

19.11.2025

The Interactive Grid: Blessing or Curse? Let's find out!

Timo Herwix, Senior Consultant

code of change

 **Hyand** by
GOD | MT

Who am I?



Timo Herwix
Senior Consultant



Senior Consultant at Hyand Solutions since 2019

Previously worked as a Data Warehouse Developer

Oracle APEX since 2016

Oracle Databases since 2008

Blog author, conference speaker

Born in 1983, two children and living in Germany



Our key facts



Germany

- Brunswick
- Ratingen
- Hamburg
- Dortmund
- Cologne
- Frankfurt
- Munich
- Berlin

Poland

- Warsaw

Romania

- Cluj-Napoca

Lithuania

- Vilnius
- Kaunas

India

- Pune

850+

Employees

150+

Customers

110+

million €
turnover

What is the Interactive Grid?

The Interactive Grid is a **powerful component** in Oracle APEX that combines the features of Interactive Reports and Tabular Forms. It allows users to view, edit, and manipulate data in a single, dynamic interface. It was introduced with APEX 5.1 to provide a modern, spreadsheet-style user interface.

What is the Interactive Grid?

Similar to Interactive Report, users can customize the display:

- **Filtering and sorting:** Apply multiple filters and sort across different columns.
- **Aggregations:** Easily calculate sums, averages, counts, and more.
- **Control breaks:** Group your data by column values for better organization.
- **Computations:** Create new columns using formulas.
- **Customizations:** Hide or show columns, rearrange them with drag & drop, and freeze columns.
- **Saving reports:** Users can save their views as either private or public reports.
- **Export:** Export your data to CSV, Excel, and other formats.

What is the Interactive Grid?

Editable interactive grids let users easily change or update data.

- **Inline editing:** Users can edit cells right on the page, just like in a spreadsheet, without needing to reload.
- **Add/delete rows:** You can create new records or delete the ones you don't need anymore.
- **CRUD operations:** Fully supports creating, reading, updating, and deleting data.
- **Master-detail relationships:** Easily set up master-detail forms with multiple interactive grids on one page.

Architecture of the Interactive Grid.

Architecture of the Interactive Grid.

The Interactive Grid is a highly complex JavaScript construct that combines the advantages of a powerful client-side model (speed, interactivity) with the declarative configuration of APEX.

The architecture can be divided into three main components:

- Client side (browser)
- Data model
- Server side (APEX engine & database)

The Ups and Downs of the Interactive Grid!

The Ups and Downs of the Interactive Grid!

- **Maximum interactivity:** Allows direct inline editing of data like Excel
- **Combined functionality:** Merges reporting functions with editing features in one component
- **Client-side architecture:** Uses a client-side data model for fast execution
- **Extensibility:** Offers a JavaScript API for customizing
- **Master-detail capability:** Simplifies creating master-detail forms
- **Modern look:** Integrates with the Universal Theme for a modern appearance



The Ups and Downs of the Interactive Grid!

- **Complexity for developers:** The client-side data model (apex.model) is more complex than other reports, requiring advanced JavaScript knowledge for customizations.
- **Performance with large data sets:** IG is optimized for scrolling, but handling very large data sets can still slow performance compared to server-side reports.
- **Overhead without editability:** If editing isn't needed, IG can add unnecessary loading time and JavaScript logic compared to simpler reports.
- **Learning curve:** Users who want to use the full range of features may need a short training period to understand all the options



APEX

App Builder

SQL Workshop

Team Development

Gallery

Search

TH

Timo Herwix

tm-apex

Application 109

Edit Page 1

Interactive Grid Demo

Edit Application Definition

Run Application

Supporting Objects

Shared Components

Utilities

Export / Import

Go

Actions

Create Page

0 - Global Page

1 - Home

2 - Server-Side Condition (MLE)

3 - Show an additional scrollbar on top

4 - Getting Selected Records

5 - CRUD Operations Like a Pro

6 - Skip Readonly Cells

7 - Change No data found Icon

8 - Hide deleted Rows

9 - Activate Edit Mode

10 - Column Heading Menu

11 - Column formatting

12 - Validation Styling

13 - Column-Groups Styling

15 - Show/Hide Columns

19 - HTML Expression & Template Di...

9999 - Login Page

1 - 17

About

Use the Application home page to run, edit, import, export, copy, or delete an application. Select a page to edit the page properties, or click **Create Page** to add a page to the application.

Learn More ...

Tasks

Delete Application

Copy Application

Add Feature Page

Browse by Facets

Workflow Designer

Working Copy Tasks

Create Working Copy

Recently Edited Pages

13. Column-Groups Styling

11. Column formatting

3. Show an additional scrollba...

7. Change No data found Icon

4. Getting Selected Records

19. HTML Expression & Templ...

2. Server-Side Condition (MLE)

5. CRUD Operations Like a Pro

Backups

Last Backup: 14 hours ago

1

Interactive Grid
like a Pro

2

JavaScript
Enhancements

3

CSS
Enhancements

Try to keep the code logic in the APEX app to a minimum and shift it over to a PL/SQL package instead. This way, it's easier to reuse and maintain!

CRUD Operations Like a Pro

Process

Filter

Identification

Name: CRUD Operations Like a Pro - Save Interactive Grid Data

Type: Interactive Grid - Automatic Row Processing (DML)

Execution Chain: None

Editable Region: CRUD Operations Like a Pro

Settings

Target Type: Table / View

Table Owner: Parsing Schema

Table Name: SP_TEAM_MEMBERS

Prevent Lost Updates: ☒

Lock Row: Yes

Return Primary Key(s) after Insert: ☒



CRUD Operations Like a Pro

Process

Filter

Identification

Name: CRUD Operations Like a Pro - Save Interactive Grid Data

Type: Interactive Grid - Automatic Row Processing (DML)

Execution Chain: None

Editable Region: CRUD Operations Like a Pro

Settings

Target Type: PL/SQL Code

PL/SQL Code to Insert/Update/Delete

```
begin
  sp_team_member.sp_team_members_crud_apex;
end;
```

Prevent Lost Updates: ☒

Lock Row: Yes



CRUD Operations Like a Pro

The procedure, `sp_team_members_crud_apex`, is called from the APEX Process. For an IG, it's called once for each record you create, update, or delete, depending on how you set the Execution Scope property.

The whole point of this procedure is to grab the values sent to the APEX Process, fill up a PL/SQL record, and then call the right procedure to (C)reate, (U)pdate, or (D)elete the record.

You might think of it as extra work, but it's really helpful for keeping PL/SQL complexity out of our APEX page.

```
1  create or replace package body SP_TEAM_MEMBER as
2
3      PROCEDURE sp_team_members_crud_apex
4      IS
5
6          l_row_status      VARCHAR2(1);
7          lr_member_rec      sp_team_members%ROWTYPE;
8          l_new_id          sp_team_members.id%TYPE;
9
10     BEGIN
11
12         -- Determine if user (C)reated, (U)pdated or (D)eleted this record.
13         l_row_status := V('APEX$ROW_STATUS');
14
15         -- Only need the ID for Update and Delete.
16         IF l_row_status IN ('U','D') THEN
17             -- Save ID to the PL/SQL Record.
18             lr_member_rec.id := V('ID');
19         END IF;
20
21         IF l_row_status IN ('C','U') THEN
22             -- Populate PL/SQL record with values from Interactive Grid.
23             lr_member_rec.first_name := V('FIRST_NAME');
24             lr_member_rec.last_name  := V('LAST_NAME');
25             lr_member_rec.email      := V('EMAIL');
26         END IF;
27
28         -- Call appropriate API to Create, Update or Delete the member.
29         CASE l_row_status
30             WHEN 'C' THEN
31                 create_member (pr_member_rec => lr_member_rec, x_id => l_new_id);
32                 -- Return the new member ID (Primary key) to the field
33                 -- identified as the Primary Key in the APEX IG / Form Region
34                 apex_util.set_session_state ('ID', l_new_id);
35             WHEN 'U' THEN
36                 update_member (pr_member_rec => lr_member_rec);
37             WHEN 'D' THEN
38                 delete_member (p_id => lr_member_rec.id);
39         END CASE;
40
41     END sp_team_members_crud_apex;
42
43     -----
44
45     ...
46
47 end SP_TEAM_MEMBER;
```

Try to keep your business logic in SQL simple and free from any UI logic. This makes your SQL queries easier to read, improves performance, and reduces the risk of cross-site scripting attacks.

Keep your Code clean with Template Directives!

```
1  select ID,  
2      FIRST_NAME,  
3      LAST_NAME,  
4      EMAIL,  
5      CASE IS_CURRENT_YN  
6          WHEN 'Y' THEN 'fa fa-check-circle u-color-5-text'  
7          WHEN 'N' THEN 'fa fa-minus-circle u-color-9-text'  
8          ELSE 'fa fa-minus-circle u-color-9-text'  
9      END IS_CURRENT_YN,  
10     CREATED,  
11     CREATED_BY,  
12     UPDATED,  
13     UPDATED_BY  
14  from SP_TEAM_MEMBERS
```



Keep your Code clean with Template Directives!

Identification

Column NameIS_CURRENT_YN

TypeHTML Expression

Heading

HeadingIs Current

Alignment

Alternative Label

Settings

HTML Expression

```
{case IS_CURRENT_YN/}
{when Y/}
  <span class="fa fa-check-circle u-color-5-text"></span>
{when N/}
  <span class="fa fa-minus-circle u-color-9-text"></span>
{otherwise/}
  <span class="fa fa-minus-circle u-color-9-text"></span>
{endcase/}
```



Keep your Code clean with Template Directives!

≡

Interactive Grid Demo

 admin

▼

Q

▼

Search: All Text Columns

Go

Actions

▼

 Reset

First Name	Last Name	Email	Created	Created By	Updated	Updated By	Is Current
Timo	Doe	john.doe@acme.com	3/21/2025	ADMIN	9/23/2025	ADMIN	✓
Jane	Doe	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN	✓
Scott	Tiger	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN	✓
Timo	Herwix	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN	✓
Test	Test	test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN	✗
Test 2	Test2	test2@test2.de	3/24/2025	ADMIN	4/11/2025	ADMIN	✗

1 rows selected

Total 6

Don't try to reinvent the wheel!

Use what's already available to keep your code to a minimum.

Using MLE JavaScript modules for APEX validations.

```
// Regular expression pattern for email validation
const emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;

/**
 * Function to validate an email address
 * @param {string} email - The email address to validate
 * @returns {boolean} - True if the email is valid, false otherwise
 */
function validateEmail(email) {
    return emailPattern.test(email);
}

// Example usage
const emailToValidate = 'user@example.com';
if (validateEmail(emailToValidate)) {
    console.log('Valid email address!');
} else {
    console.log('Invalid email address.');
}
```



Using MLE JavaScript modules for APEX validations.

Starting with Oracle Database 21c, developers can now execute JavaScript within the database.

validator.js

npm v13.15.22 CI no status codecov 100% downloads rate limited by upstream service backers 9 sponsors 1

License MIT chat on github

A library of string validators and sanitizers.

Strings only

This library validates and sanitizes strings only.

If you're not sure if your input is a string, coerce it using `input + ''`. Passing anything other than a string will result in an error.

☒ Validation

Editable Region Team Members IG >

Type Function Body (returning Boolean) ⋮

Language JavaScript (MLE) v

JavaScript Function Body

```
const { default: v } = await import ('validator-module');  
return v.isEmail( apex.env.EMAIL );
```



Using MLE JavaScript modules for APEX validations.

JavaScript modules on NPM can really save you a lot of time and effort with coding.
All you need to do is set up an MLE Module & MLE Environment.

The screenshot displays the Oracle APEX SQL Workshop interface. On the left, the 'Object Browser' shows the schema 'APEX_WS'. The main area is titled 'Create Database Objects' with options for 'Table' and 'View'. A modal dialog titled 'Create MLE Module - JavaScript' is open, showing the following fields:

- Name: VALIDATOR_MODULE
- Version: 13.11.0
- Source Type: URL (selected)
- URL: <https://cdn.jsdelivr.net/npm/validator@13.11.0/+esm>

Buttons for 'Cancel' and 'Create MLE Module' are at the bottom of the dialog. On the right, the 'Object Browser' shows the schema 'APEX_WS' with a folder 'VALIDATOR_ENV'. An 'Add Import' dialog is open, showing the following fields:

- Module Owner: APEX_WS
- Module Name: VALIDATOR_MODULE
- Import Name: validator

Buttons for 'Cancel' and 'Create' are at the bottom of the dialog.

Using MLE JavaScript modules for APEX validations.

 Interactive Grid Demo

 **1 error has occurred** 
■ Please enter a valid email address 

Server-Side Condition (MLE)

 Search: All Text Columns

Go

Actions 

Edit

Save

Add Row

 Reset

☐		First Name	Last Name	Email	Created	Created By	Updated	Updated By
☒		Timo	Doe	john.doe@@me.de 	3/21/2025	ADMIN	9/23/2025	ADMIN
☐		Jane	Doe	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN
☐		Scott	Tiger	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐		Timo	Herwix	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐		Test	Test	test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN
☐		Test 2	Test2	test2@test2.de	3/24/2025	ADMIN	4/11/2025	ADMIN

1 rows selected

Total 6

1

Interactive Grid
like a Pro

2

JavaScript
Enhancements

3

CSS
Enhancements

JavaScript Enhancements

Get the Selected Row Column Values from an Interactive Grid and then process them, like in PL/SQL.



```
1 function( config ) {  
2     config.defaultGridViewOptions = {  
3         selectionStateItem: "P4_SELECTED_IDS",  
4         multiple: true  
5     };  
6     return config;  
7 }
```

Interactive Grid Demo admin

Getting Selected Records

	Search: All Text Columns	Go	Actions	Edit	Save	Add Row	Reset
☐	First Name	Last Name	Email	Created	Created By	Updated	Updated By
☐	Timo	Doe	john.doe@acme.com	3/21/2025	ADMIN	9/23/2025	ADMIN
☐	Jane	Doe	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN
☐	Scott	Tiger	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	Timo	Herwix	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	Test	Test	test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN

Selected Ids

JavaScript Enhancements

By default, when you switch between pages with pagination set to "Page," any selected records get lost!

Interactive Grid Demo

admin

Getting Selected Records

Q

Search: All Text Columns

Go

Actions

Edit

Save

Add Row

Reset

☐		First Name	Last Name	Email	Created	Created By	Updated	Updated By
☐		Timo	Doe	john.doe@acme.com	3/21/2025	ADMIN	9/23/2025	ADMIN
☐		Jane	Doe	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN
☐		Scott	Tiger	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐		Timo	Herwix	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐		Test	Test	test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN

<

>

1

2

>

>

1 - 5 of 6

Selected Ids

JavaScript Enhancements

Easy fix! Just add Persistent Selection 😊



```
1 function( config ) {  
2   config.defaultGridViewOptions = {  
3     selectionStateItem: "P4_SELECTED_IDS",  
4     multiple: true,  
5     persistSelection: true  
6   };  
7   return config;  
8 }
```

Interactive Grid Demo admin

Getting Selected Records

Q Search: All Text Columns Go Actions Edit Save Add Row Reset

☐	☐	First Name	Last Name	Email	Created	Created By	Updated	Updated By
☐	☐	Timo	Doe	john.doe@acme.com	3/21/2025	ADMIN	9/23/2025	ADMIN
☐	☐	Jane	Doe	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN
☐	☐	Scott	Tiger	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	☐	Timo	Herwix	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	☐	Test	Test	test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN

< < 1 2 > > 1 - 5 of 6

Selected Ids

JavaScript Enhancements

Skip the Read-Only Cells



```
1 function (options) {  
2     options.defaultGridViewOptions = {  
3         skipReadOnlyCells: true  
4     };  
5     return options;  
6 }
```

Interactive Grid Demo

admin

Skip Readonly Cells

Search: All Text Columns

Go

Actions

Edit

Save

Reset

		Activity Type Id	Project Id	Team Member Id	Initiative Id	Start Date	End Date	Comments	Ticket Number	Url	Activity Status	Tags	Created	Created By	Updated	Updated By
☒		Development	AAAE	Timo		2/7/2025	3/11/2025	Active development			CLOSED		3/21/2025	ADMIN	3/21/2025	ADMIN
☐		Development	AAAE	Timo		3/9/2025	3/11/2025	Demo prep			CLOSED		3/21/2025	ADMIN	3/21/2025	ADMIN
☐		Review	AAAE	Jane		3/13/2025	3/13/2025	Review			CLOSED		3/21/2025	ADMIN	3/21/2025	ADMIN

1 rows selected

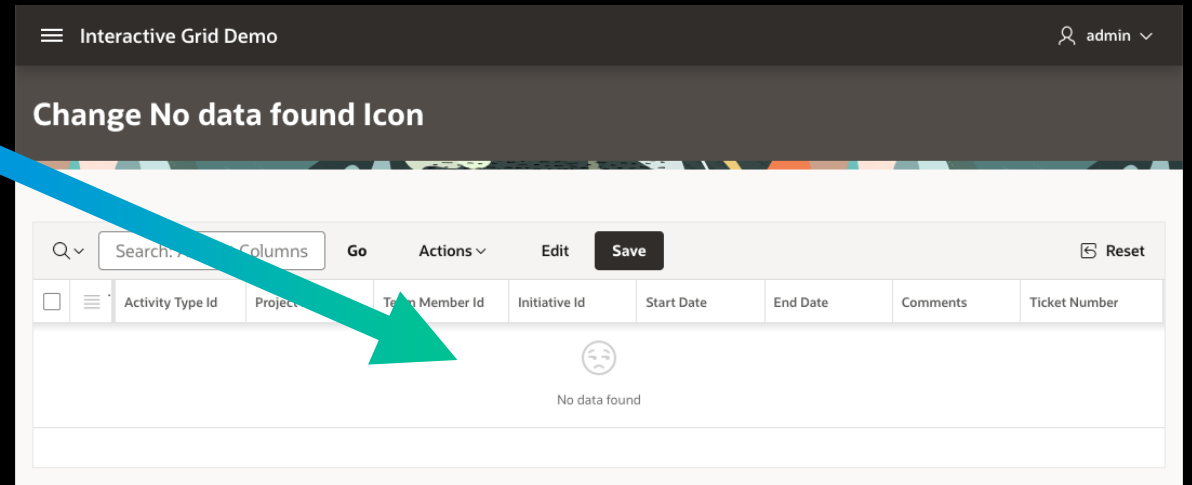
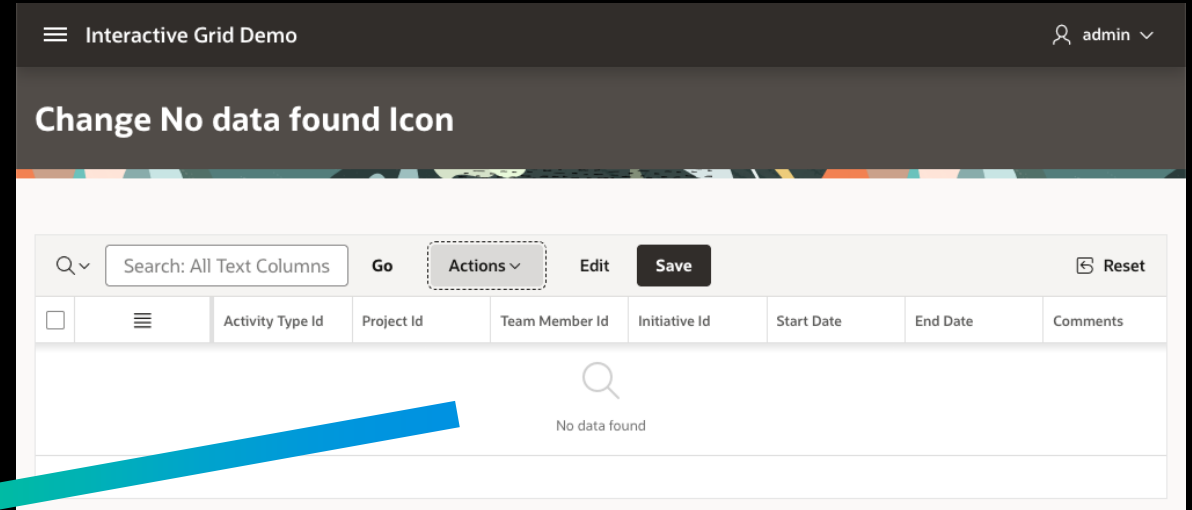
Total 3

JavaScript Enhancements

Change the "No Data Found" Icon



```
1 function(options) {  
2     options.defaultGridViewOptions = {  
3         noDataIcon: "fa fa-emoji-unamused"  
4     };  
5     return options;  
6 }
```



JavaScript Enhancements

Show/Hide Columns

≡ Interactive Grid Demo

admin ▾

Show Hide Audit ☒

←

Q ▾

Search: All Text Columns

Go

Actions ▾

Reset

Activity Type Id	Project Id	Team Member Id	Start Date	End Date	Comments	Activity Status	Created	Created By	Updated	Updated By
Development	AAAE	Timo	2/7/2025	3/11/2025	Active development	CLOSED	3/21/2025	ADMIN	3/21/2025	ADMIN
Development	AAAE	Timo	3/9/2025	3/11/2025	Demo prep	CLOSED	3/21/2025	ADMIN	3/21/2025	ADMIN
Review	AAAE	Jane	3/13/2025	3/13/2025	Review	CLOSED	3/21/2025	ADMIN	3/21/2025	ADMIN

1 rows selected

Total 3

JavaScript Enhancements

Show/Hide Columns



```
1  let auditColumns = [  
2      "CREATED",  
3      "CREATED_BY",  
4      "UPDATED",  
5      "UPDATED_BY"  
6  ],  
7  view = apex.region("p15_ig").widget().interactiveGrid("getCurrentView").view$;  
8  
9  auditColumns.forEach( function (columnName) {  
10     view.grid(  
11         (apex. items.P15_SHOW_HIDE_AUDIT.value === "y" ? "showColumn" : "hideColumn"),  
12         columnName  
13     );  
14 });
```

JavaScript Enhancements

Show/Hide Columns

Interactive Grid Demo

admin

Show Hide Audit ☒

Q	Search: All Text Columns	Go	Actions	Reset						
Activity Type Id	Project Id	Team Member Id	Start Date	End Date	Comments	Activity Status	Created	Created By	Updated	Updated By
Development	AAAE	Timo	2/7/2025	3/11/2025	Active development	CLOSED	3/21/2025	ADMIN	3/21/2025	ADMIN
Development	AAAE	Timo	3/9/2025	3/11/2025	Demo prep	CLOSED	3/21/2025	ADMIN	3/21/2025	ADMIN
Review	AAAE	Jane	3/13/2025	3/13/2025	Review	CLOSED	3/21/2025	ADMIN	3/21/2025	ADMIN
1 rows selected					Total 3					

JavaScript Enhancements

JavaScript API Documentation (<https://apex.oracle.com/jsapi>)

Index

Namespaces

- apex
- apex.actions
- apex.da
- apex.date
- apex.debug
- apex.event
- apex.item
- apex.lang
- apex.locale
- apex.message
- apex.model
- apex.navigation
- apex.navigation.dialog
- apex.navigation.popup
- apex.page
- apex.pwa
- apex.region
- apex.server
- apex.storage
- apex.theme

Introduction



Oracle APEX JavaScript API Reference

This section describes the JavaScript APIs available to Oracle APEX applications. You can use these functions to provide client-side functionality, such as showing and hiding page elements, or making Ajax (Asynchronous JavaScript and XML) requests.

Most of the APEX JavaScript APIs are organized into namespaces. A namespace is simply a global singleton object that contains a number of functions. There is one top level APEX namespace called `apex`. This has a number of sub namespaces such as `apex.server` and `apex.util`. Namespaces help to organize code and reduce the chance of name conflicts with other JavaScript libraries.

There are some older global functions that are not in a namespace. Most of these start with a \$ character. These are known as [Non-namespace APIs](#). Global symbols that start with `apex` or `$` are reserved by APEX.

Some functions return an interface that allows access to a specific instance of a page component or other entity. The returned interface is an object that contains functions known as methods and variables known as properties.

APEX also includes a number of UI widgets based on the jQuery UI widget factory. Widgets are high level user interface components such as menus, trees, or grids. APEX makes it easy to declaratively add components such as items and regions to a page. Internally some components are implemented using these widgets. Default component behavior does not require any JavaScript programming. To implement advanced use cases you can leverage the documented widget methods, options and events. The jQuery UI library in APEX is deprecated, but it is still used by some APEX components so a basic understanding of its widget factory can be helpful in understanding APEX widgets.

The jQuery library is used by APEX and is always loaded on every page. It can be used by your code.

1

Interactive Grid
like a Pro

2

JavaScript
Enhancements

3

CSS
Enhancements

Missing Scrollbar?

Sometimes a grid has tons of columns and ends up wider than the screen. This makes it hard to see the columns on the right. You have to scroll down to move the horizontal bar, then scroll back up. Kind of annoying, right?

CSS Enhancements

Missing Scrollbar?

Interactive Grid Demo

admin

Show an additional scrollbar on top

Q	Search: All Text Columns	Go	Actions	Reset		
Created	Created By	Updated	Updated By	Task Type	Static Id	Description
3/21/2025	ADMIN	3/21/2025	ADMIN	Review	REVIEW	Reviews
3/21/2025	ADMIN	3/21/2025	ADMIN	Functional Spec Review	REVIEW-FUNCSPEC	A review of the functional specification document
3/21/2025	ADMIN	3/21/2025	ADMIN	Spec Review	REVIEW-SPEC	A review of the implementation designed to implement a function
3/21/2025	ADMIN	3/21/2025	ADMIN	Peer Review	REVIEW-PEER	Review of the implementation by your fellow developer
3/21/2025	ADMIN	3/21/2025	ADMIN	Security Review	REVIEW-SECURITY	Review of the security exposure, remediations, and risk of this implementation
3/21/2025	ADMIN	3/21/2025	ADMIN	Accessibility Review	REVIEW-ACCESSIBILITY	Review web accessibility
3/21/2025	ADMIN	3/21/2025	ADMIN	Product Management Review	REVIEW-PM	Project Management (PM) team review
3/21/2025	ADMIN	3/21/2025	ADMIN	Documentation Review	REVIEW-DOC	Review of the documentation
3/21/2025	ADMIN	3/21/2025	ADMIN	Functional Testing Review	REVIEW-TESTING	Review of the testing plan
3/21/2025	ADMIN	3/21/2025	ADMIN	User Experience Review	REVIEW-UX	Review of end-user experience
3/21/2025	ADMIN	3/21/2025	ADMIN	Implementation Review	REVIEW-IMPLEMENTATION	An implementation review is typically performed before merge
3/21/2025	ADMIN	3/21/2025	ADMIN	Other Review	REVIEW-OTHER	A review that does not fall into any other category
3/21/2025	ADMIN	3/21/2025	ADMIN	Milestone	MILESTONE	Milestones

1 rows selectedTotal 30

CSS Enhancements

Show an additional scrollbar on top!



```
1 #my_ig .a-GV-w-hdr{  
2     overflow-x: auto !important;  
3 }
```

Interactive Grid Demo admin

Show an additional scrollbar on top

Created	Created By	Updated	Updated By	Task Type	Static Id	Description
3/21/2025	ADMIN	3/21/2025	ADMIN	Review	REVIEW	Reviews
3/21/2025	ADMIN	3/21/2025	ADMIN	Functional Spec Review	REVIEW-FUNCSPEC	A review of the functional specification document
3/21/2025	ADMIN	3/21/2025	ADMIN	Spec Review	REVIEW-SPEC	A review of the implementation designed to implement a function
3/21/2025	ADMIN	3/21/2025	ADMIN	Peer Review	REVIEW-PEER	Review of the implementation by your fellow developer
3/21/2025	ADMIN	3/21/2025	ADMIN	Security Review	REVIEW-SECURITY	Review of the security exposure, remediations, and risk of this im
3/21/2025	ADMIN	3/21/2025	ADMIN	Accessibility Review	REVIEW-ACCESSIBILITY	Review web accessibility
3/21/2025	ADMIN	3/21/2025	ADMIN	Product Management Review	REVIEW-PM	Project Management (PM) team review
3/21/2025	ADMIN	3/21/2025	ADMIN	Documentation Review	REVIEW-DOC	Review of the documentation
3/21/2025	ADMIN	3/21/2025	ADMIN	Functional Testing Review	REVIEW-TESTING	Review of the testing plan
3/21/2025	ADMIN	3/21/2025	ADMIN	User Experience Review	REVIEW-UX	Review of end-user experience
3/21/2025	ADMIN	3/21/2025	ADMIN	Implementation Review	REVIEW-IMPLEMENTATION	An implementation review is typically performed before merge
3/21/2025	ADMIN	3/21/2025	ADMIN	Other Review	REVIEW-OTHER	A review that does not fall into any other category

1 rows selected Total 30

CSS Enhancements

Column formatting using :has()

≡ Interactive Grid Demo

👤 admin ▾

Column formatting

Q ▾

Search: All Text Columns

Go

Actions ▾

Edit

Save

Add Row

🔄 Reset

☐	≡	Task Type	Project	Name	Status	Assigned To	Updated	Updated By	Created	Created By
☒	≡	Review	Project 1	Review: Functional Spec Review	Completed	Jane Doe	3/21/2025	admin	3/21/2025	admin
☐	≡	Review	Project 1	Review: Peer Review	Completed	Jane Doe	3/21/2025	admin	3/21/2025	admin
☐	≡	Review	Project 1	Review: User Experience Review	Identified	Jane Doe	3/21/2025	admin	3/21/2025	admin
☐	≡	Milestone	Project 1	Milestone: Spec	Completed	Timo Doe	3/21/2025	admin	3/21/2025	admin
☐	≡	Milestone	Project 1	Milestone: Demonstration	Completed	Timo Doe	3/21/2025	admin	3/21/2025	admin
☐	≡	Milestone	Project 1	Milestone: Code Complete	Completed	Timo Doe	3/21/2025	admin	3/21/2025	admin
☐	≡	Milestone	Project 1	Milestone: Merged	In Progress	Timo Doe	3/21/2025	admin	3/21/2025	admin

1 rows selected

Total 7

CSS Enhancements

Define an HTML expression using a template directive and add a bit of CSS for style.

▼ Identification

Column NameSTATUS

TypeHTML Expression

▼ Heading

HeadingStatus

Alignment

Alternative Label

▼ Settings

HTML Expression

<span class="status-{case STATUS/}
 {when Completed/}completed
 {when Identified/}identified
 {when In Progress/}in_progress
 {otherwise/}other
 {endcase/}"
>&STATUS.

```
1  td:has(.status-completed) {
2    --a-gv-row-hover-background-color: var(--a-palette-success);
3    background-color: var(--a-palette-success);
4    color: var(--a-palette-success-contrast);
5  }
6
7  td:has(.status-identified) {
8    --a-gv-row-hover-background-color: var(--a-palette-danger) ;
9    background-color: var(--a-palette-danger);
10   color: var(--a-palette-danger-contrast);
11 }
12
13 td:has(.status-in_progress) {
14   --a-gv-row-hover-background-color: var(--a-palette-warning);
15   background-color: var(--a-palette-warning);
16   color: var(--a-palette-warning-contrast);
17 }
18
19 td:has(.status-other) {
20   --a-gv-row-hover-background-color: var(--a-palette-color-14);
21   background-color: var(--a-palette-color-14);
22   color: var(--a-palette-color-14-contrast);
23 }
```

CSS Enhancements

Client-Side Validation Styling

Interactive Grid Demo

admin

Validation Styling

Q

Search: All Text Columns

Go

Actions

Edit

Save

Add Row

Reset

☐		First Name	Last Name	Initials	Zip code	Email	Created	Created By	Updated	Updated By
☒		Timo	Doe	JD	187	john.doe@acme.com	3/21/2025	ADMIN	9/23/2025	ADMIN
☐		Jane	Doe	JD	187	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN
☐		Scott	Tiger	ST	187	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐		Timo	Herwix	TH	187	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐		Test	Test	TT		test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN
☐		Test 2	Test2	TT		test2@test2.de	3/24/2025	ADMIN	4/11/2025	ADMIN

1 rows selected

Total 6

Release 1.0

^

CSS Enhancements

This CSS styles table cells in a grid view, highlighting error and changed states.

Error state cells:

- Red border and text color
- Shake animation to draw attention

Changed state cells:

- Green border and text color
- No animation

Shake animation:

- Creates a horizontal oscillation effect
- Enhances visibility for invalid entries



```
1  /* Target Item Container with invalid page item */
2  td.a-GV-cell.is-error:not(.is-active), td.a-GV-cell.is-error:not(.is-active):before{
3      /* item border color to red */
4      border-color: var(--a-palette-danger);
5      /* item value text color to red */
6      color: var(--a-palette-danger);
7      /* shake animation */
8      animation: shake-horizontal 0.5s cubic-bezier(0.455, 0.030, 0.515, 0.955) both;
9  }
10
11 /* Target Item Container with invalid page item */
12 td.a-GV-cell.is-changed:not(.is-active), td.a-GV-cell.is-changed:not(.is-active):before{
13     /* item border color to green */
14     border-color: var(--a-palette-success) var(--a-palette-success) transparent transparent;
15     /* item value text color to green */
16     color: var(--a-palette-success);
17 }
18
19 @keyframes shake-horizontal {
20     0% { transform: translateX(0) }
21     25% { transform: translateX(5px) }
22     50% { transform: translateX(-5px) }
23     75% { transform: translateX(5px) }
24     100% { transform: translateX(0) }
25 }
```

CSS Enhancements

Column-Groups Styling

☰ Interactive Grid Demo

👤 admin ▾

Column-Groups Styling

🔍 ▾ Search: All Text Columns

Go

Actions ▾

Edit

Save

Add Row

🔄 Reset

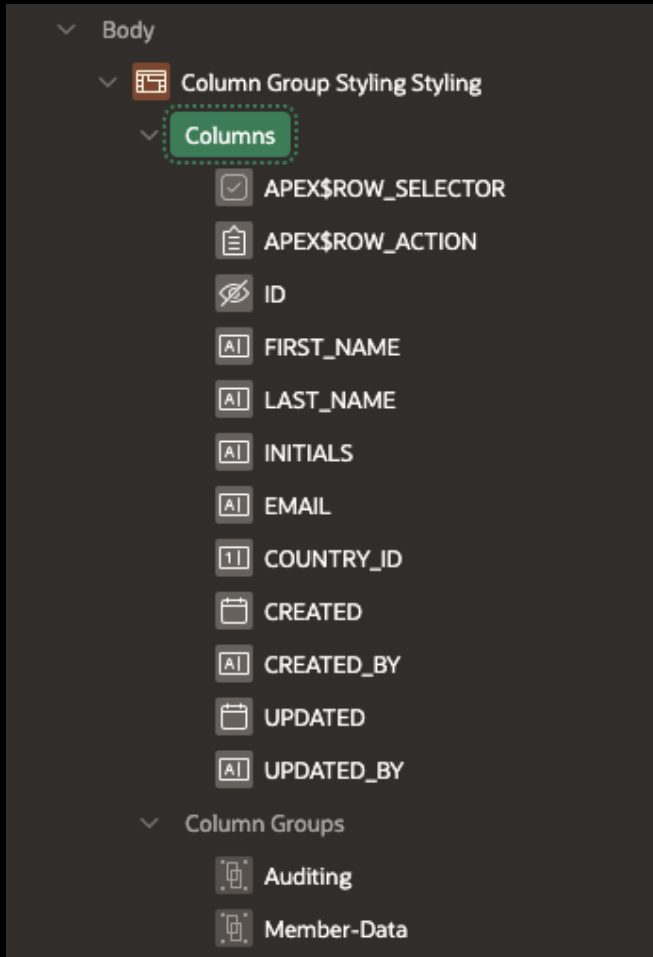
		Member-Data					Auditing			
☐	☰	First Name	Last Name	Initials	Zip code	Email	Created	Created By	Updated	Updated By
☒	☰	Timo	Doe	JD	187	john.doe@acme.com	3/21/2025	ADMIN	9/23/2025	ADMIN
☐	☰	Jane	Doe	JD	187	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN
☐	☰	Scott	Tiger	ST	187	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	☰	Timo	Herwix	TH	187	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	☰	Test	Test	TT		test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN
☐	☰	Test 2	Test2	TT		test2@test2.de	3/24/2025	ADMIN	4/11/2025	ADMIN

1 rows selected

Total 6

CSS Enhancements

Column-Groups Styling



```
1  #my_ig .a-GV-headerGroup[data-idx="1"] {
2      background-color: var(--a-palette-color-14);
3      color: white;
4      font-weight: bold;
5  }
6
7  #my_ig .a-GV-headerGroup[data-idx="2"] {
8      background-color: var(--a-palette-color-1);
9      color: white;
10     font-weight: bold;
11 }
```

```
> <th role="columnheader" class="a-GV-headerGroup" aria-roledescription="Group header" colspan="5" data-idx="1" tabindex="0">...
</th>
> <th role="columnheader" class="a-GV-headerGroup" aria-roledescription="Group header" colspan="4" data-idx="2" tabindex="-1">...
</th>
</tr>
```

CSS Enhancements

Column-Groups Styling

≡ Interactive Grid Demo

admin ▾

Column-Groups Styling

Q ▾

Search: All Text Columns

Go

Actions ▾

Edit

Save

Add Row

Reset

		Member-Data					Auditing			
☐	≡	First Name	Last Name	Initials	Zip code	Email	Created	Created By	Updated	Updated By
☒	≡	Timo	Doe	JD	187	john.doe@acme.com	3/21/2025	ADMIN	9/23/2025	ADMIN
☐	≡	Jane	Doe	JD	187	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN
☐	≡	Scott	Tiger	ST	187	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	≡	Timo	Herwix	TH	187	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	≡	Test	Test	TT		test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN
☐	≡	Test 2	Test2	TT		test2@test2.de	3/24/2025	ADMIN	4/11/2025	ADMIN

1 rows selected

Total 6

CSS Enhancements

Indeed, we want to add a background color to the column headings too. But there's no declarative CSS attribute for the column header cell! So, we'll need to add a bit of JavaScript, and then we can add a CSS rule for the column headings.



```
1 function(config) {  
2     config.defaultGridColumnOptions = {  
3         headingCssClasses: "master-header"  
4     }  
5     return config;  
6 }  
7  
8 function(config) {  
9     config.defaultGridColumnOptions = {  
10         headingCssClasses: "audit-header"  
11     }  
12     return config;  
13 }
```



```
1 #my_ig .master-header {  
2     background-color: var(--a-palette-color-14);  
3     color: white;  
4     font-weight: bold;  
5 }  
6  
7 #my_ig .audit-header {  
8     background-color: var(--a-palette-color-1);  
9     color: white;  
10    font-weight: bold;  
11 }
```

CSS Enhancements

Column-Groups Styling

☰ Interactive Grid Demo

👤 admin ▾

Column-Groups Styling

🔍 ▾ Search: All Text Columns

Go

Actions ▾

Edit

Save

Add Row

🔄 Reset

		Member-Data					Auditing			
☐	☰	First Name	Last Name	Initials	Zip code	Email	Created	Created By	Updated	Updated By
☒	☰	Timo	Doe	JD	187	john.doe@acme.com	3/21/2025	ADMIN	9/23/2025	ADMIN
☐	☰	Jane	Doe	JD	187	jane.doe@acme.com	3/21/2025	ADMIN	3/21/2025	ADMIN
☐	☰	Scott	Tiger	ST	187	scott.tiger@acme.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	☰	Timo	Herwix	TH	187	timo.herwix@hyand.com	3/21/2025	ADMIN	3/24/2025	ADMIN
☐	☰	Test	Test	TT		test@test.de	3/24/2025	ADMIN	4/11/2025	ADMIN
☐	☰	Test 2	Test2	TT		test2@test2.de	3/24/2025	ADMIN	4/11/2025	ADMIN

1 rows selected

Total 6

Wrap-Up

Wrap-Up

Interactive Grid Like a Pro: Best Practices and Enhancements

Code Structure:	Move your business logic into PL/SQL packages to make it easier to reuse and maintain, and keep your SQL clean from any UI logic.
JavaScript Enhancements:	Try to use the APEX JavaScript API if you want to dive deeper into customization.
CSS Styling:	To make the UI/UX better for your users, use CSS and pay attention to how things look (like color, font, and layout) and how the user feels when they use it.
Clean Code:	Don't try to reinvent anything that's already working well, and be sure to put any extra code in the right spots.

5 things to keep in mind



Only use it when you really need to



Makes things smoother for users



Think about security



Try not to overwhelm your users



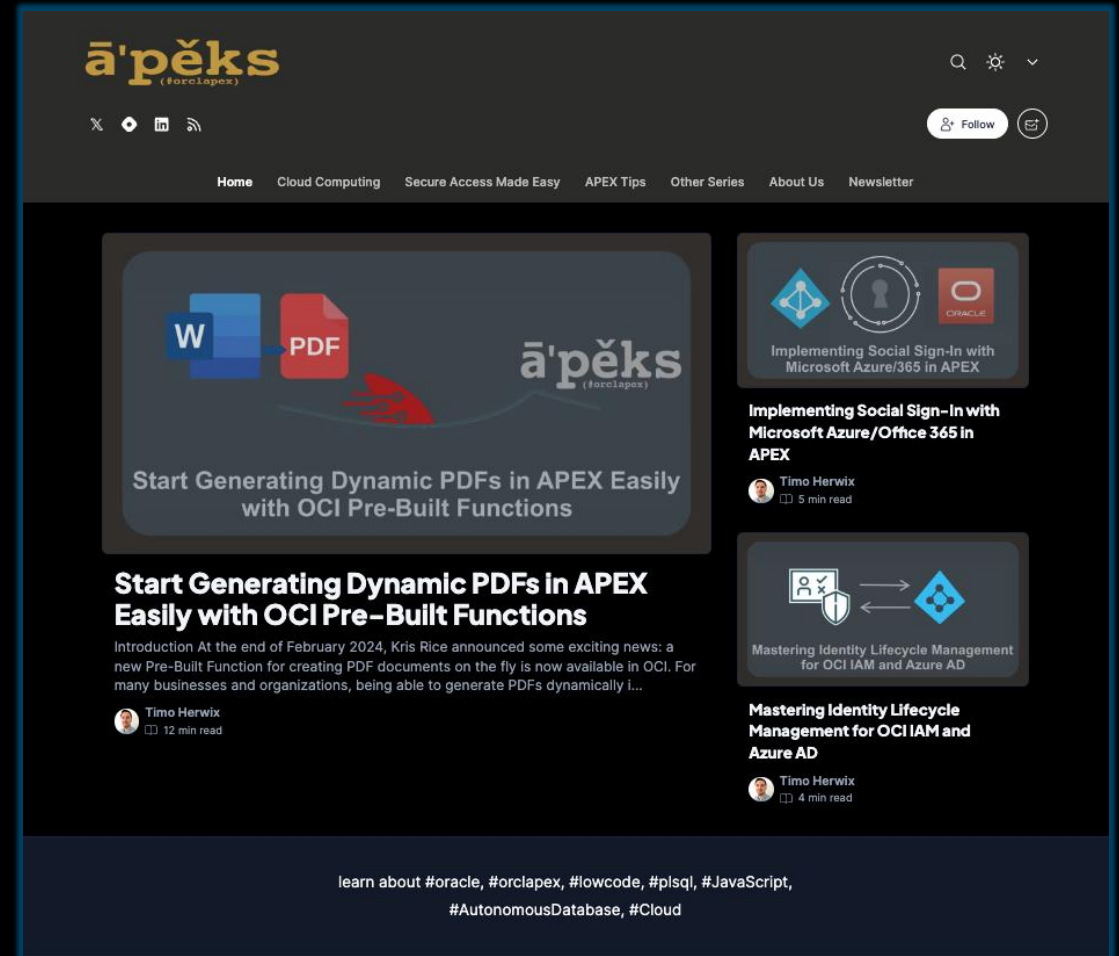
Make sure to understand everything you do



Blog



Scan me!



Are you interested?



Timo Herwix
Senior Consultant

Telefon: +49 2102 30 961-0
Mobil: +49 176 20185455
Mail: timo.herwix@hyand.com



Timo Herwix



@Therwix



tm-apex.hashnode.dev