

Template Directives



FINALLY BACK TO OUGN!

SET THE DATE: 24-25 APR 2023

Don't miss out on what the future holds: A two-day conference at Rebel Oslo,

Lino Schildenfeld

@LinoSchilde
lschilde.blogspot.com

APEX Office hours

April 20, 2023 14:00 - 15:00 UTC
Start Times Around the World

Subscribe

 Actions

Part 1: What's new in Oracle APEX 23.1

In this session, we will present and demonstrate the marquee features of APEX 23.1.

Catch up on the recording for our **last** session!

Don't miss our **next** live session!

This session: <https://asktom.oracle.com/pls/apex/asktom.search?oh=20407>

apex.oracle.com/officehours

The banner features a dark blue background with a subtle grid of glowing lines on the right side. A red asterisk is positioned above the version number. The text 'Oracle Database 23c Free Developer Release' is in a large, white, sans-serif font. Below it, the tagline 'Easy, early access to application development features.' is in a smaller, white, sans-serif font. A large white greater-than sign is located at the bottom right.

Oracle Database 23^c Free Developer Release

Easy, early access to application development features.

23^c

>

Oracle Database 23c Free Version Now Available to Developers



Mentor and Speaker Hub

- Our goal is to *connect* speakers with mentors to assist in *preparing* technical sessions and *improving* presentation skills

Interested? Read more and get in touch

<https://mashprogram.wordpress.com>

AUSOUG APEX News



<https://www.ausoug.org.au/news/apex-special-series-2023/>

My story



AUSOUG



FOEX



APEX RD
research & development

- AUSOUG APEX webinars
- NZ APEX meetup organizer
- APEX World Member of the Month x2
- Conference speaker
- Blogger

@LinoSchilde



Oracle ACE
Pro






APEX & TEMPLATES

Sample Application

Welcome: DEMO [Print](#) [Feedback](#) [Logout](#)

[Home](#) [Customers](#) **[Products](#)** [Orders](#) [Reports](#)

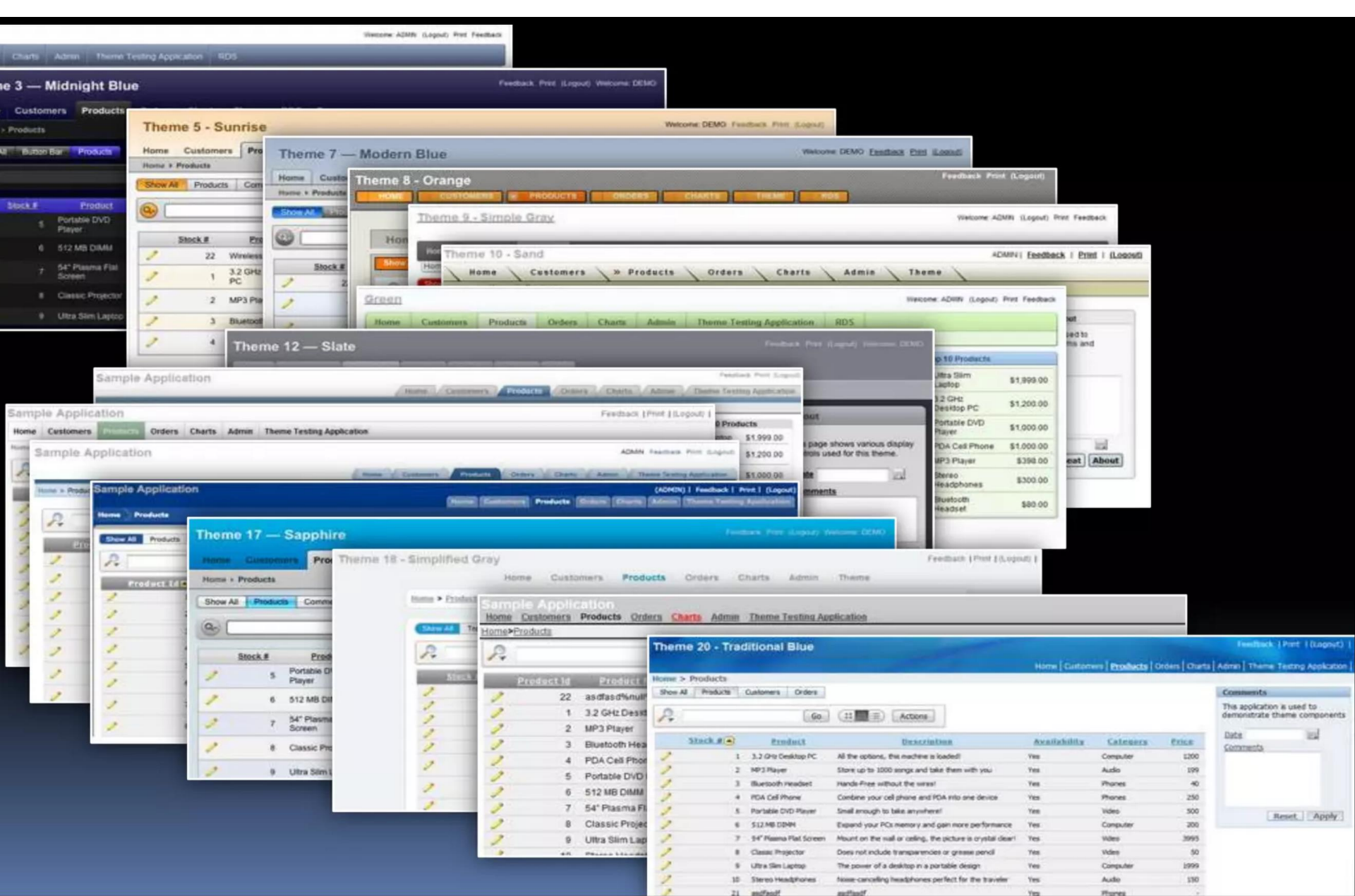
[Home](#) > [Products](#)

Name	Category	Available	Price	Units	Sales	Customers	Last Sale	Image
 Bag	Accessories	Yes	\$125.00	16	\$2,000.00	6	7/13/2010	
 Belt	Accessories	Yes	\$30.00	11	\$330.00	3	7/13/2010	
 Blouse	Womens	Yes	\$60.00	16	\$960.00	5	7/13/2010	
 Business Shirt	Mens	Yes	\$50.00	23	\$1,150.00	5	7/16/2010	

Top 5 Products

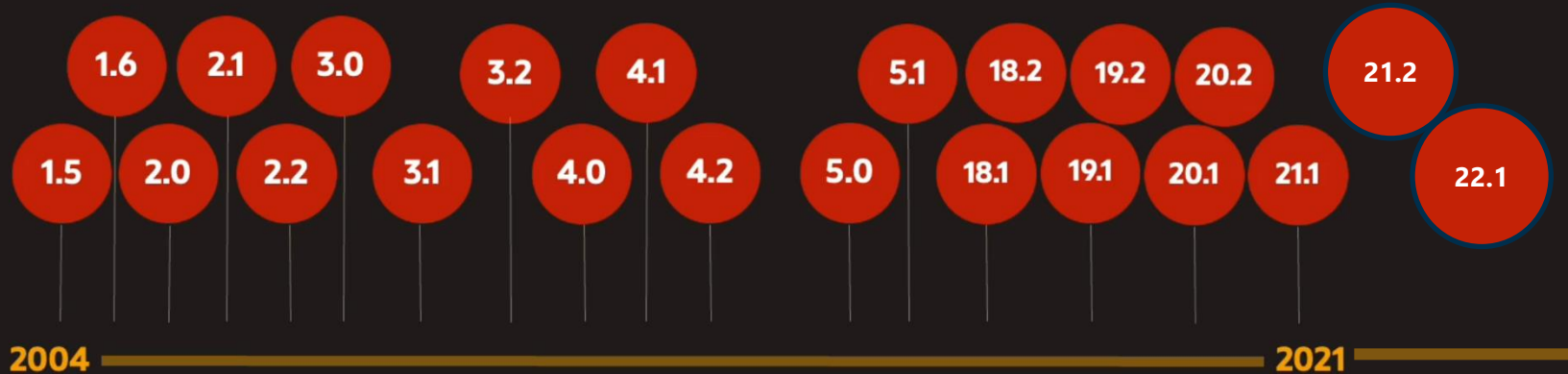
Jacket	\$2,700.00
Bag	\$2,000.00
Trousers	\$1,680.00
Ladies Shoes	\$1,440.00
Business Shirt	\$1,150.00

<https://www.slideshare.net/InSyncConference/apex-themes-and-templates>



APEX versions

APEX Product Timeline



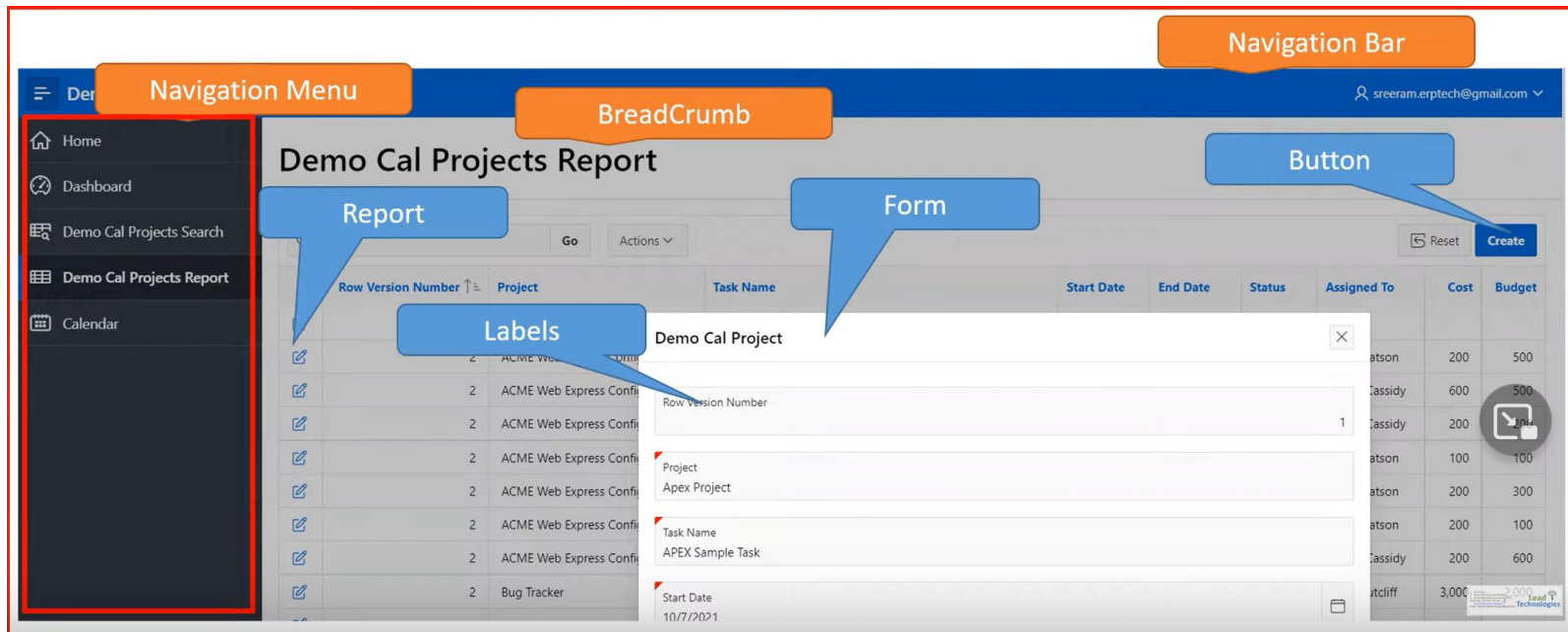
20
Releases

250+
Major features

0
required rewrites

Basics

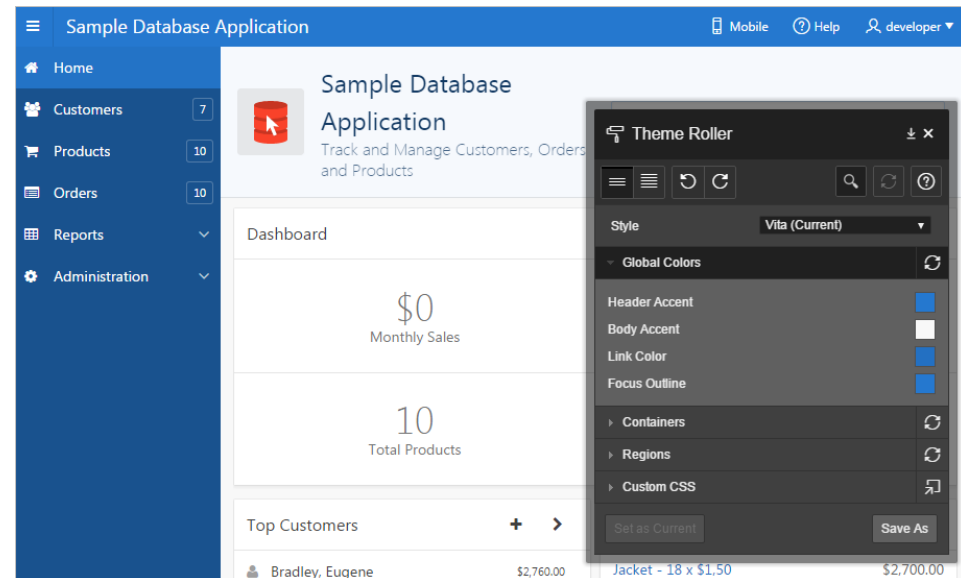
- Theme -> Template -> Template Classes
- Templates dictate how page and page component display



https://www.youtube.com/watch?v=FaLuG_NHVqc

Anatomy of a theme

- Default settings
- Subscription and theme refresh
- Global template options and groups
- Third party icon libraries
- Path to files
- Theme styles
- Templates



- Show All
- Name
- Theme Subscription
- JavaScript and Cascading Style Sheets
- Component Defaults
- Region Defaults
- Dialog Defaults
- Template O

Theme Subscription

Verify

Unsubscribe

Refresh Theme

Subscribed to Standard Theme : **Universal Theme**

No themes subscribe to this theme.

JavaScript and Cascading Style Sheets

JavaScript File URLs	#APEX_FILES#libraries/apex/#MIN_DIRECTORY#widget.stickyWidget#MIN#.js?v=#APEX_VERSION# #THEME_FILES#js/theme42#MIN#.js?v=#APEX_VERSION#	?
CSS File URLs	#THEME_FILES#css/Core#MIN#.css?v=#APEX_VERSION#	?

Component Defaults

Specify component defaults by component type.

Page	Standard	?
Navigation Bar List	Navigation Bar	?
Navigation Menu List Position	Side	?
Navigation Menu List (Top)	Top Navigation Menu	?
Navigation Menu List (Side)	Side Navigation Menu	?
Login Page	Login	?
Error Page	Login	?
Printer Friendly Page	Standard	?

Q

Go

Actions

Reset

Create

Type	Name	References	Updated	Updated By	Default	Theme	Copy
Breadcrumb	Breadcrumb	3	2 days ago	Ischilde@gmail.com	✓	42	
Button	Icon	1	2 days ago	Ischilde@gmail.com		42	
Button	Text	13	2 days ago	Ischilde@gmail.com	✓	42	
Button	Text with Icon	3	2 days ago	Ischilde@gmail.com		42	
Label	Hidden	3	2 days ago	Ischilde@gmail.com		42	
Label	Optional	1	2 days ago	Ischilde@gmail.com		42	
Label	Optional - Above	0	2 days ago	Ischilde@gmail.com		42	
Label	Optional - Floating	4	2 days ago	Ischilde@gmail.com	✓	42	

Column Heading Template

1<th class="t-Report-colHead" #ARIA_SORT# #ALIGNMENT# id="#COLUMN_HEADER_NAME#" #COLUMN_WIDTH#>#COLUMN_HEADER#</th>

Before Each Row

Column Templates

Column Template 1

1<td class="t-Report-cell" #ALIGNMENT# #ACCESSIBLE_HEADERS#>#COLUMN_VALUE#</td>

Basics

- APEX supports templates, substitution strings and template options
 - #ITEM# vs &ITEM. vs &ITEM!HTML.
 - APEX Template type is processed by code specific to the task
 - Static with limitations
 - Often with duplication between APEX Templates (Page Templates)
 - little or no logic => can't change the data

Examples

Change the background and text colors

```
<span style="background-color:yellow; color:red;">&EMPLOYEE_STATUS.</span>
```

Use Heading 2

```
<h2>&EMPLOYEE_NAME.</h2>
```

Use a dynamic class from the SQL query

```
<span class="&COLUMN_CLASS!ATTR.">&EMPLOYEE_STATUS.</span>
```

Alignment

Alternative Label

Settings

HTML Expression

&NAME.

Dynamic templates

- How to make templates dynamic????
 - use JS frameworks like in Mustache, Handlebars or JavaServer Pages
- HTML Expression with "Show Null Values as" property does not work
- Situations like multiple values, null vs not null we handled with SQL

```
1 <!DOCTYPE html lang="en">
2 <head>
3   <script src="https://code.jquery.com/jquery-3.2.1.min.js"
4     integrity="sha256-hwg4gsxgFZh0sEEamdOYGBf13FyQuiTwIAQgXVNgt4="
5     crossorigin="anonymous"></script>
6   <script type="text/javascript" src="js/mustache.min.js"></script>
7   <script type="text/javascript" src="js/js.js"></script>
8   <link rel="stylesheet" type="text/css" href="css/css.css" />
9 </head>
10
11 <body>
12   <script id="template" type="x-templ-mustache">
13
14     {{#people}}
15     <ul id={{id}} class="character">
16       <li></li>
17       <li>{{name.firstname}} {{name.lastname}}</li>
18       <li>{{address}}</li>
19       <li>{{occupation}}</li>
20       <li>{{hometown}}</li>
21       <li>{{name.firstname}} has a friend, {{friend}}</li>
22     </ul>
23   </people>
24
25   </script>
26   <div id="char-info"></div>
27 </body>
28 </html>
```



Common way - SQL

Code Editor - SQL Query

```
1  select PRODUCT_ID,
2     PRODUCT_NAME,
3     ...
4     case when PRODUCT_STATUS != 'orderable' then
5         '<span class="fa fa-sm fa-exclamation-triangle-o u-color-7-text" aria-hidden="true"></span>Out of Stock!'
6     else '<span class="fa fa-cart-plus u-color-5-text" aria-hidden="true"></span> Order now '
7 end OUT_OF_STOCK,
8 '<ul class="a-tags">'
9     || case
10         when PRODUCT_DESCRIPTION is not null then
11             '<li class="a-tag">'
12             || apex_string.join(apex_string.split(rtrim(PRODUCT_DESCRIPTION, '.'), '.'), '</li><li>')
13             || '</li>'
14         else null
15     end
16     || '</ul>' DESC_LIST
17
18 from OEHR_PRODUCT_INFORMATION
```

Escape special
characters



Cancel

OK

Common way - SQL



Save and Run Page



```
1  select
2      ...
3      PRODUCT_DESCRIPTION,
4      case when product_avail != 'Y' then
5          '<span class="fa fa-sm fa-exclamation-triangle-o" aria-hidden="true"></span>Out of Stock!'
6      else null end STOCK_WARN,
7      '<ul class="a-tags">' || case when tags is not null then
8          '<li class="a-tag">' ||
9          apex_string.join(apex_string.split(tags, ','), '</li><li>') ||
10         '</li>'
11     else null end || '</ul>' TAGS_LIST,
12     ...
13 from DEMO_PRODUCT_INFO
```

```
1  &PRODUCT_DESCRIPTION.
2  &STOCK_WARN!RAW.
3  &TAGS_LIST!RAW.
```

Processors

- A template processor takes an input and combines them to produce output text(s)
- Template processors usually support features:
 - Substituting a placeholder for a data value
 - Conditionals to output one part of the template or another depending on data values
 - Looping over some kind of data sequence, list, array, or collection
 - Inclusion of one template inside another

Template manipulation

- Special **tokens** in a template that control how the text around it is processed
- Processing as part of client side substitutions
- Allow us to more easily control the display and formatting of data returned by SQL

We've seen it before?!!

Angular expressions – AOP

Product	Quantity	Unit Price	Extended Total	Info
{#product}{product_name}{^onstock} ¹ {/onstock}	{quantity}	{unit_price}	{#cur=='USD'}\$/{cur=='USD'}{quantity * unit_price} {#cur=='EUR'}€/{cur=='EUR'}	{#unit_price>70}High Price Item {/unit_price>70} {^unit_price>70}Low Price Item{/unit_price>70} {/product}
⁽¹⁾ This product is currently not in stock, your order may be delayed.				

will result in:

Product	Quantity	Unit Price	Extended Total	Info
Business Shirt	3	50	150€	Low Price Item
Trousers ⁽¹⁾	3	80	\$240	High Price Item
Jacket	3	15	\$45	Low Price Item
Blouse ⁽¹⁾	3	60	180€	Low Price Item
⁽¹⁾ This product is currently not in stock, your order may be delayed.				

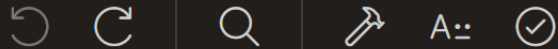
APEX Template Directives

- In APEX 21.1 - supported by specific attributes of the cards, interactive grid and email templates
- In APEX 22.2 support for Classic and Interactive Reports
- Any APEX components that support client-side rendering
- Use template directives to control how attributes that support substitution strings are processed

Additional Information

- Type: HTML
- Supported Substitutions: Application, Page Items, Region Columns and System Variables
- Supports Template Directives

New way



```
1  select PRODUCT_ID,  
2     PRODUCT_NAME,  
3     PRODUCT_DESCRIPTION,  
4     CATEGORY_ID,  
5     WEIGHT_CLASS,  
6     WARRANTY_PERIOD,  
7     SUPPLIER_ID,  
8     PRODUCT_STATUS,  
9     LIST_PRICE,  
10    MIN_PRICE,  
11    CATALOG_URL  
12  from OEHR_PRODUCT_INFORMATION
```

```
1  <ul class="a-tags">  
2    {loop "." PRODUCT_DESCRIPTION/  
3      <li class="a-tag">&APEX$ITEM.</li>  
4    {endloop/  
5  </ul>
```

```
1  {{case PRODUCT_STATUS/  
2    {when orderable/  
3      <span class="fa fa-cart-plus u-color-5-text" aria-hidden="true"></span> Order now  
4    {otherwise/  
5      <span class="fa fa-sm fa-exclamation-triangle-o u-color-7-text" aria-hidden="true"></span>Out of Stock!  
6  {endcase/}}
```

Old vs New way

Code Editor - SQL Query

```
1 select PRODUCT_ID,  
2 PRODUCT_NAME,  
3 ...  
4 case when PRODUCT_STATUS != 'orderable' then  
5     '<span class="fa fa-sm fa-exclamation-triangle-o u-color-7-text" aria-hidden="true"></span>Out of Stock'  
6 else '<span class="fa fa-cart-plus u-color-5-text" aria-hidden="true"></span> Order now '  
7 end OUT_OF_STOCK,  
8 '<ul class="a-tags">'  
9     || case  
10         when PRODUCT_DESCRIPTION is not null then  
11             '<li class="a-tag">'  
12             || apex_string.join(apex_string.split(rtrim(PRODUCT_DESCRIPTION, '.'), '.'), '</li><li>')  
13             || '</li>'  
14         else null  
15     end  
16     || '</ul>' DESC_LIST  
17  
18 from OEHR_PRODUCT_INFORMATION
```

```
1 <ul class="a-tags">  
2 {loop "." PRODUCT_DESCRIPTION/  
3     <li class="a-tag">&APEX$ITEM.</li>  
4 {endloop/  
5 </ul>
```

```
1 {case PRODUCT_STATUS/  
2     {when orderable/  
3         <span class="fa fa-cart-plus u-color-5-text" aria-hidden="true"></span> Order now  
4     {otherwise/  
5         <span class="fa fa-sm fa-exclamation-triangle-o u-color-7-text" aria-hidden="true"></span>Out of Stock!  
6 {endcase/}
```

Benefits - Why

Simplify SQL

- All UI logic can be put into APEX
- Business (retrieval) logic goes in the SQL

Security

- Reduce the risk of Cross-Site Scripting
 - As there is no need to use 'RAW' item substitution syntax, e.g., &P1_TAGS!RAW.
 - Or Escape special character settings

Improve performance

- by reducing the size of the SQL results

If conditions

```
{if [!]NAME/}  
  TRUE_TEMPLATE_TEXT  
{elseif [!]NAME2/}  
  ELSE_TRUE_TEMPLATE_TEXT  
{else/}  
  FALSE_TEMPLATE_TEXT  
{endif/}
```



```
{if DESCRIPTION/}  
  &DESCRIPTION.  
{else/}  
  No description.  
{endif/}
```


Null checks

```
{if [!]NAME/}  
  TRUE_TEMPLATE_TEXT  
{endif/}
```



```
{if DESCRIPTION/}  
  &DESCRIPTION.  
{else/}No description.  
{endif/}
```

Case statements

```
{case NAME/}  
  {when STRING1/}  
    TEMPLATE_TEXT1  
    .....  
  {otherwise/}  
    TEMPLATE_TEXT  
{endcase/}
```



```
{case ENABLED_FLAG/  
  {when Y/} <span class="u-success-text">Enabled</span>  
  {when N/} <span class="u-color-15-text">Disabled</span>  
  {otherwise/} <span class="u-danger-text">Unknown</span>  
{endcase/}
```

Loop

```
loop ["SEP"] NAME/}  
  TEMPLATE_TEXT  
{endloop/}
```



```
{loop ":" TAGS/}  
  <span class="my-CardView-badge">&APEX$ITEM. </span>  
{endloop/}
```

- The default separator is ":"
- APEX\$ITEM - This is the value of the current item in the list.
- APEX\$I - This is 1 based index of the current item in the list.

Email example

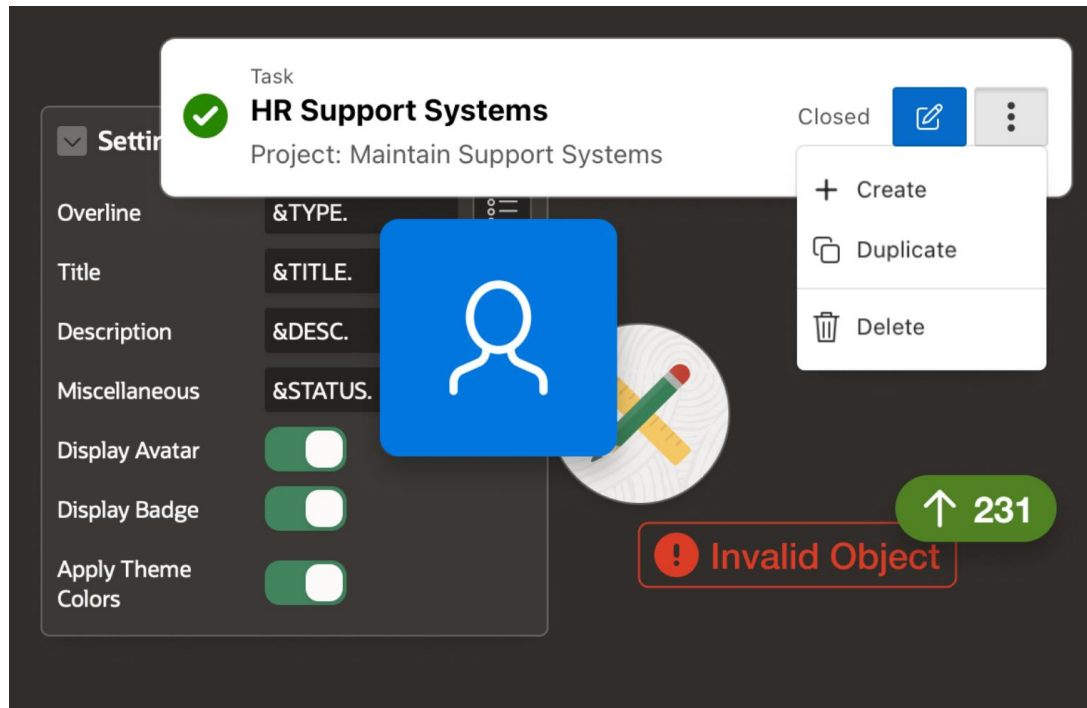
* Body ?

```
1 {case JOB/}
2   {when PRESIDENT/}
3     <p>your salary is SECRET!</p>
4   {when SALESMAN/}
5     <p> salary is <strong>#SAL#</strong> and commission is <strong>#COMM#</strong>.</p>
6   {otherwise/}
7     <p>Salary is <strong>#SAL#</strong>.</p>
8 {endcase/}
9
10
11 <p>Thank you for being a happy chap.</p>
```




Template Components

- a new plug-in type for building re-usable UI components with support for actions, menus, and custom attributes



<https://apex.oracle.com/pls/apex/r/timplating/apex-alpe-adria-2023/home>

template component

<div><div><div>Q</div><div></div></div><div>Go</div><div>Actions</div></div>							
Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Improvment
7839	KING	PRESIDENT		11/17/1981	5000		
7698	BLAKE	MANAGER	7839	5/1/1981	2850		
7782	CLARK	MANAGER	7839	6/9/1981	2450		
<div><div>App 177852</div><div>Page 2</div><div>Session</div><div>Debug</div><div>Quick Edit</div><div>Customize</div><div></div><div></div></div>							

Behnam Malekara


```

Partial
1 {if APEX$IS_LAZY_LOADING/}
2   <!-- Placeholder for when loading -->
3   <div class="metricCard" aria-hidden="true">
4     <div class="metricCard-body">
5       {if ?TITLE%assigned/}<div class="metricCard-title is-placeholder"></div>{endif/}
6       {if ?METRIC%assigned/}<div class="metricCard-metric is-placeholder"></div>{endif/}
7       {if ?META%assigned/}<div class="metricCard-meta is-placeholder"></div>{endif/}
8     </div>
9     <div class="metricCard-details">
10      {if DISPLAY_BADGE/}
11      <div class="metricCard-badge">
12        {with/}
13        LABEL:=
14        VALUE:=
15        SIZE:=#BADGE_SIZE#
16        SHAPE:=#BADGE_SHAPE#
17        {apply THEME$BADGE/}
18      </div>
19      {endif/}
20    </div>
21  </div>
22 {else/}
23 <!-- Metric Card -->
24 {if LINK/}<a href="#LINK!ATTR#" #LINK_ATTR!RAW#{else/}<div{endif/} class="metricCard">
25   <div class="metricCard-body">
26     {if ?TITLE/}<div class="metricCard-title">#TITLE#</div>{endif/}
27     {if ?METRIC/}<div class="metricCard-metric">#METRIC#</div>{endif/}
28     {if ?META/}<div class="metricCard-meta">#META#</div>{endif/}
29   </div>
30   <div class="metricCard-details">
31     {if DISPLAY_BADGE/}

```

JS API - applyTemplate

Functions ■

(static) `applyTemplate(pTemplate, pOptionsopt)` → {string} ■

Directives

```
{<directive-name>[ <directive-arguments>]/}
```

- Directive names are case insensitive
- There can be no whitespace between the open bracket '{' and the directive name
- Directives often come in sets that work together

Directive processing

Directives control the processing of the template and they are processed first

- {<directive-name>[<directive-arguments>]/}

Then **placeholders**

- #<placeholder-name>#

and finally **data substitutions**

- &<item-name>.
- &<item-name>!<escape-filter>.
- &"<quoted-item-name>".
- &"<quoted-item-name>"!<escape-filter>.
- &APP_TEXT\$<message-key>.
- &APP_TEXT\$<message-key>!<escape-filter>.
- &"APP_TEXT\$<message-key>".
- &"APP_TEXT\$<message-key>"!<escape-filter>.

Items

- &P1_NAME%LABEL.
- The properties and the values they return for a page item are:
 - LABEL - The item label.
 - DISPLAY - The display value of the item's current value.
 - CHANGED - 'Y' if the item has been changed and 'N' otherwise.
 - DISABLED - 'Y' if the item is disabled and 'N' otherwise.

Columns

- The properties for a column item are:
 - HEADING - The column heading text.
 - LABEL - If there is no label the heading will be used with markup removed.
 - DISPLAY - The display value of the column value for the current row/record.
 - HEADING_CLASS - Any CSS classes defined for the column heading.
 - COLUMN_CLASS - Any CSS classes defined for the column.
 - REQUIRED - 'Y' if the column is required and 'N' otherwise.

Remember

Server Side

- Classic Report
- Interactive Report
- Data is merged into the template using our directives

Client Side

- Cards
- Interactive Grid
- Data is sent to the client in JSON, interpretation of directives through JavaScript

Q&A

Thank you for attending



Resources

<https://docs.oracle.com/en/database/oracle/apex/22.2/htmldb/using-template-directives.html#GUID-596537AB-1697-4704-9193-102658452399>

<https://docs.oracle.com/en/database/oracle/apex/22.2/aexjs/apex.util.html#.applyTemplate>

<https://hardlikesoftware.com/weblog/2020/11/04/apex-template-directives/>

<https://blog.cloudnueva.com/apex-template-directives>

<https://tm-apex.blogspot.com/2022/10/using-template-directives-in-email.html>

<https://hartenfeller.dev/blog/oracle-apex-template-components-introduction>

<https://github.com/rhjmartens/public-presentations>