ORACLE®

Making RESTful Web Services the Easy Way with Node.js



October 1–5, 2017 SAN FRANCISCO, CA

Dan McGhan Oracle Developer Advocate October 3, 2017



Get excited for...





Get excited for...



JavaScript in the Database!!!



JavaScript in the Database!!!

- New Multi-Language Engine from Oracle Labs
- Now available for <u>preview on OTN</u>
 - Via a Virtual Box VM
- Go to their talk Wednesday at 5:30pm
 - CON6744: Accelerating Your Database Applications Simply with Graal
 - Marriott Marquis (Yerba Buena Level) Salon 13
 - Ignore the description in the session builder

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



About me



- Dan McGhan
 - Oracle Developer Advocate
 - Focus on JavaScript and Oracle Database
- Contact Info
 - dan.mcghan@oracle.com
 - @dmcghan
 - -jsao.io

Agenda

- Rough intro to APIs and Node.js
- Manual API creation
- 3 API creation with Sails.js
- 4 API creation with ORDS

Agenda

- Rough intro to APIs and Node.js
- Manual API creation
- 3 API creation with Sails.js
- 4 API creation with ORDS

What's REST?

- REpresentation State Transfer
 - Architectural style for distributed hypermedia systems
 - Originally defined in <u>Roy Fielding's doctoral dissertation</u>
- Defines six constraints

Uniform Interface	Stateless	Cacheable
Client-server	Layered System	Code on demand

Most implementations don't comply 100%



Client communicates intent via

URL paths based on nouns, not verbs

Туре	Example
Collection	http://server.com/api/employees
Resource	http://server.com/api/employees/101

HTTP methods

Method	CRUD/Database Action
POST	Create/INSERT
GET	Read/SELECT
PUT	Update/UPDATE
DELETE	Delete/DELETE

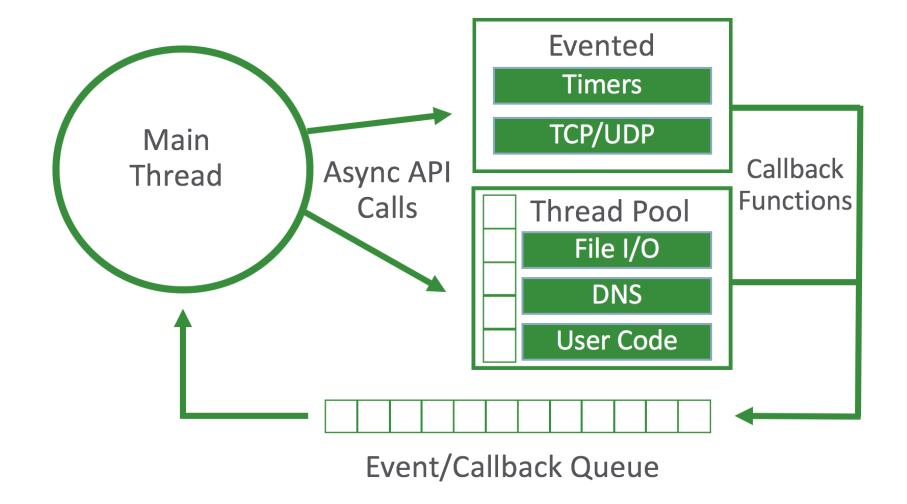


What is Node.js?

- Server-side JavaScript runtime
 - One language, front-end and back-end
- JavaScript runtime built on Chrome's V8 JavaScript engine
 - Though V8 is being decoupled to allow for other JavaScript engines
- Package ecosystem (NPM) is world's largest repo of open-source libraries
- Lightweight and efficient: event-driven, non-blocking I/O model



Node.js architecture





Roll your own or use a framework?

Roll your own

- More code
- More flexibility
- Lower short-term learning curve
- Lower long-term productivity

Framework

- Less code
- Less flexibility
- Higher short-term learning curve
- Higher long-term productivity



What features do you need to support?

- Pagination, sorting, & filtering
- Authentication & authorization
- Caching/ETag
- Doc/Swagger
- Real-time push/WebSockets
- Throttling
- Multiple representations (JSON, XML, CSV)
- CORS

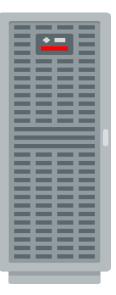


Client



Middle-tier







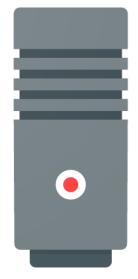
Web Server

accept HTTP requests, send HTTP responses

Client







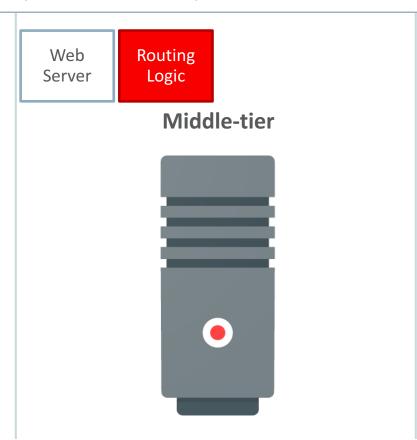


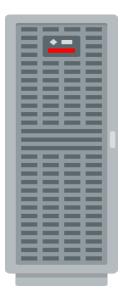


identify intent, route request to correct controller logic

Client



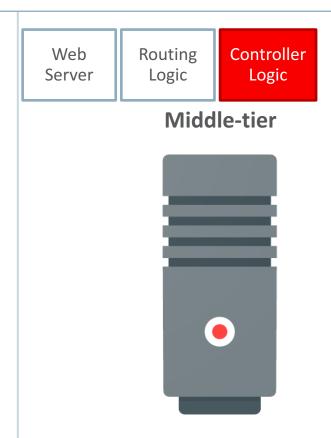


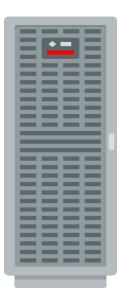


extract data, apply business & database logic, generate response

Client



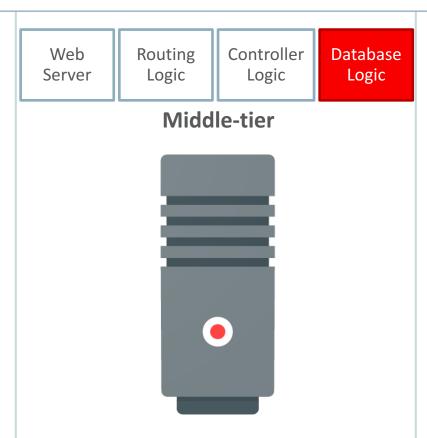


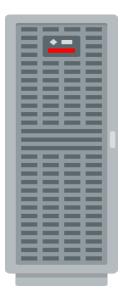


apply business logic, update persisted state

Client



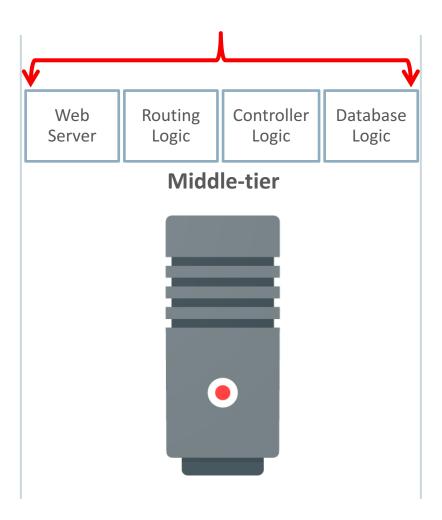


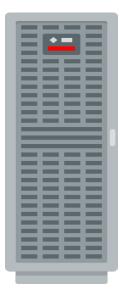


How should this be done? You decide!

Client







Agenda

- 1 Rough intro to APIs and Node.js
- Manual API creation
- 3 API creation with Sails.js
- 4 API creation with ORDS

Web server options

- Express
 - Highly popular, general purpose web server
- Restify
 - Web server focused on APIs, adds DTrace support
- Kracken
 - from PayPal, layer over Express, adds structure and convention
- Hapi
 - from Walmart, alternative to Express
 - Includes modules for validation, CORS, sessions, caching, logging, etc.

Database logic options

Drivers

- Low level, but extremely flexible
- May require specific database knowledge

ORMs

- Higher level, can be more productive
- Attempt to abstract away the database

node-oracledb class overview

Base class

- Get connections or create pools
- Set configuration parameters

Connection Pooling

- Dramatically increases performance
- Built-in pool cache for convenience

SQL and **PL/SQL** Execution

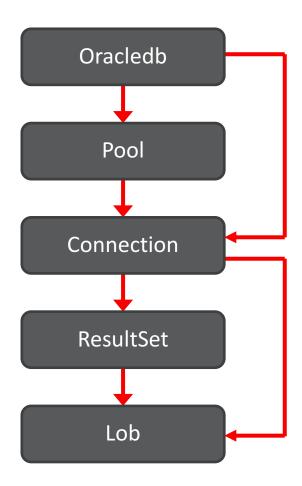
- Transaction support w/data type conversion
- Bind using JavaScript objects or arrays

Read-consistent, pageable cursor

- Used for large result sets
- Recursive callbacks or Node.js streams

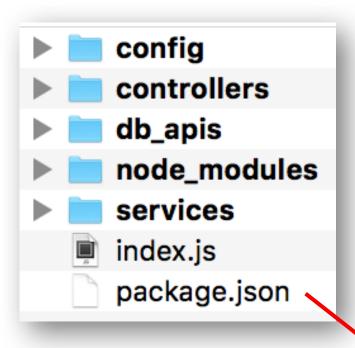
Large object support

- Can fetch smaller LOBs as string/buffer
- Stream large LOBs with this class





config
 controllers
 db_apis
 node_modules
 services
 index.js
 package.json



Name, description, dependencies, scripts



config
controllers
db_apis
node_modules
services
index.js
package.json

Main file, starts the application

Name, description, dependencies, scripts

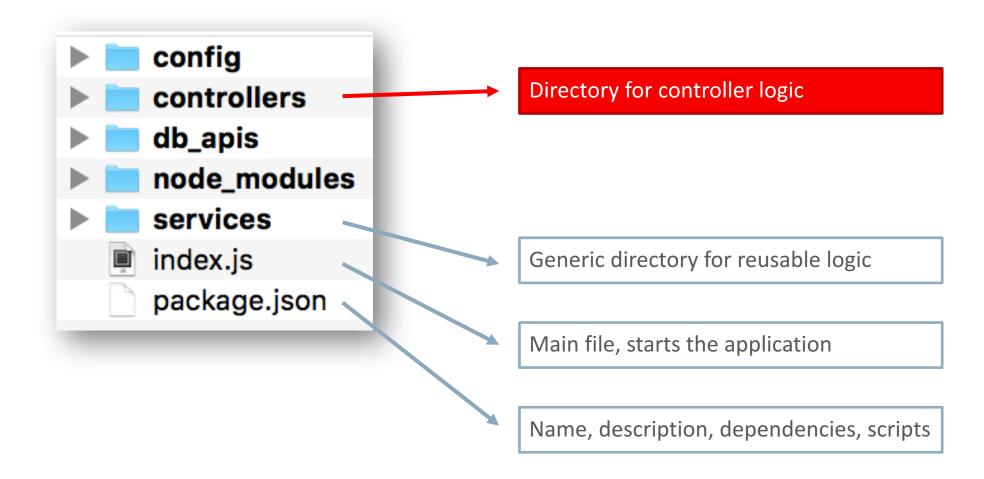


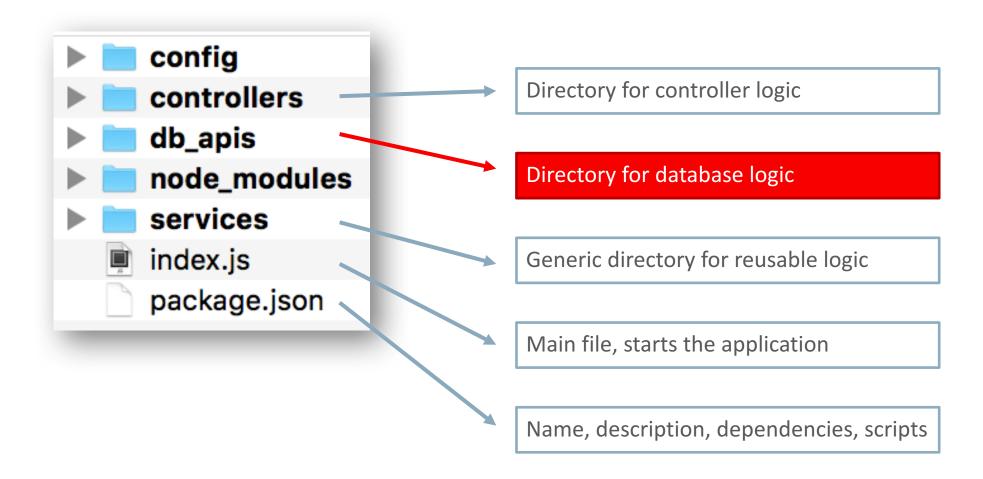
config
controllers
db_apis
node_modules
services
index.js
package.json

Generic directory for reusable logic

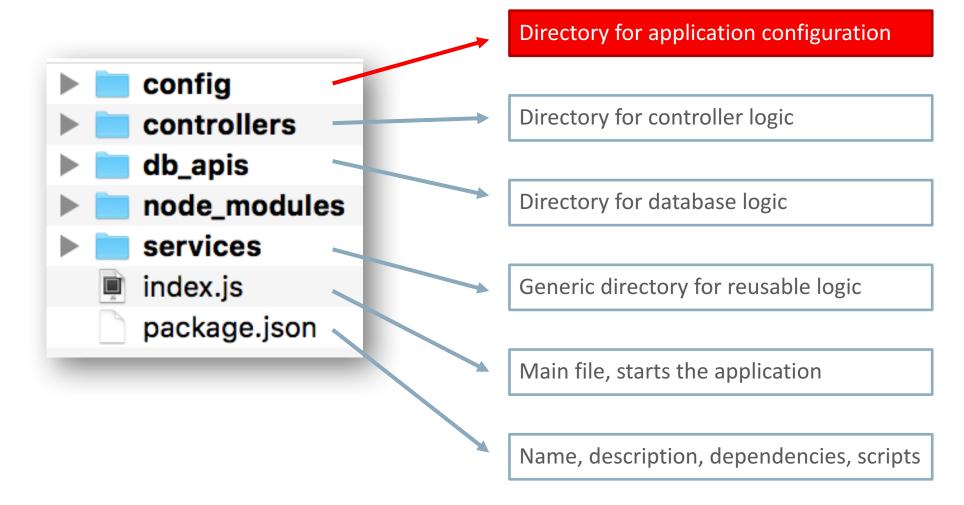
Main file, starts the application

Name, description, dependencies, scripts











Manual API Demo



Manual API review

- Have to manually implement CRUD functionality
 - Not so bad if you're comfortable with SQL (could even be scripted)
 - Could be advantageous for complex tasks (multiple tables, transactions, etc.)
- GET requests can be tough
 - Pagination not so bad, but a pain to repeat
 - No generic solution for converting JS objects to where clauses
 - No generic solution for converting JS objects to order by clauses



Agenda

- Rough intro to APIs and Node.js
- Manual API creation
- 3 API creation with Sails.js
- 4 API creation with ORDS

MVC and API Frameworks

- Frameworks offer all-in-one solutions
 - Include web servers
 - Generate routing, controller, and model logic
 - Many other features, vary by the framework
- Favor convention over configuration



Sails.js and Loopback

- Sails
 - An MVC framework for Node.js
 - Uses Waterline ORM for database modeling
 - Favorite features
 - Blueprints: easily prototype APIs
 - WebSocket integration
- Loopback
 - An API framework for Node.js
 - Uses a custom ORM for database modeling
 - Favorite feature
 - Discovery APIs for Oracle and others



Getting started with Sails.js

- Installation
 - -npm install sails -g
- Create an application
 - -sails new my-project
- Start up the application
 - -cd my-project
 - -sails lift
- Adding the Oracle adapter for Waterline
 - -npm install sails-oracle-db --save

Sails API Demo



Sails API review

- Sails brings out-of-the-box convention to each project
- Initial learning curve can be a little steep
 - But subsequent use gets better
- Blueprints can help with API prototyping

Use caution with ORMs

Some enforce case sensitivity

select column from user

becomes

select "column" from "user"

- A pain when you want to run ad-hoc queries
- Some don't use bind variables
 - Forces frequent hard parses
 - Loopback wins
- Some don't support transactions

Agenda

- 1 Rough intro to APIs and Node.js
- Manual API creation
- 3 API creation with Sails.js
- 4 API creation with ORDS

Introducing Oracle REST Data Services (ORDS)

- Provides data access consistent with modern app-dev frameworks
 - Mid-tier application
 - Can map RESTful requests to SQL
 - Can return results in JSON format
 - JavaScript friendly
- Services
 - HTTP(s) relational data access
 - Oracle JSON collection based schemaless access
 - Oracle NoSQL access over HTTP
 - Oracle APEX mid-tier, web toolkit apps, mod_plsql replacement



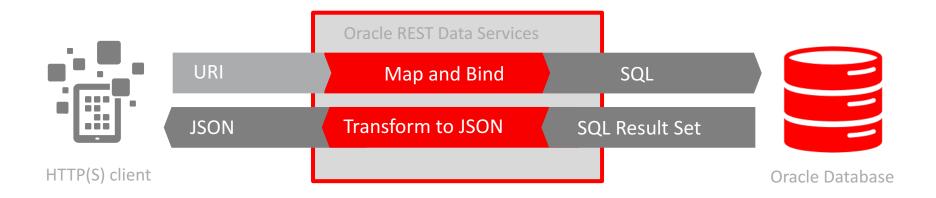


ORDS release history

Version	Date	Description
1.0	2010	First release as Oracle APEX Listener with with support for OWA toolkit used by APEX
1.1	2011	First release with REST support for JSON, Microdata, CSV, Pagination. Also added FOP
2.0	2012	OAuth2 support, Integrated with APEX, Multi Database, SQL Developer integration
2.0.5	2013	Added PDB support
2.0.6	2014	Renamed to Oracle REST Data Services to emphasize REST commitment
2.0.8	2014	Added REST Filtering
3.0	2016	REST AutoTable, NoSQL, DB12 JSON, Bulk loading over REST



Serving JSON results from relational data



- Data stored in standard relational tables and columns
- ORDS developer defines URI => SQL mapping
- App developer calls named URI over HTTP(S)
 - Works for JavaScript (Node.js), Java, Python, PHP, Ruby, etc.



ORDS API Demo



ORDS review

- ORDS can greatly simplify API creation on Oracle Database
 - AutoREST for tables
 - Built-in support for pagination, filtering, even "asof"
- Has a built in OAuth Server
- Can use ORDS in combination with Node.js
 - Hide ORDS behind a firewall
 - Proxy select requests from Node.js to ORDS (use authentication for Node.js server)
 - Look at <u>node4ords</u> for an example

Summary

- There's no perfect solution for creating REST APIs with Node.js
- Manual API implementations provide the most control
 - But also require the most knowledge and code
 - I'd like to see <u>a layer</u> over the driver for where, order, limit, and offset
- ORMs provide abstractions for databases
 - But may not use the database "correctly"
- API frameworks provide convention
 - But only make sense if you agree with those conventions

Next steps: learn and *code*!

Web Servers

- Express
- Restify
- Hapi
- Kracken

Drivers and ORMs

- Driver
 - <u>node-oracledb</u>
- Query/Schema Builder
 - Knex
- ORMs
 - <u>Sequelize</u>
 - Bookshelf
 - Waterline

Frameworks

- Sails
- Loopback
- ORDS



ORACLE®