Uniface 10 Deep Dive

Palettes and Templates

Thomas S. Shore III

TABLE of CONTENTS

1. Introduction

The purpose of this workshop is to help you to understand the power and flexibility available in Uniface 10 through the use of Palettes and a collection of templates.

2. Modifying the IDE.ASN

We need to modify the IDE.ASN to allow use to create and maintain templates and palettes.

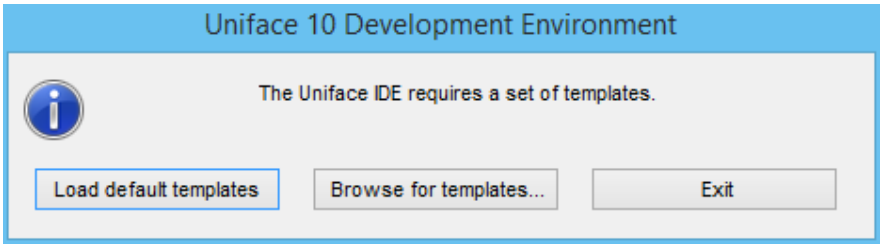
Open the IDE.ASN file located in your Uniface 10 installation folder:

uniface\adm\ide.asn

```
; By default, the templates and palettes used in the Resource Browsers cannot
; be opened in an editor. If the user wants to create and maintain user defined
; templates and palettes, the setting 'allow_browse_templates' to browse and
; open templates should be enabled: default/disabled = 0 ; enabled = 1
; NOTE: Templates and palettes provided by Uniface should not be modified
; without understanding the impact of such modifications.
allow_browse_templates = 1
```

3. Starting from an empty repository

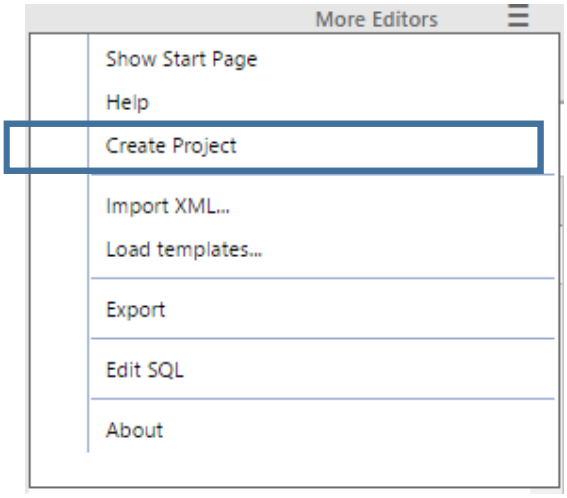
If you are starting this example/exercise from an empty repository you will be presented with the following screen:



Chose the 'Load default templates' option.

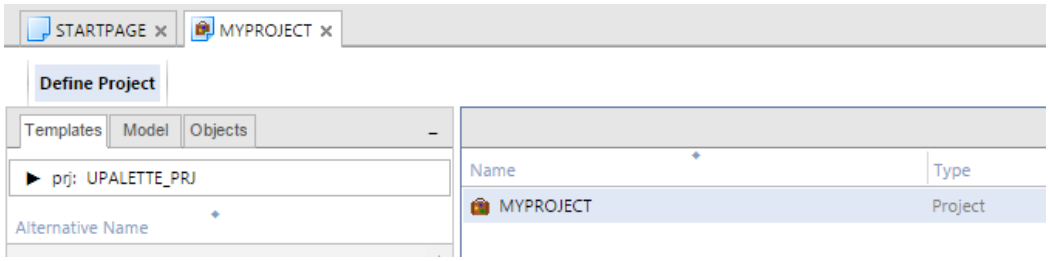
4. Creating our Project

We will need to create a project that will be used to hold our templates. From the top right-hand corner select the waffle menu and select '**Create Project**'

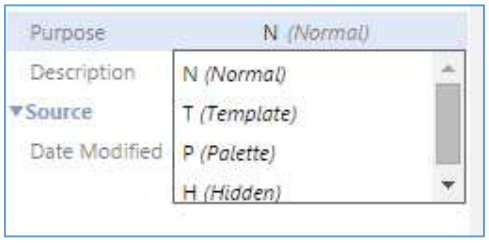


We need to create the following:

1. STANDARDS



Click on MYPROJECT in the center pane and enter STANDARDS as the name and press Enter. On the right-hand side of the screen change the Purpose from N (Normal) to P(Palette) and press Enter.



By setting the Purpose to P (Palette) you can also enter an Alternate name and Description as I have done here. Why enter alternate name?

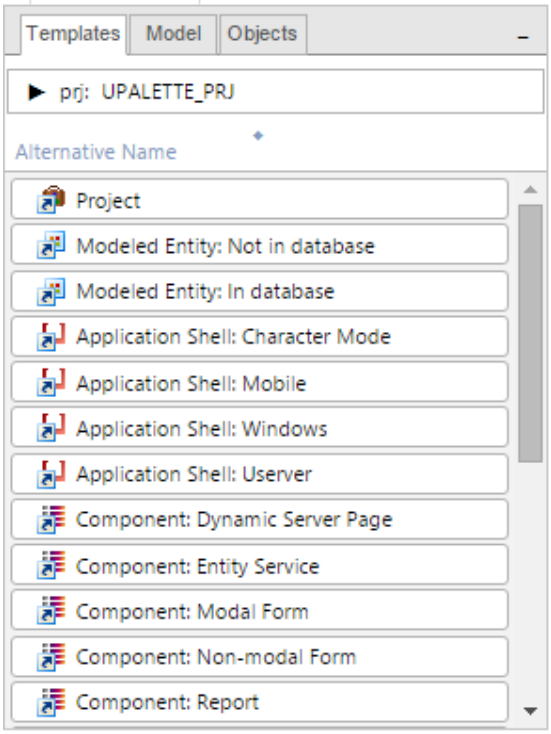
Purpose	P *(Palette)*
Alternative Name	Company Standards
Tag	
Description	Development Standards Nov 2016

BY DEFAULT, TEMPLATES AND PALETTES ARE VISIBLE FOR BROWSING ONLY IN THE TEMPLATES TAB OF THE RESOURCE BROWSERS. NORMAL DEVELOPMENT OBJECTS ARE VISIBLE WHEN BROWSING IN THE U-BAR OR RESOURCE BROWSERS SO THAT YOU CAN EASILY LOCATE AND OPEN THEM FOR EDITING OR REUSE.

5. Adding modelled components to our standards

We need to click on the STANDARDS project and select the Templates tab on the left-hand side of the screen. Click on the 

and make sure that UPALETTE_PRJ is selected. Like this:



These components will be the base implementations that will be derived to create new components.

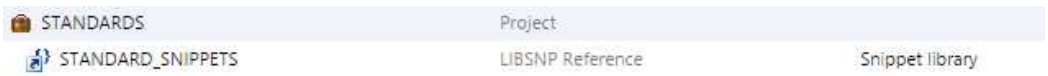
Please note that every time you rename an object you must press ENTER to accept the change.

5.1 Adding a snippet library

Drag the 

from the left-hand side of the screen onto the STANDARDS project in the center pane.

Rename the default name from "**ULIBSNP_1**" to "**STANDARD_SNIPPETS**" and press Enter.



5.2 Adding a Modeled DSP Component

Drag the 

from the left-hand side of the screen onto the STANDARDS project in the center pane.

Rename the MDSP_1 to "**RESTFULAPI_BASE**".

Select the Alternative Name field in the Properties pane and change it to "**RESTFul API Base**". Change the Description to "**RESTful API implementation**".

5.3 Adding a Modeled DSP Component for Bootstrap

Drag the 

from the left-hand side of the screen onto the STANDARDS project in the center pane.

Rename MDSP_1 to "**BOOTSTRAP_BASE**".

Select the Alternative Name field in the Properties pane and change it to "**Bootstrap DSP Base**". Change the Description to "**Bootstrap list implementation**".

5.4 Adding a Modeled Entity

Drag the 

from the left-hand side of the screen onto the STANDARDS project in the center pane.

Rename the default name of "**UENTDB_1. MODEL**" to "**TABLEWITHSTATUSFIELDS.MODEL**".

Select the Description field in the Properties pane and change it to "**Entity with Status Fields**". Repeat this for the Alternative Name field.

5.5 Definition Complete.

Your project should look like this:

6. Modifying the Modeled Entity

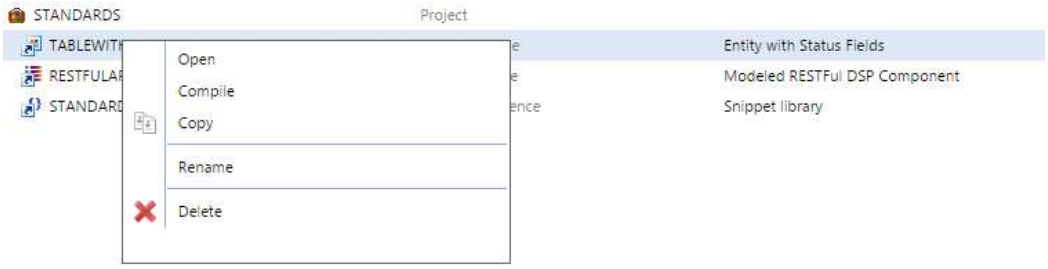
For business reasons any new table that we create must store either when the record was added and by whom or when the record was updated and by whom.

We want an entity that has 4 modeled fields named:

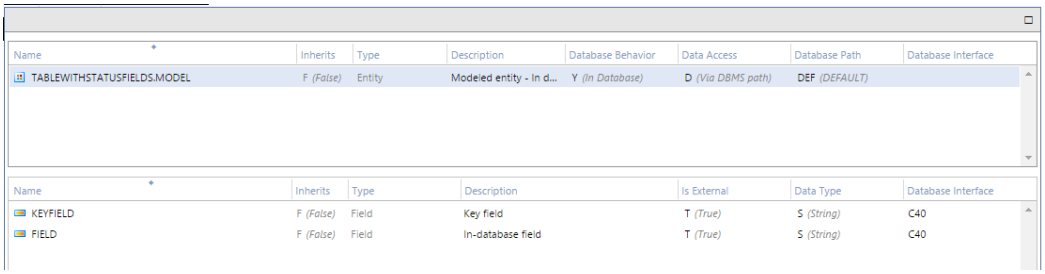
Fieldname	Data Type
CreatedOn	DateTime
CreatedBy	String 40
UpdatedOn	DateTime
UpdatedBy	String 40

We will add fields and ProcScript to the modeled entity to facilitate this business requirement.

Select the '**TABLEWITHSTATUSFIELDS.MODEL**' and right-click and then select '**Open**'

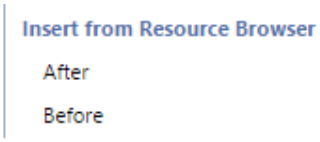


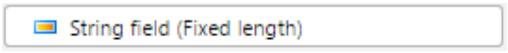
Once opened you should see the following:



Drag a 

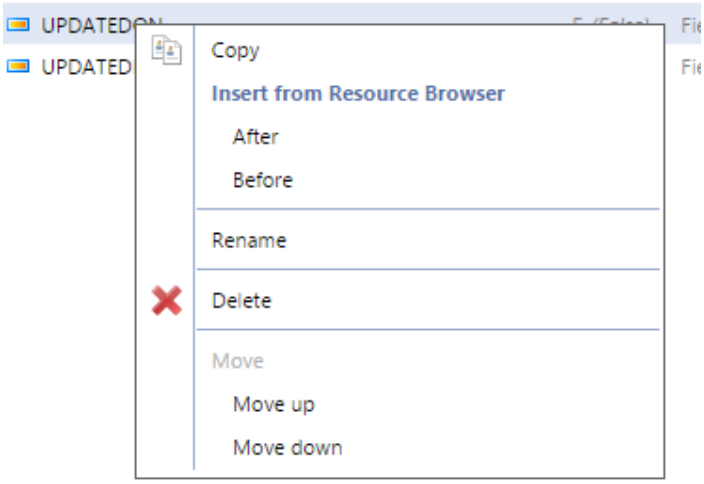
onto the fields section and select 'Insert After' then rename it to '**CreatedOn**' (Remember to press ENTER to change the name).



Drag a 

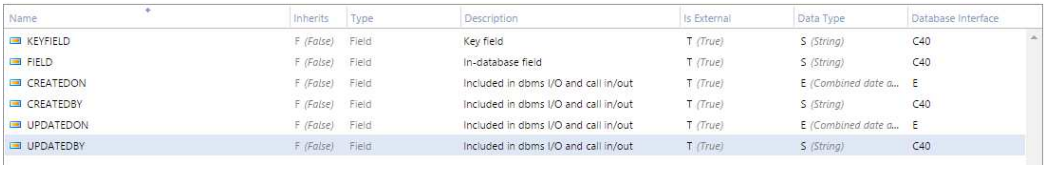
on the fields section and select 'Insert After' then rename it to "**CreatedBy**"

If by chance the fields are not in the order that you would like. You can right-click on the field and Select either 'Move Up' or 'Move Down'



Repeat this for **UpdatedOn** and **UpdatedBy**.

Your fields should look like this:



Name	Inherits	Type	Description	Is External	Data Type	Database Interface
KEYFIELD	F (False)	Field	Key field	T (True)	S (String)	C40
FIELD	F (False)	Field	In-database field	T (True)	S (String)	C40
CREATEDON	F (False)	Field	Included in dbms I/O and call in/out	T (True)	E (Combined date a...	E
CREATEDBY	F (False)	Field	Included in dbms I/O and call in/out	T (True)	S (String)	C40
UPDATEDON	F (False)	Field	Included in dbms I/O and call in/out	T (True)	E (Combined date a...	E
UPDATEDBY	F (False)	Field	Included in dbms I/O and call in/out	T (True)	S (String)	C40

6.1 Entering business logic

We now need to add the business login to populate the status fields that we created above. In order to do that we need to write some ProcScript. Click on the 'Write Script' tab located above the TABLEWITHSTATUSFIELDS.MODEL and to the right of 'Define Structure'

Click on the '**TABLEWITHSTATUSFIELDS.MODEL**' in the Structure pane.

Scroll the script window down until the `trigger write` block appears.

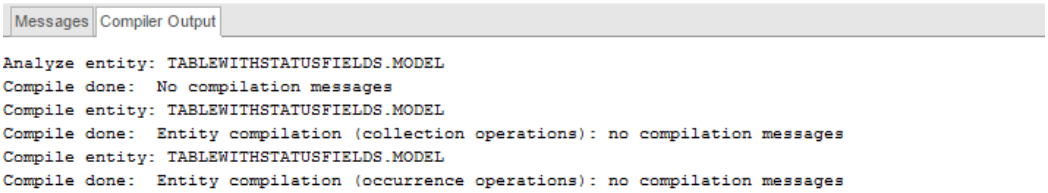
Enter the following code before the **write** statement.

```
if ($dbocc = 0)
    CREATEDBY = $user
    CREATEDON = $date
 else
    UPDATEDBY = $user
    UPDATEDON = $date
 endif
```

You write trigger should look like this:

```
trigger write
; This trigger is fired as part of the store process.
; Make sure not to leave the context of this occurrence, it would disrupt the trigger flow and give unpredictable result

  ; Your last moment field updates here...

 if ($dbocc = 0)
    CREATEDBY = $user
    CREATEDON = $date
  else
    UPDATEDBY = $user
    UPDATEDON = $date
  endif

  write

end
```

In the upper right-hand portion of the IDE below the 'More Editors' button. Click '**Compile**'. Verify that the Compiler Output tab at the bottom of the screen shows no compilation errors.



Click the 

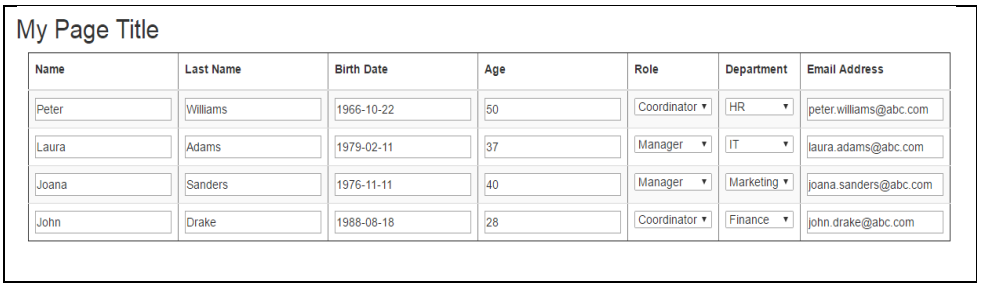
tab to navigate back to our Project.

7. Creating a Modelled DSP Component for Bootstrap

It is often very useful to create a series of standardized pages that utilize current industry frameworks. We will be creating a modelled DSP component that takes advantage of Bootstrap[1], Tablesaw[2] and jQuery[3] to produce a rich and responsive user interface.

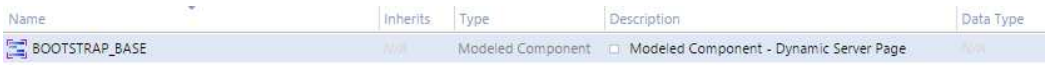
For this exercise we will utilize the EMPLOYEES entity.

7.1 Desired Result



7.2 Extending our BOOTSTRAPDSP modeled component

Right-click on the BOOTSTRAP_BASE component and select Open.



As we have done before click on the Write Script tab

The default code is populated. If there was specific code that you wanted to include as part of a Modeled DSP page you could add it here.

Click on the Design Layout tab:

[1] Bootstrap - http://getbootstrap.com/ Originally created by a designer and a developer at Twitter, Bootstrap has become one of the most popular front-end frameworks and open source projects in the world.
[2] Tablesaw - https://github.com/filamentgroup/tablesaw
[3] jQuery -https://jquery.com/ jQuery is a fast, small, and feature-rich JavaScript library

7.3 HTML Layout

Replace the code in the Design Layout with the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="default" content="Uniface 10">
  <link rel="stylesheet" type="text/css"  href="../css/uniface.css">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"   integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on
  <script src="https://code.jquery.com/jquery-3.1.1.min.js" integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8=" crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNI
  <title>This is a HTML title block</title>
</head>
<body>
<div class="container">
<h2>My Page Title</h2>
<div class="form-group col-xs-5">
    <!-- Your table should be styled like this:
        <table id="uent:EMPLOYEE.BOOTSTRAP" class="table table-striped" border="1">
    -->
</div>
</div>
</body>
</html>
```

It is also located in 'Code Samples\Bootstrap DSP HTML Layout.txt'

Click the 

tab to navigate back to our Project.

8. Creating a RESTful DSP Component

We have a business requirement that some of our company information is made available via RESTful services to the public. For the foreseeable future we only expect this information to be read-only.

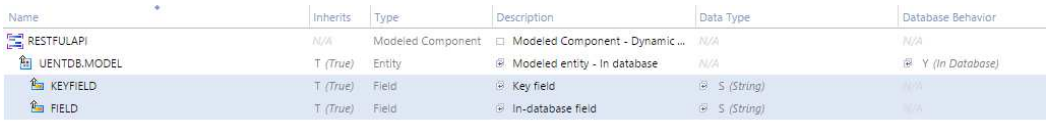
From our 'STANDARDS' project, click on the **RESTFULAPI_BASE**, right-click and select Open.

In the properties make sure that the Purpose is set to Template from Normal.

Click on the 

tab, select UENTDB and drag it onto **RESTFULAPI_BASE.**

Multi-Select the two fields on the left (KEYFIELD and FIELD) and drag them onto the **UENTDB.MODEL** resulting in the following:



MULTI-SELECT THE ENTITY AND THE FIELDS AND MAKE SURE THAT THE 'ISGENERIC' PROPERTY IS SET TO TRUE. This will allow us to bind other fields to the generic ones.

As we have done before click on the Write Script tab

Add this code to the Declarations container:

```
variables
    string output_prefix
endvariables
```

Observe the code:

```
operation exec
public web
; Your SCOPE  block here (optional)...
; Your VARIABLES block here (optional)...

  ; Your implementation here...

  ; Return execution status:
  ;   return 0    ; return >= 0 typically indicates a succesful execution
  ;   return -1   ; return < 0 typically indicates an error during execution
  return 0

end
```

Replace entire 'operation exec' with the following code located in 'Code Samples\RESTful API Code.txt'

Note: Remove the preActivate and postActivate triggers as we will not be doing any state management.

Add the **runthis** code from 'Code Samples\Restful API Processing.txt' to the end of the Script block following **getErrorMessage**

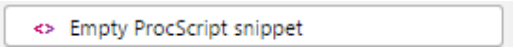
Click the 

tab to navigate back to our Project

9. Populating a Snippet library

There are several application elements that need to be added to some components to provide sorting and row highlighting on a grid. While this is not complicated it can be bothersome to copy and paste code. In order to implement this let's do the following:

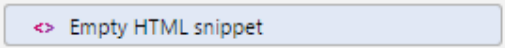
From our 'STANDARDS' click on the **STANDARD_SNIPPETS**, right-click and select Open.

Select 

and drag it onto our **STANDARD_SNIPPETS** library.

As before rename **EMPTY.PROCSCRIPT** to the name that we want, in this case '**ROWHILIGHTING.PROCSCRIPT**'.

Repeat and create another ProcScript called '**ENTITYDEFINES**'.

Create an HTML snippet called '**BOOTSTRAPBASE**' by dragging the 

onto the **STANDARD_SNIPPETS** library.

Modify the Description fields for each to be:

1. Row Hilighting
2. Entity definitions for Hilightrow
3. Bootstrap Base

ALSO MAKE SURE TO SET THE ALTERNATIVE NAME TO SOMETHING USEFUL OTHER THAN 'EMPTY HTML SNIPPET'

As we have done before click on the Write Script tab

Select 'ROWHIGHLIGHTING' and enter the following:

```
if ($curocc % 2 )
    call OccurrenceSetFieldColors( "<GRIDENTITY>","<SELECTIONCOLOR>")
endif
```

Select ‘ENTITYDEFINES’ and enter the following Procscript:

```
#define SELECTIONCOLOR = #DDDDDD
#define GRIDENTITY = MYENTITY.MODEL
```

Enter the following into the HTML snippet named BOOTSTRAPBASE. It is located in ‘Code Samples\Bootstrap DSP HTML Layout.txt’

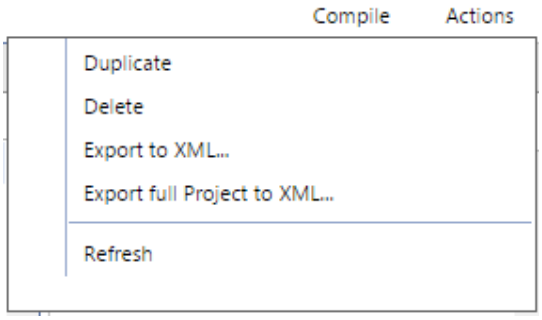
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="default" content="Uniface 10">
  <link rel="stylesheet" type="text/css"  href="../css/uniface.css">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"   integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on
  <script src="https://code.jquery.com/jquery-3.1.1.min.js" integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8=" crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNI
  <title>This is a HTML title block</title>
</head>
<body>
<div class="container">
 <h2>My Page Title</h2>
<div class="form-group col-xs-5">
    <!-- Your table should be styled like this:
        <table id="uent:EMPLOYEE.BOOTSTRAP" class="table table-striped" border="1">
    -->
</div>
</div>
</body>
</html>
```

Click the 

tab to navigate back to our Project

10. Packaging up your Standards

From the Action Menu:



Select '**Export full Project to XML…**'

Save the file to a name of your choice or use the default of '**prj_full_standards.xml**'

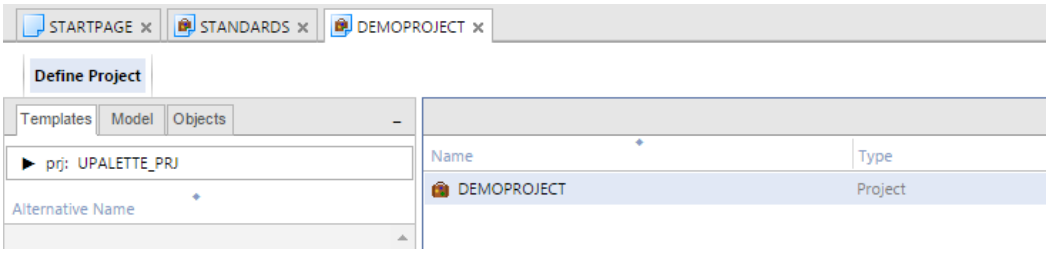
11. Applying what we have done

Stop and take a deep breath.

You can either create an empty repository or utilize the repository that you have been using to create the STANDARDS project.

We need to create a new project to leverage the work that we have done.
Create a project called DEMOPROJECT, leaving the Purpose set to N (Normal).

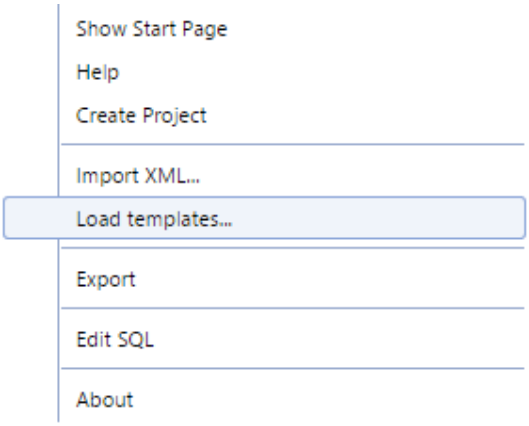
You should see the following in the IDE:

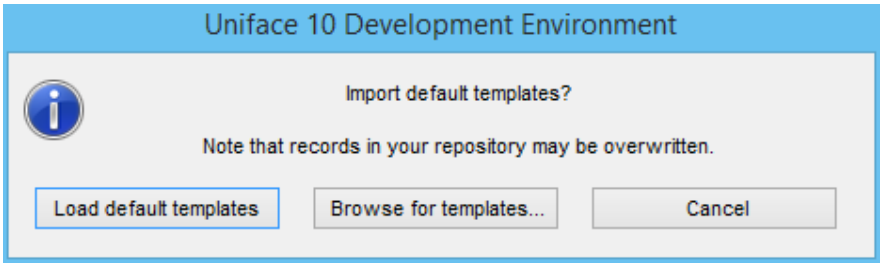


Jump to Adding some data for testing if you are using the existing repository.

11.1 Importing our standards if starting from an empty repository

Click on the upper right-hand menu  and select Load Templates





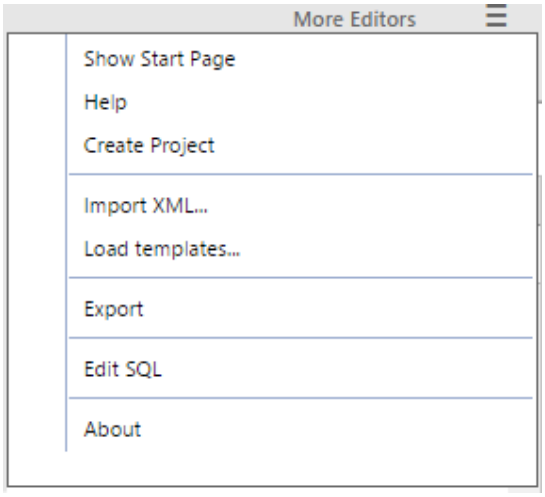
Click Browse for templates... and browse to the location that you placed your Standards export and select the file, for example: **prj_full_standards.xml**

11.2 Adding some data for testing

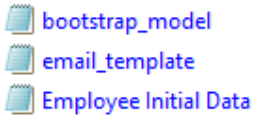
For ease of testing we will import an existing entity named: EMPLOYEES.BOOTSTRAP and load some initial data. The EMPLOYEES entity also utilizes a field syntax template named: EMAIL. Click on the DEMOPROJECT that we created earlier.

11.3 Import the EMAIL template

From the waffle menu in the upper right-hand portion of the Uniface 10 IDE select 'Import XML…'



The following files will need to be installed:



Import the following files in this order:

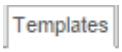
1. email_template.xml
2. bootstrap_model.xml
3. Employee Initial Data.xml

11.4 Drag the imported objects into the Project

Click on the on the Objects tab and change the resource browser to ▶ ent:

Drag the 

onto the **DEMOPROJECT**.

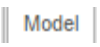
Go back to the 

tab.

Drag the 

onto the **DEMOPROJECT**.
Rename the new object from 'UCFRM_1' to **EMPFORM**.

Right-click on EMPFORM and select Open

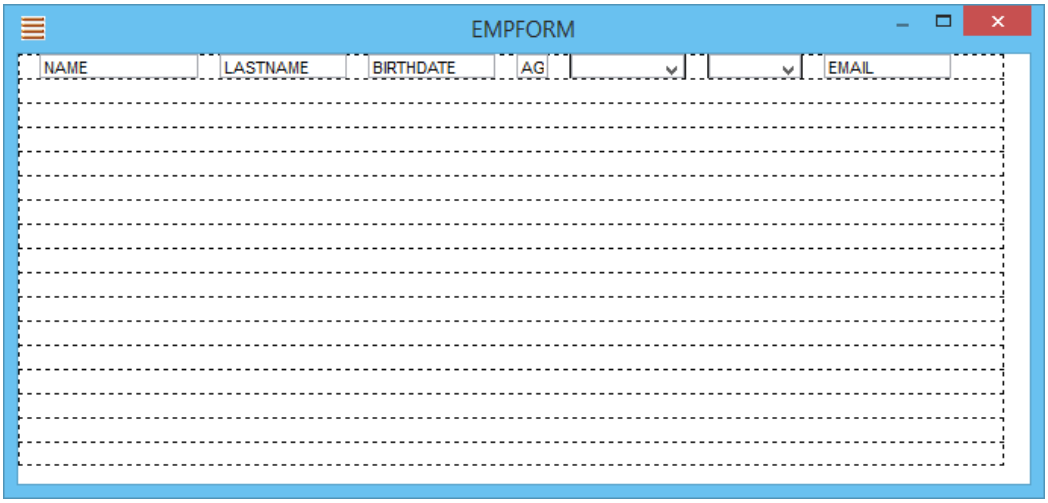
Click on the 

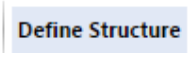
tab and select the **EMPLOYEE** entity.

Click Define Frames

Click in the new form and draw the entity frame.

Click each field and paint on the canvas like this:



Go to the 

tab and change the FRM Widget Type to '**EGRID**'

Click on the 

tab.

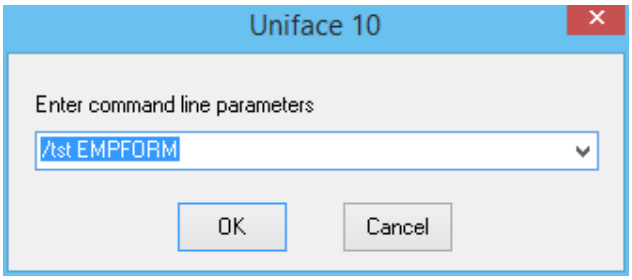
Add a 'retrieve' statement before the 'edit' statement like this:

```
operation exec

  ; Your pre-edit code here...
  retrieve
  ; Shows the form component and allow data entry (script execution stops here and continues after a close request)
  edit

  ; At this point, $status is:
  ;   9 when the trigger accept performend a close request
  ;   10 when the trigger quit performed a close reques

  ; Your post-edit code here...

  ; Return execution status:
  ;   return 0    ; return >= 0 typically indicates a succesful execution
  ;   return -1   ; return < 0 typically indicates an error during execution
  return 0

end
```

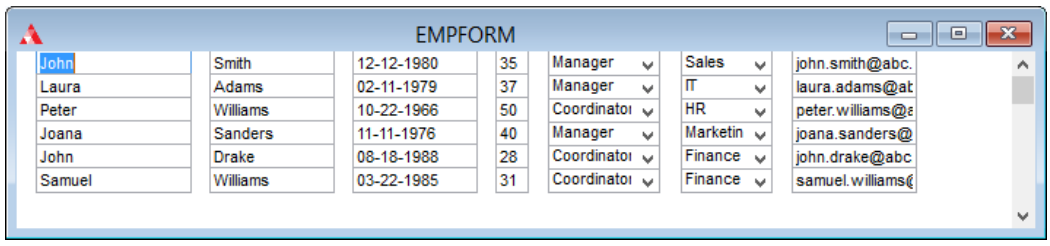
From the right-hand side of the screen click 'Compile'.

Start another instance of the IDE and enter the following into the command line:

/tst EMPFORM

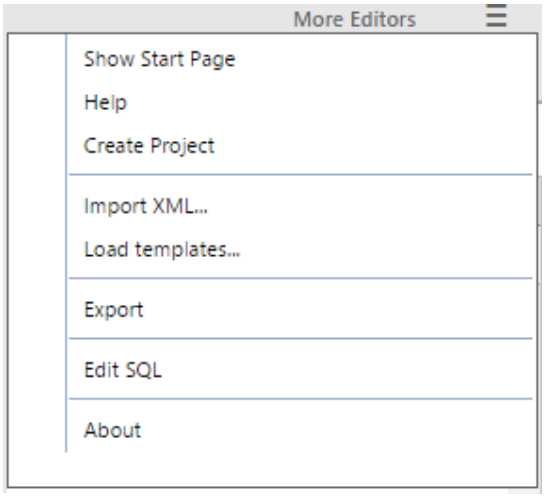


You should see this:



11.5 Adding Highlighting

Import the include proc



Select 'Import XML…' and select 'Code Samples\HILIGHTROW Include Proc.xml'

IF YOU IMPORTED THE EMPLOYEE.BOOTSTRAP MODEL THE READ TRIGGER WILL ALREADY HAVE THE FOLLOWING CODE IN IT AND YOU WILL NOT NEED TO MODIFY THE READ TRIGGER:

```
trigger read
read
if ($status >= 0)
 age = $int( ( $date - birthdate ) / 365)
#ifdefined GRIDENTITY
  if ($curocc % 2 )
    call OccurrenceSetFieldColors( "<GRIDENTITY>","<SELECTIONCOLOR>")
  endif
#endif
endif
end
```

If you have NOT imported the EMPLOYEE.BOOTSTRAP model then you will need to do the following:

Open the 'STANDARD_SNIPPETS'

Paste the <> Row Hilighting into the read trigger.

Paste <> Entity Defines for Row Hilighting into the Declarations section of the form. Change the GRIDENTITY to EMPLOYEE.BOOTSTRAP.

Then insert the following into the Declarations Section in the component.

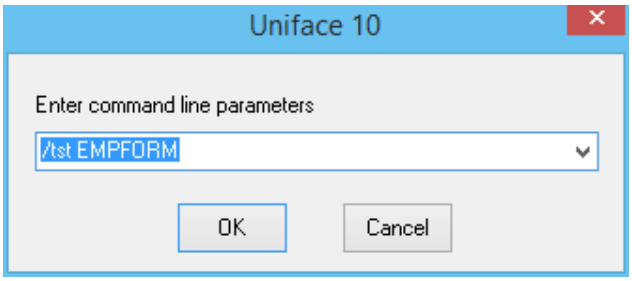
```
variables
    string sortorder
endvariables
; Your component VARIABLES block here (optional)...
#include SYSTEM_LIBRARY:HILIGHTROW
```

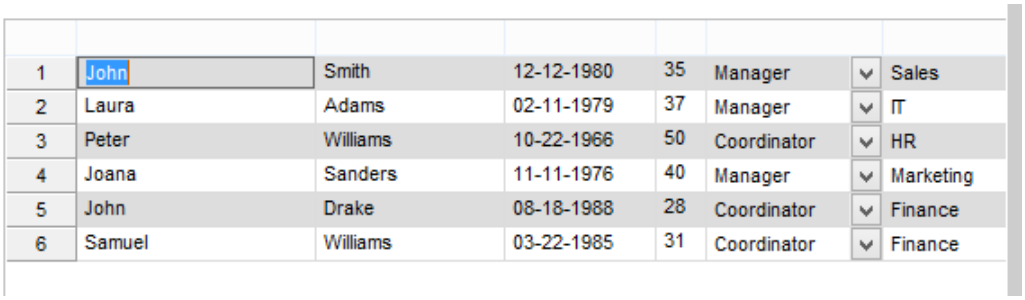
Everything should look like this:

```
Declarations
; Your component defines here (optional)...
#define SELECTIONCOLOR = #DDDDDD
#define GRIDENTITY = EMPLOYEE.BOOTSTRAP

variables
    string sortorder
endvariables
; Your component VARIABLES block here (optional)...
#include SYSTEM_LIBRARY:HILIGHTROW
```

From the right-hand side of the screen click 'Compile'. Start another instance of the IDE and enter the following into the command line:

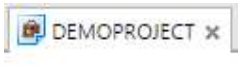
/tst EMPFORM





1	John	Smith	12-12-1980	35	Manager	Sales
2	Laura	Adams	02-11-1979	37	Manager	IT
3	Peter	Williams	10-22-1966	50	Coordinator	HR
4	Joana	Sanders	11-11-1976	40	Manager	Marketing
5	John	Drake	08-18-1988	28	Coordinator	Finance
6	Samuel	Williams	03-22-1985	31	Coordinator	Finance

Close and Exit.

Click the 

tab to navigate back to our Project.

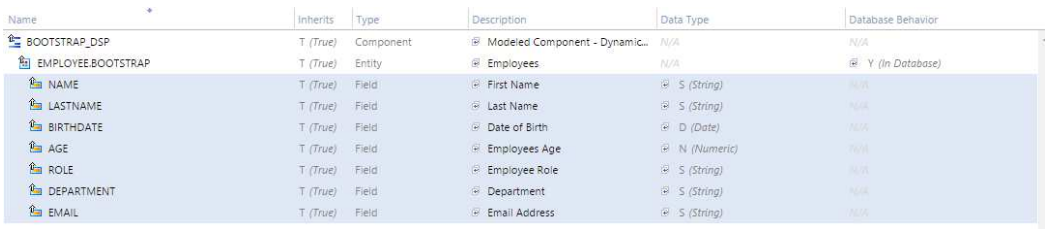
12. Using the BootStrap DSP Component

Select the Model tab and drag the BOOTSTRAP_BASE onto the DEMOPROJECT.

Click on the BOOTSTRAP_BASE_1 component and rename it to 'BOOTSTRAP_DSP'. Right click and open the BOOTSTRAP_DSP component.

Select the Model tab and drag the EMPLOYEE.BOOTSTRAP entity onto the DSP.

Multi-Select and drag all of the EMPLOYEE fields except the ID field.

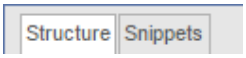


12.1 Copy in the layout

Select libsnp:STANDARD_SNIPPETS

Select 'Bootstrap Base HTML' and then right-click and select 'Insert into Layout'.

Click on the Structure tab.



Select '**EMPLOYEE.BOOTSTRAP**' and all of its fields. Right-click and Select 'As horizontal table'

In the Design Layout right-click and Paste after the table comment.

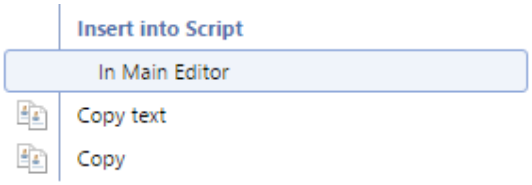
Add the **class="table table-striped"** to the Table definition right before the border="1"

Click on the Define Structure | Define Frames | **Write Script** tab.

Select libsnp:USCRIPT_DSP.

Position your cursor into the Script code fold.

Click on <> [cpt] operation exec and right-click and select Insert into Script -> In Main Editor.

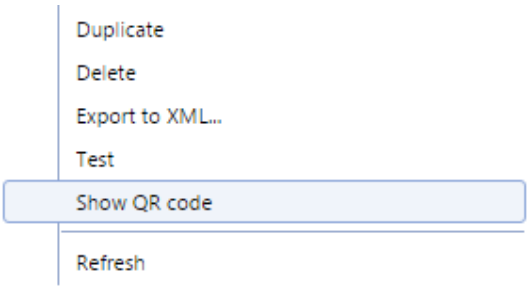


Insert a '**retrieve**' statement into the operation like this:

```
operation exec
public web
; Your SCOPE  block here (optional)...
; Your VARIABLES block here (optional)...

  retrieve

  ; Return execution status:
  ;   return 0    ; return >= 0 typically indicates a succesful execution
  ;   return -1   ; return < 0 typically indicates an error during execution
  return 0

end
```

Compile

From the Actions Menu



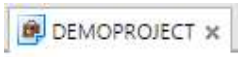
And click on the QR Code to bring up the page in a browser. For example:

12.2 Change the Page Title

Change <h2>My Page Title</h2> to <H2>Employees</H2>

Employees

John	Smith	1980-12-12	35	Manager	Sales	john.smith@abc.com
Laura	Adams	1979-02-11	37	Manager	IT	laura.adams@abc.com
Peter	Williams	1966-10-22	50	Coordinator	HR	peter.williams@abc.com
Joana	Sanders	1976-11-11	40	Manager	Marketing	joana.sanders@abc.com
John	Drake	1988-08-18	28	Coordinator	Finance	john.drake@abc.com
Samuel	Williams	1985-03-22	31	Coordinator	Finance	samuel.williams@abc.com

Click the 

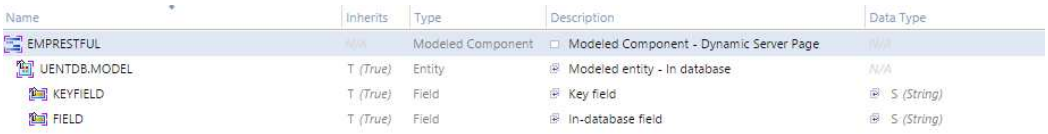
tab to navigate back to our Project.

13. Using the RESTful Template

Select the Model tab and drag the RESTFUl_BASE onto the DEMOPROJECT.

Rename the RESTFUL_BASE_1 to EMPRESTFUL.

Right click and open the EMPRESTFUL component.



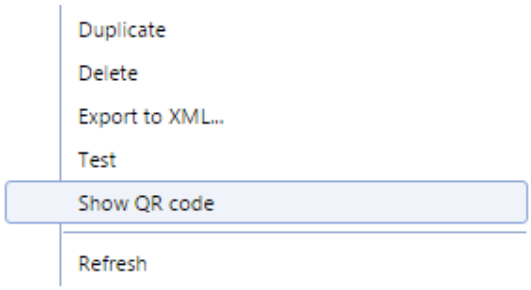
Rename UENTDB.MODEL to EMPLOYEE.BOOTSTRAP

Rename KEYFIELD to ID

Rename FIELD to NAME.

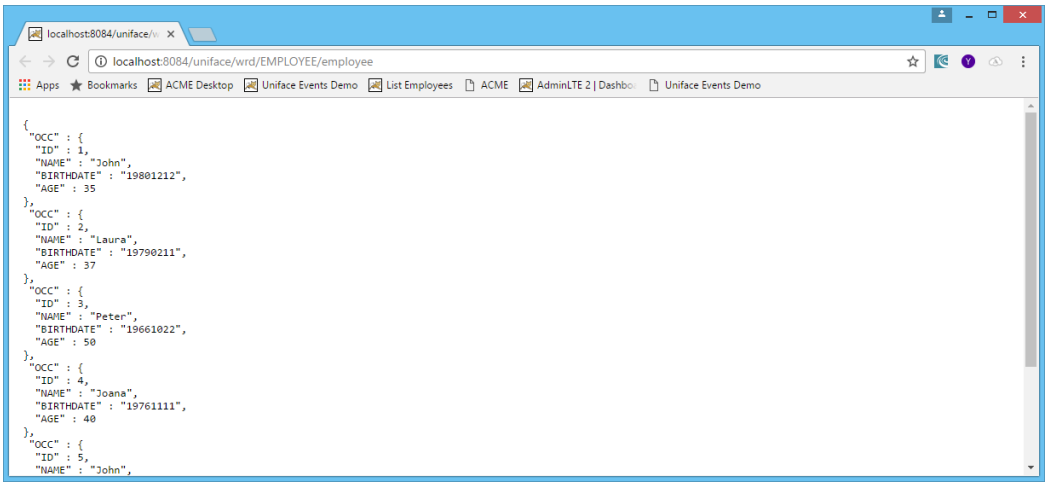
Compile

From the Actions Menu

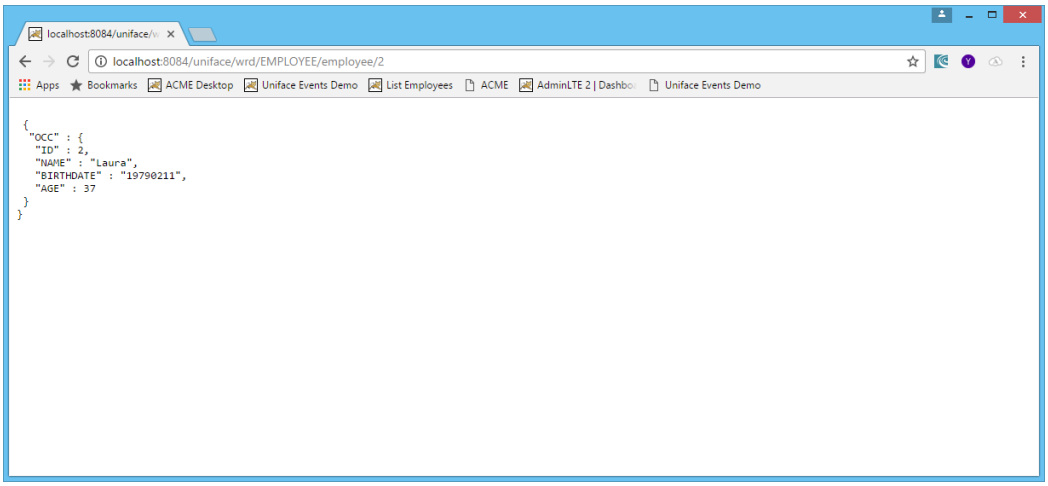


And click on the QR Code to bring up the page in a browser. For example:

Make sure that you add '/employee' to the test page as the code looks for the entity name on the URL.



Try it with /employee/2



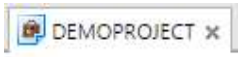
Try it with /employee?format=xml

13.1 Logfile entries for RESTful

```
EMPLOYEE - URL = 'http://localhost:8084/uniface/wrd/EMPLOYEE/employee'
EMPLOYEE - Bound to = 'EMPLOYEE.BOOTSTRAP'
EMPLOYEE - output_type = 'JSON'
EMPLOYEE - type = 'employee'
EMPLOYEE - qualifier = ''
EMPLOYEE - item = ''
EMPLOYEE - subitem = ''
EMPLOYEE - URL = 'http://localhost:8084/uniface/wrd/EMPLOYEE/employee'
EMPLOYEE - Bound to = 'EMPLOYEE.BOOTSTRAP'
EMPLOYEE - output_type = 'JSON'
EMPLOYEE - type = 'employee'
EMPLOYEE - qualifier = ''
EMPLOYEE - item = ''
```

```
EMPLOYEE - subitem = ''
EMPLOYEE - runthis
EMPLOYEE - totalOcc = '6'
EMPLOYEE - Setting HTTPResponseHeaders
EMPLOYEE - Checking ContentType
EMPLOYEE - content-type = application/json
EMPLOYEE - ComponentToStruct $status = 0
EMPLOYEE - JSONRecords = '
 {
  "OCC" : {
   "ID" : 1,
   "NAME" : "John",
   "BIRTHDATE" : "19801212",
   "AGE" : 35
 },
  "OCC" : {
   "ID" : 2,
   "NAME" : "Laura",
   "BIRTHDATE" : "19790211",
   "AGE" : 37
 },
  "OCC" : {
   "ID" : 3,
   "NAME" : "Peter",
   "BIRTHDATE" : "19661022",
   "AGE" : 50
 },
  "OCC" : {
   "ID" : 4,
   "NAME" : "Joana",
   "BIRTHDATE" : "19761111",
   "AGE" : 40
 },
  "OCC" : {
   "ID" : 5,
   "NAME" : "John",
   "BIRTHDATE" : "19880818",
   "AGE" : 28
 },
  "OCC" : {
   "ID" : 6,
   "NAME" : "Samuel",
   "BIRTHDATE" : "19850322",
   "AGE" : 31
  }
}'
EMPLOYEE - URL = 'http://localhost:8084/uniface/wrd/EMPLOYEE/employee/2'
EMPLOYEE - Bound to = 'EMPLOYEE.BOOTSTRAP'
EMPLOYEE - output_type = 'JSON'
EMPLOYEE - type = 'employee'
EMPLOYEE - qualifier = '2'
EMPLOYEE - item = ''
EMPLOYEE - subitem = ''
EMPLOYEE - runthis
EMPLOYEE - totalOcc = '1'
EMPLOYEE - Setting HTTPResponseHeaders
EMPLOYEE - Checking ContentType
EMPLOYEE - content-type = application/json
EMPLOYEE - ComponentToStruct $status = 0
EMPLOYEE - JSONRecords = '
```

```
 {
  "OCC" : {
   "ID" : 2,
   "NAME" : "Laura",
   "BIRTHDATE" : "19790211",
   "AGE" : 37
  }
 }'
```

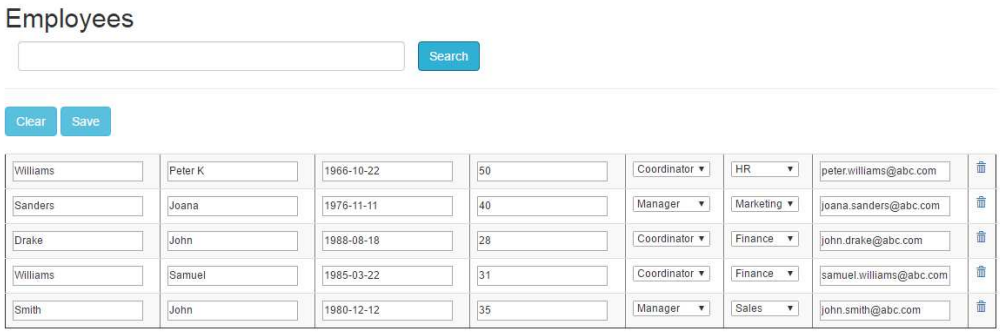
Click the 

tab to navigate back to our Project.

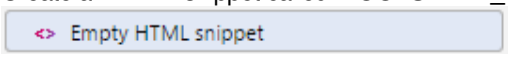
14. Putting it all together - Building the Employee List

This final exercise applies all of the concepts that have previously been accomplished. We will create snippets, modeled components, adding fields and entering procscript.

The objective is to create a modelled component that produces the following page:



Navigate to the STANDARDS project and open STANDARD_SNIPPETS.

Create an HTML snippet called '**BOOTSTRAP_LIST**' by dragging the 

onto the **STANDARD_SNIPPETS** library. In the 

for BOOTSTRAP_LIST enter the following:

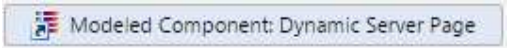
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="Default" content="Uniface 9">
  <link rel="stylesheet" type="text/css"  href="../css/udsp.css">
  <link  href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"  rel="stylesheet"  integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RY
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"  integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8=" crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"  integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD
  <title>This is a HTML title block</title>
</head>
<body>
<div class="container">
 <h2>Employees</h2>
<div class="form-group col-xs-5">
    <input id="ufld:SEARCHTEXT.DUMMY.NOMODEL" class="form-control"  value="Input Element">
</div>
 <span id="ufld:SEARCH.DUMMY.NOMODEL">Static Text</span>
 <br>
 <hr><span id="ufld:CLEAR.DUMMY.NOMODEL">Static Text</span>
 <span id="ufld:SAVE.DUMMY.NOMODEL">Static Text</span>
 <br>
 <br>
<!--
        <table id="uent:EMPLOYEE.BOOTSTRAP" class="table table-striped" border="1">
    -->

</div>
</body>
</html>
```

The above code is available in 'Code Samples\Bootstrap List HTML Layout.txt'

Make sure to add a description and alternate name for the snippet.

Click the 

tab to navigate back to our Project.

Drag the 

from the left-hand side of the screen onto the STANDARDS project in the center pane.

Rename MDSP_1 to "**BOOTSTRAP_LIST**".

Select the Alternative Name field in the Properties pane and change it to "**Bootstrap List**". Change the Description to "**Bootstrap list implementation**".

Right-click on the BOOTSTRAP_LIST component and click 'Open'.

14.1 Add the Layout

Insert the Bootstrap List layout from the previous step into the Layout of the BOOTSTRAP_LIST component. You should also change the <H2>Data Title</H2> to something more useful such as <H2>Employees</H2>.

14.2 Establish Controls

Select Model tab and drag UENT.MODEL onto the BOOTSTRAP_LIST structure.

Rename the UENT.MODEL to DUMMY.NOMODEL.

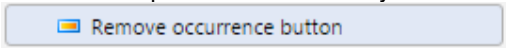
From the Templates tab drag and insert under the DUMMY.NOMODEL the following:

Field Name	Datatype	Size	Widget	Widget Properties	Initial Value
SEARCHTEXT	String	C128	Editbox		
SEARCH	String	C10	Flatbutton	html: btn btn-info	Search
CLEAR	String	C10	Flatbutton	html: btn btn-info	Clear
SAVE	String	C10	Flatbutton	html: btn btn-info	Save

Select Model tab and drag UENTDB.MODEL onto the BOOTSTRAP_LIST structure.

Drag the 'FIELD' under UENTDB.MODEL and rename it to 'NAMEFIELD'.

From the Templates tab under 'Entity control buttons' drag '

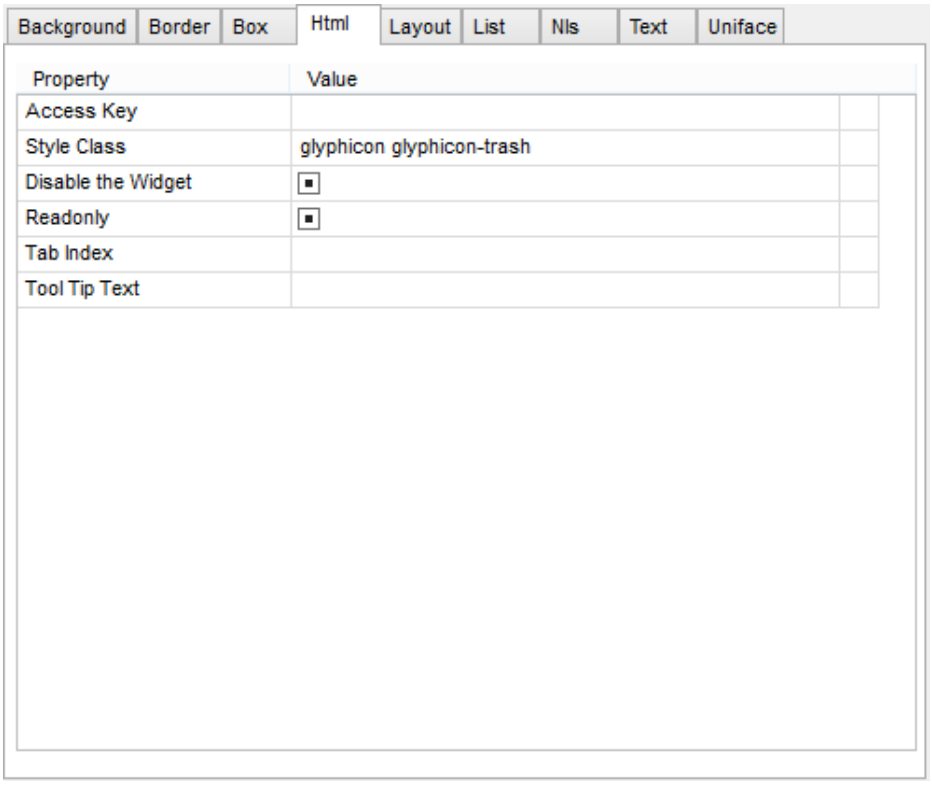


'under the NAMEFIELD created above.

Clear the Initial Value of 'Delete' to empty.

Set the DSP Widget Type to 'flatbutton'

Set the DSP Widget Properties to:



Set the 'Is Generic' property to TRUE for UENTDB.MODEL and NAMEFIELD.UENTDB.

Add the following code to the BTN_REMOCC:

```
trigger detail

public web
scope
  input
  output
endscope

  remocc "<$entname>", 0
  store/e "<$entname>"
  commit

end
```

SEARCH

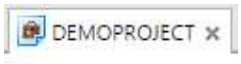
```
Script
trigger detail
public web
; Your SCOPE  block here (optional)...
; Your VARIABLES block here (optional)...

  ; Your implementation here...

clear/e "<UENTDB>"
if ($length(SEARCHTEXT.DUMMY) > 0)
    <NAMEFIELD.<UENTDB>>/init = "%%searchtext"
endif
retrieve/e "<UENTDB>"
putmess "Returned %%$hits(<UENTDB>) records from '<UENTDB>' with profile of '%%searchtext'"
return (0)

end
```

CLEAR

```
Script
trigger detail
public web
; Your SCOPE  block here (optional)...
; Your VARIABLES block here (optional)...

  ; Your implementation here...
clear/e "<UENTDB>"
return (0)
end
```

SAVE

```
Script
trigger detail
; Remove the current occurrence
public web
; Mark the occurence as deleted
putmess "Attempting to store data for '<UENTDB>'"

store/e "<UENTDB>"
if ($status = 1)
    webmessage/info "No data has been changed. Nothing to save."
    return (0)
endif

putmess "Saving '%%$entname' status was '%%$status'"
commit
if ($status = 0)
    webmessage "Data Saved"
endif
return (0)
```

Click the 

tab to navigate back to our Project.

14.3 Implement the BOOTSTRAP_LIST

Drag a BOOTSTRAP_LIST onto your DEMOPROJECT from the Model tab.

Rename it 'EMPLIST'.

Rename UENTDB.MODEL to EMPLOYEE.BOOTSTRAP.

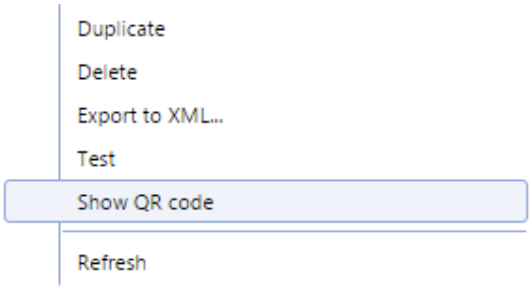
Rename NAMEFIELD to LASTNAME.

From the Model tab drag the remaining EMPLOYEE fields and insert them between the LASTNAME and BTN_REMOCC.

Name	Inherits	Type	Description	Data Type	Database Behavior
EMPLIST	T (True)	Component	Modeled Component - Dynam...	N/A	N/A
DUMMY.NOMODEL	F (False)	Entity		N/A	(Not in Database)
SEARCHTEXT	F (False)	Field	Editbox (Single-line String)	S (String)	N/A
SEARCH	F (False)	Field	Flat Button	S (String)	N/A
CLEAR	F (False)	Field	Flat Button	S (String)	N/A
SAVE	F (False)	Field	Flat Button	S (String)	N/A
EMPLOYEE.BOOTSTRAP	T (True)	Entity	Employees	N/A	Y (In Database)
LASTNAME	T (True)	Field	Last Name	S (String)	N/A
NAME	T (True)	Field	First Name	S (String)	N/A
BIRTHDATE	T (True)	Field	Date of Birth	D (Date)	N/A
AGE	T (True)	Field	Employees Age	N (Numeric)	N/A
ROLE	T (True)	Field	Employee Role	S (String)	N/A
DEPARTMENT	T (True)	Field	Department	S (String)	N/A
EMAIL	T (True)	Field	Email Address	S (String)	N/A
BTN_REMOCC	F (False)	Field	Remocc Button	S (String)	N/A

Compile.

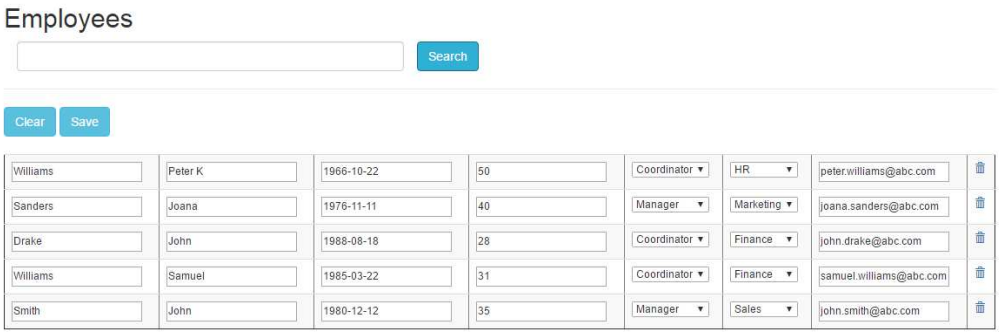
Actions -> Show QR Code



And Click on the QR Code to bring up the page.

Result:

Click the Search button:


Employees
Search
Clear
Save
Williams
Peter K
1966-10-22
50
Coordinator
HR
peter.williams@abc.com
Sanders
Joana
1976-11-11
40
Manager
Marketing
joana.sanders@abc.com
Drake
John
1988-08-18
28
Coordinator
Finance
john.drake@abc.com
Williams
Samuel
1985-03-22
31
Coordinator
Finance
samuel.williams@abc.com
Smith
John
1980-12-12
35
Manager
Sales
john.smith@abc.com