1

Changes in This Release for JavaScript Developer's Guide

This chapter lists the changes in *Oracle Database JavaScript Developer's Guide* for Oracle Database 23ai:

- July 2024, Release Update 23.5
- January 2025, Release Update 23.7
- April 2025, Release Update 23.8

July 2024, Release Update 23.5

Included are some notable *Oracle Database JavaScript Developer's Guide* updates with Oracle Database 23ai, Release Update 23.5:

Feature	Description
MLE Support on Linux for Arm (aarch64)	In addition to Linux x86-64, Multilingual Engine (MLE) is supported on Linux for Arm (aarch64). Overview of Multilingual Engine for JavaScript
Operator Overloading with `OracleNumber`	Rather than using methods such as `add` and `sub` to perform arithmetic operations with instances of the type `OracleNumber`, arithmetic operators such as `+` and `-` are now supported as well. Examples using this new syntax can be found in Type Mapping. Server-Side JavaScript API Documentation

January 2025, Release Update 23.7

Included are some notable *Oracle Database JavaScript Developer's Guide* updates with Oracle Database 23ai, Release Update 23.7:

Feature	Description
Foreign Function Interface	The Foreign Function Interface (FFI) allows you to handle PL/SQL packages, functions, and procedures as JavaScript objects, providing more direct access to objects created in PL/SQL. Introduction to the PL/SQL Foreign Function Interface Server-Side JavaScript API Documentation
Fetch Type Handler	The `fetchTypeHandler` property of `mle-js-oracledb` is available to modify query result sets in JavaScript. Using the fetch type handler you can, for example, change the data types of the resulting row(s) of a `SELECT` statement. Server-Side JavaScript API Documentation

April 2025, Release Update 23.8

Included are some notable *Oracle Database JavaScript Developer's Guide* updates with Oracle Database 23ai, Release Update 23.8:

Feature	Description
Restricted JavaScript Execution Contexts	The `PURE` keyword is used in the creation of MLE environments and in inline call specifications to specify the use of a restricted execution context. During PURE execution, JavaScript code cannot access database state. About Restricted Execution Contexts