type
- Create from database table or view
- Import XSD

Data Source *

jdbc/Appian (Tomcat)

Table or View*

address

CANCEL

CONTINUE

ADV Interfaces Practice IMPORT PACKAGE PACKAGES COMPARE AND DEPLOY appian

Create Data Type

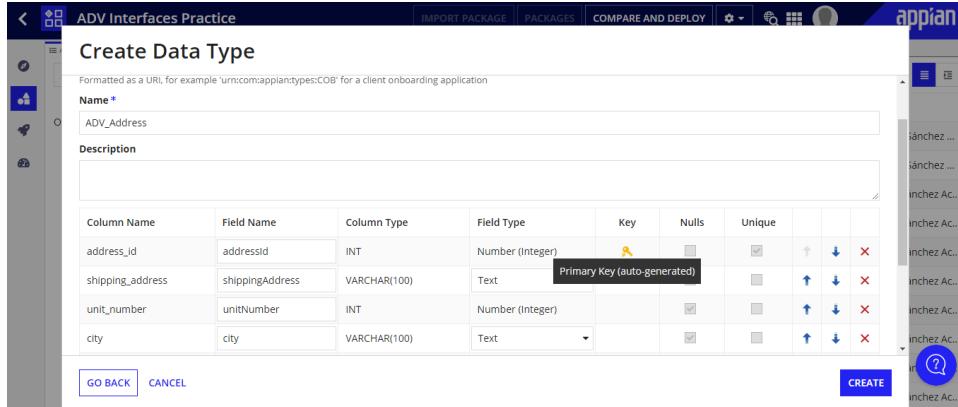
Formatted as a URI, for example 'urn:com:appian:types:COB' for a client onboarding application

Name * ADV_Address

Description

Column Name	Field Name	Column Type	Field Type	Key	Nulls	Unique		
address_id	addressId	INT	Number (Integer)			<input checked="" type="checkbox"/>		
shipping_address	shippingAddress	VARCHAR(100)	Text	Primary Key (auto-generated)				
unit_number	unitNumber	INT	Number (Integer)					
city	city	VARCHAR(100)	Text					

GO BACK CANCEL CREATE



Create Data Store

Name *

ADV DS

Description

Data Store for the ADV app

CANCEL

CREATE



Tip: Basic users must have at least viewer rights to the data store in order to use its entities to query, write, or delete data. [Learn more](#)

Name

ADV DS

User or Group	Permission Level	
Default (All Other Users)	No Access	
ADV All Administrators ✖	Administrator	✖
ADV All Users ✖	Viewer	✖

[+ Add Users or Groups](#)



CANCEL

SAVE CHANGES

Create a Constant and an Expression to Query Addresses

Create Constant

Create from scratch Duplicate existing constant

Name *

ADV_DSE_ADDRESS

Description

Type *

Data Store Entity

Array (multiple values)

Data Store *

ADV DS

Entity *

ADV_Address



Environment Specific

CANCEL

CREATE

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name *

ADV_getAddresses

Description

Save In *

ADV Expression Rules



ADV Expression Rules

CANCEL

CREATE



The screenshot shows the Appian Rule Editor interface. On the left is a code editor containing Java-like pseudocode for a rule named ADV_getAddresses. The code uses `alquery` to query records and `alqueryLogicalExpression` to filter them. It defines variables like `entity`, `pagingInfo`, and `operator`. The `filters` section contains a query filter for the `addressId` field. The right side of the interface includes a sidebar with sections for 'Local Variables' and 'Test Output'. Below the sidebar is a table titled 'RULE INPUTS' with one row for 'addressId'. A status bar at the bottom indicates a time of 73 ms.

Part 2 – Build the Interface with the Dynamic Component

Create an Interface with Radio Buttons

Create Interface

Create from scratch Duplicate existing interface

Name *

ADV_DynamicAddressList

Description

Save In *

ADV Interfaces

CANCEL

CRF

Configure the Rule Inputs and Local Variables

The screenshot shows the Appian Design Mode interface for the 'ADV_DynamicAddressList' interface. On the left is a palette with components like 'RADIOS' and 'READ-ONLY GRID'. In the center, there is a 'Radio Buttons' component with two options: 'Option 1' and 'Option 2'. To the right is a configuration panel for the 'Radio Buttons' component. The configuration panel includes sections for 'COMPONENT CONFIGURATION' (with 'Label Position' set to 'Hidden'), 'Instructions', and 'Help Tooltip'.

New Rule Input

Name *

address

Type *

ADV_Address X

Array (multiple values)

CANCEL

CREATE

The screenshot shows the Appian interface definition for a rule input named "ADV_DynamicAddress...". The interface includes sections for "DESIGN MODE" and "EXPRESSION MODE". In the "EXPRESSION MODE" section, there is a code editor with the following code:

```
1. allocalVariables(
2.   locationsList: rule!ADV_getAddresses(addressId: NULL),
3.
4. )
28 )
```

Below the code editor, there is a note about "allocVariables" and "localVarN". The "allocVariables" note states: "Lets you define one or more local variables for use within an expression. When used within an interface, the value of each variable can be refreshed under a variety of conditions, configured using a!refreshVariable(). When used outside of an interface, all refresh properties configured using a!refreshVariable() are ignored." The "localVarN" note states: "The local variable to use when evaluating the given expression. Use the localVarN code editor definition to reference individual variables. By default, a local variable will automatically update whenever the variables it references are changed. To change the way variables are updated, use the a!refreshVariable() function. Variables can be refreshed under the following conditions: after each reevaluation, periodically on an interval, or when other variables change." There is also a note about "localVarN (Any Type)": "An unlimited number of local variables."

The right side of the interface shows the "RULE INPUTS" panel, which contains a table with a single row for "address" with a value of "null". Below this is a "LOCAL VARIABLES" section with a table showing a list of addresses from a database table "locationsList". The table has columns "Name" and "Value" and rows numbered 1 through 6, each containing an address ID and its corresponding shipping address.

Configure the Labels and Values for Radio Buttons

The screenshot shows the Appian interface definition for a rule input named "ADV_DynamicAddress...". The interface includes sections for "DESIGN MODE" and "EXPRESSION MODE". In the "EXPRESSION MODE" section, there is a code editor with the following code:

```
1. allocalVariables(
2.   locationsList: rule!ADV_getAddresses(addressId: NULL),
3.
4. {
5.   selectionLayout: {
6.     label: "Address",
7.     contents: [
8.       radioButtonField(
9.         label: "Radio Buttons",
10.        labelPosition: "COLLAPSED",
11.        choiceLabels: locationsList,
12.        choiceValues: locationsList,
13.        saveInto: {},
14.        choiceLayout: "STACKED",
15.        validations: {}
16.      )
17.   }
18. }
19. )
20. 
```

Below the code editor, there is a note about "radioButtonField": "label, instructions, required, disabled, choiceLabels, choiceValues, value, validations, saveInto, validationGroup, requiredMessage, labelPosition, choiceLayout, helpText, accessibilityText, showWhen, choiceStyle". The note states: "Displays a limited set of choices from which the user must select one item and saves a value based on the selected choice. See documentation for more details." There is also a note about "label": "Text to display as the field label." and "instructions": "Text: Supplemental text about this field." There are also notes for "required" and "disabled".

The right side of the interface shows the "RULE INPUTS" panel, which contains a table with a single row for "address" with a value of "null". Below this is a "LOCAL VARIABLES" section with a table showing a list of addresses from a database table "locationsList". The table has columns "Name" and "Value" and rows numbered 1 through 6, each containing an address ID and its corresponding shipping address.

Format Addresses Using the a!for Each Function

ADV_DynamicAddress...

DESIGN MODE **EXPRESSION MODE** **TEST** **SAVE CHANGES**

INTERFACE DEFINITION

```
1. allocalVariables(
2.   localLocationsList: rule!ADV_getAddresses(addressId: NULL),
3. )
4. selectionLayout
5.   label: "Address",
6.   contents: {
7.     radioButtonField(
8.       label: "Radio Buttons",
9.       labelPosition: "COLLAPSED",
10.      choiceLabels:
11.        forEach(items: localLocationsList,
12.          expression: fv!item,
13.        )
14.      )
15.    )
16.  
```

alforeach items, expression

Evaluates the provided expression once for every item and returns an array of the results.

Items (List of Variant): Array or DataSubset containing the items over which to iterate.

expression (Any Type): The expression to be evaluated with each iteration. The following special variables are available: fvIndex (the current array item), fvIndexCount (the current item's index in the array), fvItemCount (the number of items), fvIdentifier (when using a DataSubset, the current item's identifier), fvIsFirst (true during the first iteration, false afterward), fvIsLast (true during the last iteration, false beforehand).

Address

- [addressId=3, shippingAddress=358 Mayfair Dr., unitNumber=23, city=Lynn, stateOrProvince=MA, postalCode=1902]
- [addressId=6, shippingAddress=358 Mayfair Dr., unitNumber=23, city=Lynn, stateOrProvince=MA, postalCode=1902]
- [addressId=1, shippingAddress=47 New Saddle Ave., unitNumber=1, city=Goose Creek, stateOrProvince=SC, postalCode=29445]
- [addressId=4, shippingAddress=47 New Saddle Ave., unitNumber=50, city=Ravenna, stateOrProvince=OH, postalCode=44266]
- [addressId=2, shippingAddress=7208 Oakland Drive, unitNumber=50, city=Ravenna, stateOrProvince=OH, postalCode=44266]
- [addressId=5, shippingAddress=7208 Oakland Drive, unitNumber=50, city=Ravenna, stateOrProvince=OH, postalCode=44266]

RULE INPUTS

Name	Value
address	null

LOCAL VARIABLES

Name	Value
locationsList	[addressId=3, shi...]
[1]	[addressId=3, shi...]
[2]	[addressId=6, shi...]
[3]	[addressId=1, shi...]
[4]	[addressId=4, shi...]
[5]	[addressId=2, shi...]
[6]	[addressId=5, shi...]

ADV_DynamicAddress...

DESIGN MODE **EXPRESSION MODE** **TEST** **SAVE CHANGES**

INTERFACE DEFINITION

```
1. allocalVariables(
2.   localLocationsList: rule!ADV_getAddresses(addressId: NULL),
3. )
4. selectionLayout
5.   label: "Address",
6.   contents: {
7.     radioButtonField(
8.       label: "Radio Buttons",
9.       labelPosition: "COLLAPSED",
10.      choiceLabels:
11.        forEach(items: localLocationsList,
12.          expression: fv!item.shippingAddress,
13.        )
14.      )
15.    )
16.  
```

alforeach items, expression

Evaluates the provided expression once for every item and returns an array of the results.

Items (List of Variant): Array or DataSubset containing the items over which to iterate.

expression (Any Type): The expression to be evaluated with each iteration. The following special variables are available: fvIndex (the current array item), fvIndexCount (the current item's index in the array), fvItemCount (the number of items), fvIdentifier (when using a DataSubset, the current item's identifier), fvIsFirst (true during the first iteration, false afterward), fvIsLast (true during the last iteration, false beforehand).

Address

- 358 Mayfair Dr.
- 358 Mayfair Dr.
- 47 New Saddle Ave.
- 47 New Saddle Ave.
- 7208 Oakland Drive
- 7208 Oakland Drive

RULE INPUTS

Name	Value
address	null

LOCAL VARIABLES

Name	Value
locationsList	[addressId=3, shi...]

ADV_DynamicAddress...

DESIGN MODE **EXPRESSION MODE** **TEST** **SAVE CHANGES**

INTERFACE DEFINITION

```
1. Layout(Address,
2.   s: {
3.     radioButtonField(label: "Radio Buttons",
4.       labelPosition: "COLLAPSED",
5.       choiceLabels:
6.         forEach(items: localLocationsList,
7.           expression: fv!item.shippingAddress & " Unit # " & fv!item.unit#,
8.           if(isNull(fv!item.shippingAddress), null, " Unit # " & fv!item.unit#),
9.           )
10.      )
11.    )
12.  )
13.  iceValues: localLocationsList.
14.  )
15. )
16. )
17. )
18. )
```

if(condition, valueIfTrue, valueIfFalse)

Returns valueIfTrue if condition returns true; returns valueIfFalse otherwise.

Returns: Any Type

condition (Boolean): A test that determines whether valueIfTrue or valueIfFalse will be returned.

valueIfTrue (Any Type): The value to be returned if condition returns true.

valueIfFalse (Any Type): The value to be returned if condition returns false.

Address

- 358 Mayfair Dr. Unit #23
- 358 Mayfair Dr. Unit #23
- 47 New Saddle Ave. Unit #
- 47 New Saddle Ave. Unit #
- 7208 Oakland Drive Unit #50
- 7208 Oakland Drive Unit #50

RULE INPUTS

Name	Value
address	null

LOCAL VARIABLES

Name	Value
locationsList	[addressId=3, shi...]
[1]	[addressId=3, shi...]
[2]	[addressId=6, shi...]
[3]	[addressId=1, shi...]
[4]	[addressId=4, shi...]
[5]	[addressId=2, shi...]
[6]	[addressId=5, shi...]

ADV_DynamicAddress...

DESIGN MODE **EXPRESSION MODE** **TEST** **Changes saved**

INTERFACE DEFINITION

```
1. Layout(Address,
2.   s: {
3.     radioButtonField(label: "Radio Buttons",
4.       labelPosition: "COLLAPSED",
5.       choiceLabels:
6.         forEach(items: localLocationsList,
7.           expression: fv!item.shippingAddress & " Unit # " & fv!item.unit#,
8.           if(isNull(fv!item.shippingAddress), null, " Unit # " & fv!item.unit#),
9.           & " & " & fv!item.city & ", " & fv!item.stateOrProvince
10.          )
11.        )
12.      )
13.    )
14.  )
15.  iceValues: localLocationsList.
16.  )
17. )
18. )
```

alforeach items, expression

Evaluates the provided expression once for every item and returns an array of the results.

Items (List of Variant): Array or DataSubset containing the items over which to iterate.

expression (Any Type): The expression to be evaluated with each iteration. The following special variables are available: fvIndex (the current array item), fvIndexCount (the current item's index in the array), fvItemCount (the number of items), fvIdentifier (when using a DataSubset, the current item's identifier), fvIsFirst (true during the first iteration, false afterward), fvIsLast (true during the last iteration, false beforehand).

Address

- 358 Mayfair Dr. Unit #23 Lynn, MA 1902
- 358 Mayfair Dr. Unit #23 Lynn, MA 1902
- 47 New Saddle Ave. Unit # Goose Creek, SC 29445
- 47 New Saddle Ave. Unit # Goose Creek, SC 29445
- 7208 Oakland Drive Unit #50 Ravenna, OH 44266
- 7208 Oakland Drive Unit #50 Ravenna, OH 44266

RULE INPUTS

Name	Value
address	null

LOCAL VARIABLES

Name	Value
locationsList	[addressId=3, shi...]
[1]	[addressId=3, shi...]
[2]	[addressId=6, shi...]
[3]	[addressId=1, shi...]
[4]	[addressId=4, shi...]
[5]	[addressId=2, shi...]
[6]	[addressId=5, shi...]

Configure the Selected Value and Save Selection To Fields

ADV_DynamicAddressList

DESIGN MODE EXPRESSION MODE TEST SAVE CHANGES

EDIT PREVIEW PERFORMANCE

Address

Radio Buttons

358 Mayfair Dr. Unit #23 Lynn, MA 1902
358 Mayfair Dr. Unit #23 Lynn, MA 1902
47 New Saddle Ave. Unit # Goose Creek, SC 29445
47 New Saddle Ave. Unit # Goose Creek, SC 29445
7208 Oakland Drive Unit #50 Ravenna, OH 44266
7208 Oakland Drive Unit #50 Ravenna, OH 44266

RULE INPUTS

Name	Value
address	[addressId=1, ship...]
addressId	1
shippingAd...	47 New Saddle Ave.
unitNumber	null
city	Goose Creek
stateOrProv...	SC
postalCode	29445

LOCAL VARIABLES

COMPONENT CONFIGURATION

Contents

Radio Buttons

localLocationsList

Selected Value

riAddress

Save Selection To

riAddress

