

Index

A

async/await interface, [6-11](#)

B

bind variables

IN, [7-11](#)

IN OUT, [7-11](#)

OUT, [7-11](#)

built-in modules, [7-32](#)

accessing, [6-11](#)

mle-encode-base64, [3-11](#)

mle-js-bindings, [3-11](#)

mle-js-encodings, [3-11](#)

mle-js-fetch, [3-11](#)

mle-js-oracledb, [3-11](#)

mle-js-plsql-ffi, [3-11](#)

mle-js-plsqltypes, [3-11](#)

C

call specification, [2-6](#), [6-1](#), [6-12](#)

creating, [6-1](#)

elements of, [6-4](#)

collections

creating, [8-8](#)

dropping, [8-11](#)

opening, [8-9](#)

during creation, [8-8](#)

committing operations (transactions), [8-29](#)

creating a collection, [8-8](#)

creating documents, [8-12](#)

D

Data Guard, [3-1](#)

data guide for a collection, getting, [8-27](#)

DBMS_MLE, [2-5](#)

EVAL procedure

arguments of, [2-5](#)

DBMS_MLE PL/SQL package, [4-1](#)

debugging

See post-execution debugging

debugpoints, [2-10](#)

elements of, [9-2](#)

debugpoints (*continued*)

specifying, [9-2](#)

deleting a collection

See dropping a collection

dictionary views

USER_MLE_ENVS view, [3-20](#)

USER_MLE_ENVS_IMPORTS view, [3-20](#)

USER_MLE_MODULES view, [3-13](#)

USER_SOURCE view, [3-12](#)

Direct Fetch, [7-6](#)

arrays, [7-7](#)

objects, [7-8](#)

documents

creating, [8-12](#)

finding in collections, [8-17](#)

inserting into collections, [8-14](#)

removing from a collection, [8-24](#)

replacing in collections, [8-22](#)

saving into collections, [8-15](#)

DRCP (database resident connection pool), [2-3](#)

dropping a collection, [8-11](#)

dynamic execution, [2-5](#), [4-1](#)

workflow, [4-2](#)

E

ECMAScript

available features, [7-1](#)

execution context, [2-8](#)

standard, [2-2](#), [2-4](#), [3-6](#)

environment, [3-1](#)

execution context, [2-5](#), [2-8](#), [10-5](#)

F

FFI (Foreign Function Interface), [7-38](#)

finding documents in collections, [8-17](#)

Foreign Function Interface, [7-38](#)

G

global variables, [6-11](#)

H

handling transactions, [8-29](#)

I

indexing documents in a collection, [8-25](#)
 initialization parameters
 MAX_STRING_SIZE, [2-3](#)
 MLE_PROG_LANGUAGES, [10-5](#)
 inline call specification, [6-1](#), [6-12](#)
 creating, [6-7](#)
 elements of, [6-10](#)
 inserting documents into collections, [8-14](#)

J

JavaScript, [2-2](#)
 global variables, [6-11](#)
 implementation of, [2-4](#)
 invoking, [2-5](#)
 loading from files, [4-3](#)
 providing inline, [4-2](#)

L

language options, [3-18](#)
 js.console, [3-19](#)
 js.polyglot-builtin, [3-19](#)
 js.strict, [3-19](#)

M

MAX_STRING_SIZE initialization parameter, [2-3](#)
 methods
 count(), [8-15](#), [8-17](#)
 create_context(), [10-8](#)
 createCollection(), [8-8](#)
 createIndex(), [8-25](#)
 drop(), [8-11](#)
 dropIndex(), [8-25](#)
 eval(), [2-5](#)
 execute(), [7-3](#)
 filter(), [8-15](#), [8-17](#)
 find(), [8-17](#)
 getCollectionNames(), [8-10](#)
 getContentAsString, [8-27](#)
 getCursor(), [8-15](#), [8-17](#)
 getDataGuide, [8-27](#)
 getOne(), [8-15](#), [8-17](#)
 hasNext(), [8-17](#)
 headerOnly(), [8-15](#), [8-17](#)
 key(), [8-15](#), [8-17](#)
 keys(), [8-15](#), [8-17](#)
 limit(), [8-15](#), [8-17](#)
 openCollection(), [8-9](#)
 read and write, [8-15](#)
 remove(), [8-15](#), [8-24](#)

methods (*continued*)
 replaceOne(), [8-15](#), [8-22](#)
 replaceOneAndGet(), [8-15](#), [8-22](#)
 reset_package(), [10-5](#)
 save(), [8-15](#)
 saveAndGet(), [8-15](#)
 skip(), [8-15](#), [8-17](#)
 terminal and nonterminal, [8-15](#)
 version(), [8-15](#), [8-17](#)
 MLE, [2-1–2-3](#)
 call specification, [6-1](#)
 environment, [3-1](#)
 inline call specification, [6-1](#)
 module, [3-3](#), [5-1](#)
 See also module
 MLE JavaScript SQL driver, [3-11](#), [7-1](#)
 selecting data, [7-6](#)
 MLE LANGUAGE clause, [6-7](#)
 MLE_PROG_LANGUAGES initialization
 parameter, [10-5](#)
 mle-js-fetch, [2-4](#)
 mle-js-oracledb
 See MLE JavaScript SQL driver
 module, [3-3](#)
 creating, [3-4](#)
 importing, [3-6](#), [5-1](#)
 managing, [3-3](#)
 naming, [3-3](#)
 module call, [2-6](#)
 Multilingual Engine
 See MLE

N

node package manager, [2-2](#)
 nonterminal method
 definition, [8-17](#)
 NPM (node package manager), [2-2](#)

O

opening a collection
 during creation, [8-8](#)
 opening existing collections, [8-9](#)
 ORA_SECURECONFIG audit policy, [10-10](#)

P

post-execution debugging, [2-10](#), [9-2](#)
 privileges, [2-4](#), [10-8](#)
 ALTER ANY MLE, [10-3](#)
 COLLECT DEBUG INFO, [10-4](#)
 CREATE ANY MLE, [10-3](#)
 CREATE ANY PROCEDURE, [10-3](#)
 CREATE MLE, [10-3](#)
 CREATE PROCEDURE, [10-3](#)

privileges (*continued*)

- DB_DEVELOPER_ROLE, [10-3](#)
- DROP ANY MLE, [10-3](#)
- EXECUTE DYNAMIC MLE, [10-2](#)
- EXECUTE ON JAVASCRIPT, [10-2](#)
- INHERIT PRIVILEGES, [10-13](#)
- SODA_APP, [10-2](#)

PURE keyword, [2-9](#)

R

read methods, [8-15](#)

removing documents from a collection, [8-24](#)

replacing documents in collections, [8-22](#)

restricted execution context, [2-9](#), [10-5](#)

ResultSet object, [7-3](#), [7-6](#)

S

saving documents into collections, [8-15](#)

SBOM, [10-14](#)

Simple Oracle Document Access (SODA), [8-1](#)

single-byte character set, [3-5](#)

Smart-DB approach, [2-2](#)

SODA, [8-1](#)

software bill of material (SBOM), [10-14](#)

T

terminal method

definition, [8-17](#)

transaction handling, [8-29](#)

U

Unicode Standard, [3-5](#)

USER_MLE_ENV_IMPORTS view, [3-21](#)

USER_MLE_ENVS view, [3-21](#)

USER_MLE_MODULES view, [3-13](#)

USER_SOURCE view, [3-12](#)

W

write methods, [8-15](#)