




BECOMING A **JAVASCRIPT JEDI** IN APEX



Kevin Thyssen

March 22nd · Amersfoort · The Netherlands



 Oracle ACE
Associate

Senior APEX consultant @ United Codes

Located in Belgium

15+ years experience in Oracle APEX

Cycling enthusiast



@kevinthyssen



kevin.thyssen@united-codes.com



kevinthyssen

Extend APEX like a PRO



APEX Office Print
Printing & Reporting
built for APEX



APEX Media Extension
Image processing for the
Oracle Database.



APEX Office Edit
Document management
for APEX



Plug-ins Pro
Document management
for APEX

APEX Message Service
Real-time communication
for Oracle APEX



APEX Project Eye
Productivity, Quality Assurance,
and Security tool for APEX



APEX / JAVASCRIPT

APEX & JavaScript
JavaScript is everywhere

**PAGE
ITEMS**

Color Picker
Date Picker
Popup LOV
Rich Text Editor
...

COMPONENTS

Carousel
Classic / Interactive Report
Faceted Search
Interactive Grid
Map
Tree
...

**EXTERNAL
LIBRARIES**

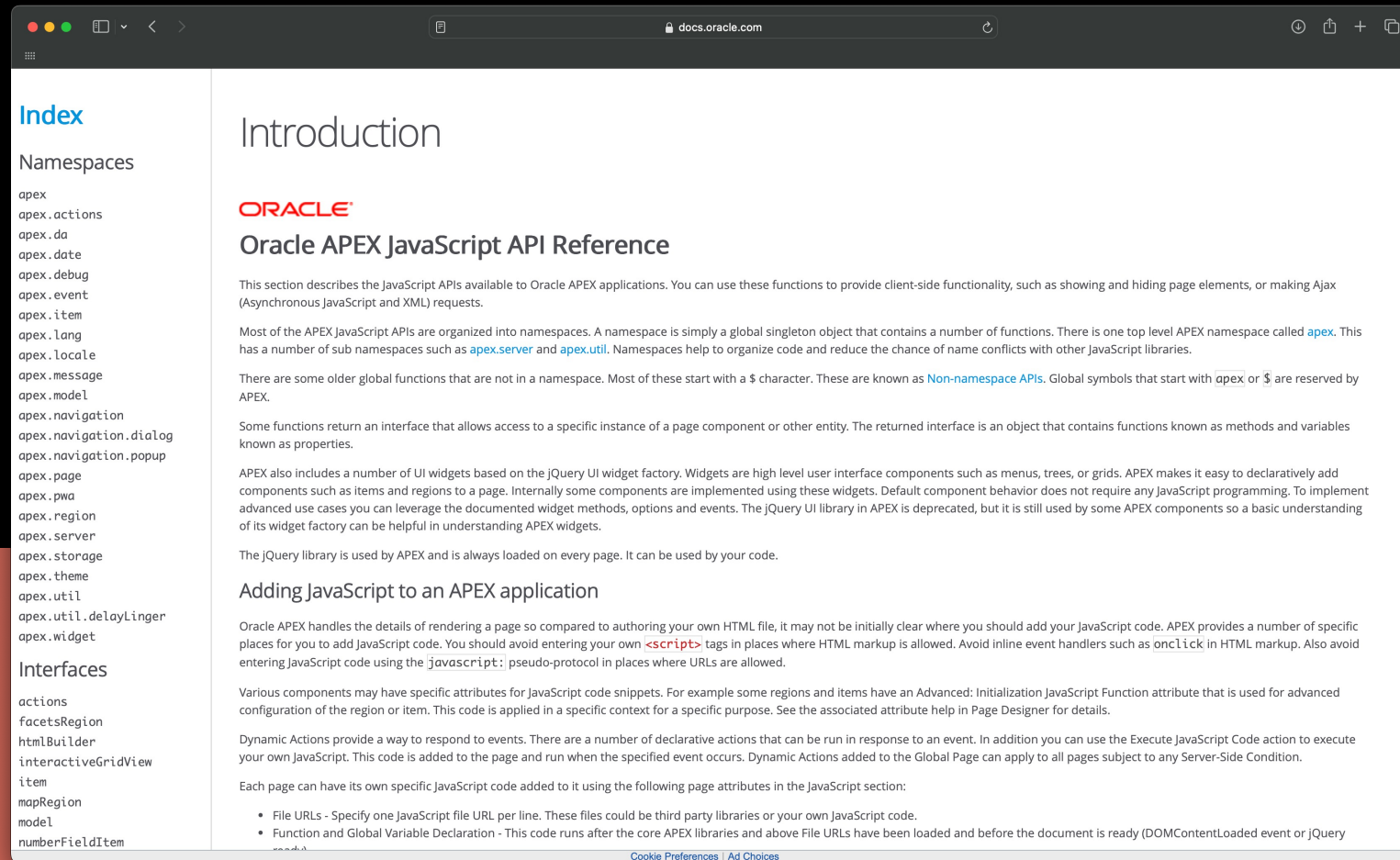
jQuery
Oracle JET
FullCalendar
TinyMCE
...

**DYNAMIC
ACTIONS**

APEX & JavaScript

Oracle APEX JavaScript API Reference

<https://apex.oracle.com/jsapi>





USE JAVASCRIPT
TO EXTEND
FUNCTIONALITY
OR ENHANCE
USER EXPERIENCE

APEX & JavaScript

Examples where JavaScript can add value

Client-side validations and interactions

- ✧ Input validations
- ✧ Hide/show input fields or regions conditionally

Asynchronous updates without page refresh

Add functionality not available by default

- ✧ e.g. tooltips

ADDING JAVASCRIPT TO AN APEX APPLICATION

Adding JavaScript to an APEX application

Most common places

Page

JavaScript

File URLs

#APP_FILES#javascript_jedi#MIN#.js

Function and Global Variable Declaration

```
function activateInactivate(pReport, pActivate, pIds) {  
  let result = apex.server.process('ACTIVATE_INACTIVATE', {  
    x01: pReport,  
    x02: pActivate,  
    f01: pIds  
  }, {  
    loadingIndicator: $(' #' + pReport),  
    loadingIndicatorPosition: 'centered'  
  });  
  
  result.done(function(data) {
```

Execute when Page Loads

Adding JavaScript to an APEX application

Most common places

Dynamic Action

The screenshot displays the APEX Dynamic Action configuration interface. The left sidebar shows a tree view of events and actions. The 'Dynamic Actions' tab is selected. Under the 'Change' event, the 'Toggle Characters' action is expanded, and the 'True' condition is selected. The 'Execute JavaScript Code' action is highlighted. The main panel shows the configuration for the 'Execute JavaScript Code' action. The 'Name' field is empty. The 'Action' dropdown is set to 'Execute JavaScript Code'. The 'Settings' section contains a code editor with the following JavaScript code:

```
const checkbox = this.triggeringElement;

if (checkbox.checked) {
  characters.push(checkbox.value);
} else {
  const index = characters.indexOf(checkbox.value);
  characters.splice(index, 1);
}
```

Static Application/Workspace Files

- ✧ Workspace Files are not exported with your app

The screenshot displays the APEX workspace editor for a file named `javascript_jedi.js`. The left sidebar contains configuration options:

- Directory:** An empty text input field.
- * File Name:** The text `javascript_jedi.js` is entered in the input field.
- Reference:** The value `#APP_FILES#javascript_jedi#MIN#.js` is shown with a copy icon.
- Mime Type:** The value `text/javascript` is shown with a help icon.
- File Character Set:** The value `Unicode UTF-8` is shown with a help icon.

The right pane shows the JavaScript source code with line numbers 1 through 35:

```
1 let jsJedi = (function() {
2     // Namespace variables are defined here.
3     let selectedIds = {};
4
5     return {
6         selectedIds: selectedIds,
7         toggleCheckbox: function(pElement) { ...
22     },
23     setCheckboxes: function() { ...
33     }
34 }
35 })();
```

Adding JavaScript to an APEX application

Most common places

User Interface Attributes

[Definition](#) [Security](#) [Globalization](#) **User Interface** [Progressive Web App](#)

Application 108 [Cancel](#) [Apply Changes](#)

[Show All](#) [Icon](#) [Logo](#) [Attributes](#) [Navigation Menu](#) [Navigation Bar](#) **[JavaScript](#)** [CSS](#) [Concatenated Files](#) [Advanced](#)

JavaScript

File URLs

Include Deprecated or Desupported Javascript Functions

☐ Pre 18.1 [?](#)
☐ 18.x

Include jQuery Migrate

☒ [?](#)

Adding JavaScript to an APEX application

Other options

APEX_JAVASCRIPT PL/SQL API

- ✧ Usually used for plug-in development

Files & File URLs in Plug-in definition

Component attributes for JavaScript code snippets

- ✧ e.g. Initialization JavaScript Function for regions or items

Adding JavaScript to an APEX application

Best practice



Declare at
page level



Create a file
and reference it

Adding JavaScript to an APEX application

Best practice

Make sure to include server side validations and check authorization when manipulating data through JavaScript

Try to avoid using the **javascript:** pseudo-protocol in URLs

▼ Target

Type	URL	▼
URL	javascript:apex.items.P1_ID.setValue(#ID#);	

DEBUGGING

Debugging

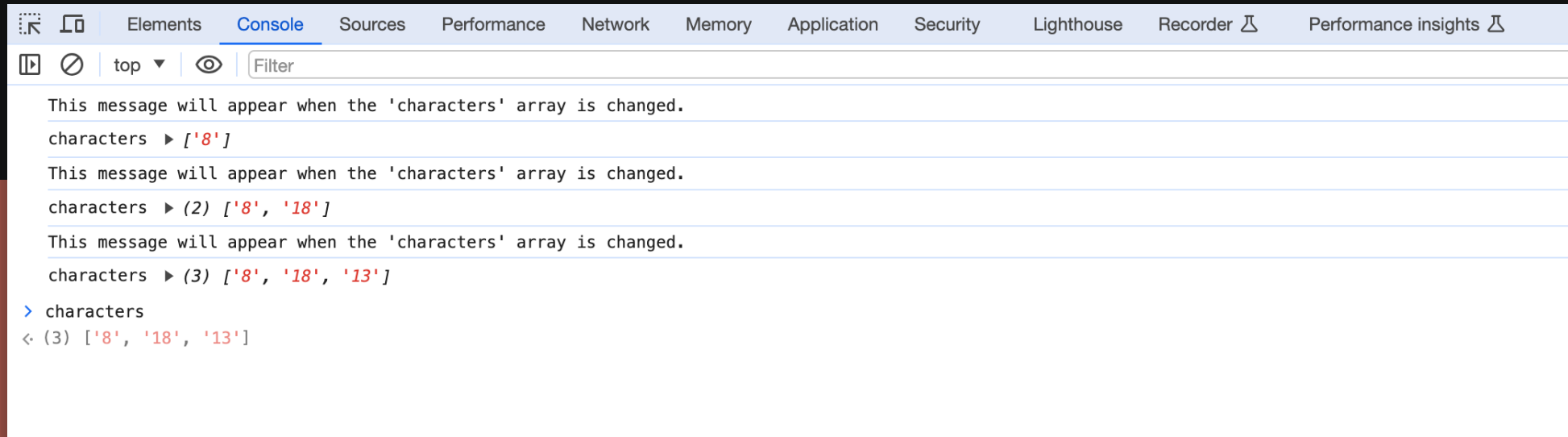
Browser console

```
let characters = [];
```

```
...
```

```
console.log('This message will appear when the \'characters\' array is changed.');
```

```
console.log('characters', characters);
```

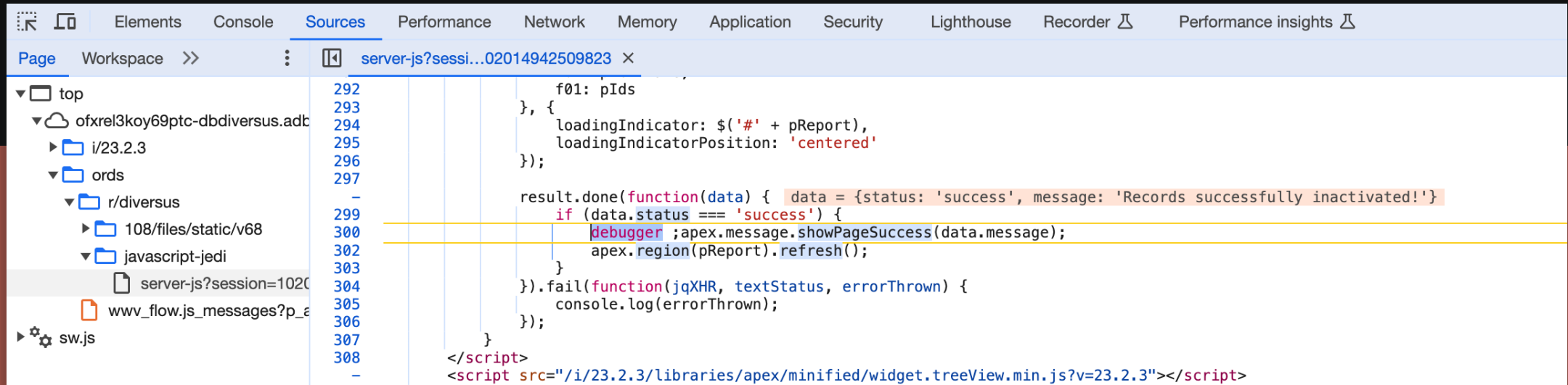


Debugging

Browser console

Set breakpoint(s) in browser console

Add *debugger*; statement to your code



Debugging

Browser console · apex.debug

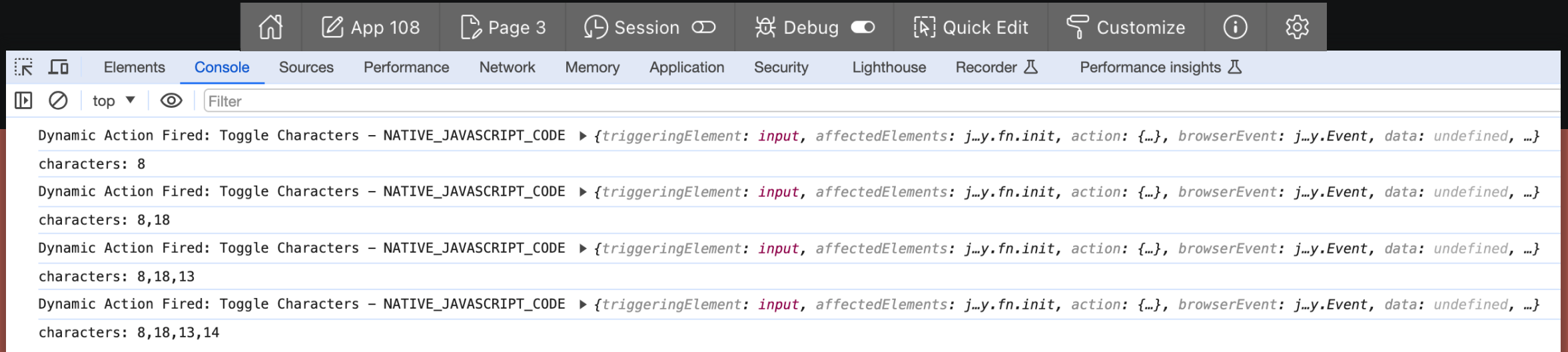
```
let characters = [];
```

```
...
```

```
apex.debug.info('characters: ' + characters.toString());
```

```
...
```

```
apex.debug.error('Something went wrong!');
```



DEMO

Demo

It's showtime!

Integrate JavaScript functionality

Make your code scalable

Enhance user experience





BECOMING A **JAVASCRIPT JEDI** IN APEX



Thank you!

kevin.thyssen@united-codes.com