

NativeScript Kinvey

PASOE

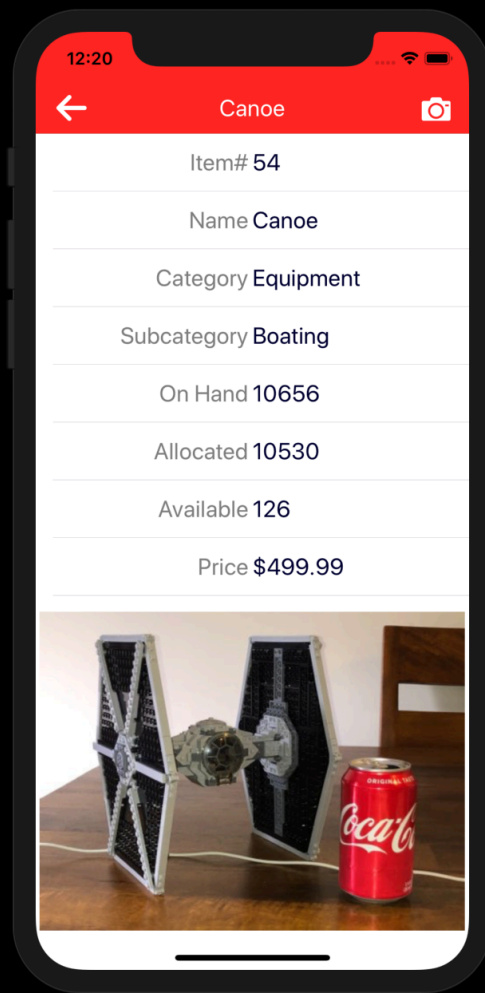
Lunch & Learn on how to create a mobile app using
NativeScript w/ Kinvey BAAS and OpenEdge PASOE

Mauricio dos Santos
April 2019

NativeScript Kinvey PASOE Architecture

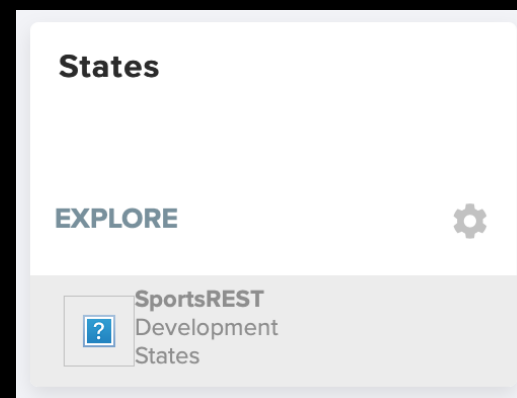
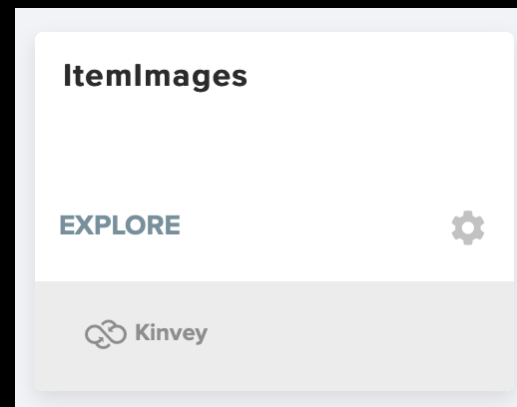
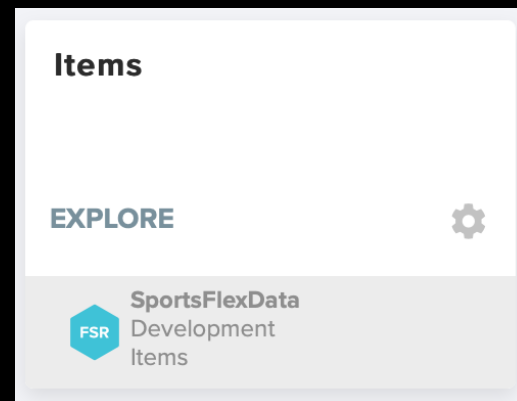
NativeScript

Kinvey SDK

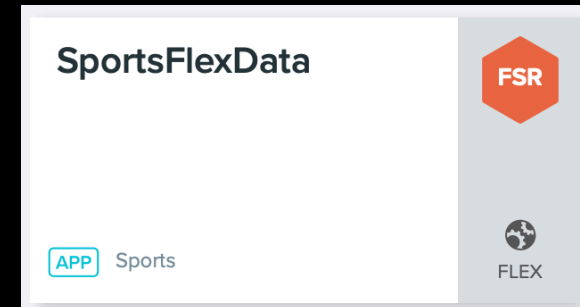


iPhone X - 12.1

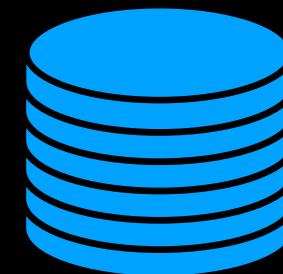
Collections



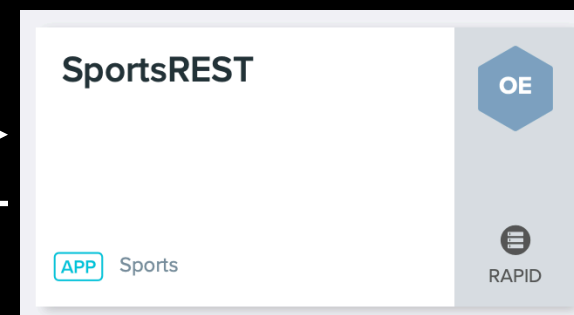
Kinvey



Kinvey Flex (JSDO)

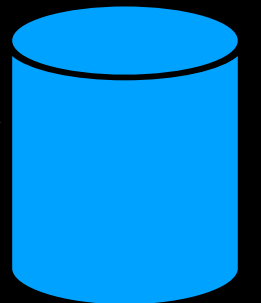


Kinvey Data Store

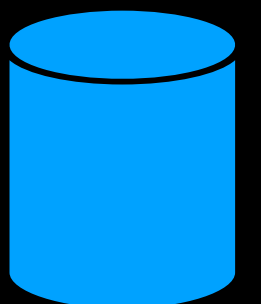


Kinvey RapidData ProgressData

PASOE



OpenEdge Database








OpenEdge Database

NativeScript

- Single source-code with tag based UI declaration and JavaScript, TypeScript, CSS, Vue and Angular
- Renders native UI on iOS and Android
- TypeScript + Angular provides OO and excellent binding
- TypeScript is a typed superset of JavaScript
- Angular lightweight framework provides a flexible structure including modularization, routing and data/event binding without enforcing MVC / no UI of its own
- Microsoft Visual Studio Code (free) has TypeScript IntelliSense support and NativeScript includes the compiler `nativescript-dev-typescript` as a dev dependency

Kinvey

-  Backend as a service (BAAS), low-code, scalable etc.
-  MongoDB data collections (~tables) comprised of JSON-like documents (Kinvey entities — ~schema-less records or rows)
-  Kinvey's own data stores are extremely efficient and ideal for mobile apps
-  RapidData connectors add CRUD support for external data
-  Flex Services created using node.js and can be used for almost anything (microservices)

PASOE

- Running on Progress arcade AWS VM instance w/ basic AUTH
- Simple Sports2000 database w/ updated dates (+20 years overall) but schema as-is
- Business entities created using PDSOE w/ additional JFP logic from Progress community site
- Avoid sorting issues: define temp-tables to be LIKE table + seq, id w/o redundant indexes: `DEFINE TEMP-TABLE ttCustomer LIKE Customer BEFORE-TABLE bttCustomer FIELD id AS CHARACTER FIELD seq AS INTEGER INDEX seq seq.`
- Add lots of MESSAGE statements on CRUD and INVOKE methods to ensure everything works as you like (and sometimes expect)

Sports App



- Plain NativeScript tags for UI w/ CSS and TypeScript / Angular



- Uses [kinvey-nativescript-sdk](#) for CRUD operations and one custom endpoint to call specialized PASOE REST API (invoke) which returns complex dataset (3 temp-tables)



- Connect to Kinvey using appKey and appSecret then login Kinvey's app user








- Kinvey's FSR (flex service run-time) used for flex data (CRUD) and flex function (custom endpoint) all using JSDO




- Flex service created as node project using [kinvey-flex-sdk](#) and deployed using Kinvey CLI

Sports App (demo)

-  • Login using sales rep initials from sports database (both username and password, e.g. BBB/bbb)
-  • App will show sales rep's customers (filter capability)
-  • Customers can be shown by number or name (sort capability)
-  • loadMoreItems: auto paging scroll will fetch next 25 records when user reaches bottom (app wide *limit* prop)
-  • App also shows and maintains customer orders and order lines (insert, update and delete via flex.) Camera plugin used to take picture to associate with an item (no flex.)

flex (how-to/demo)

- 
- Create flex service (FSR) within Kinvey's app at console.kinvey.com
 - On your computer, install node, then Kinvey CLI, create profile, login
 - Create and initialize node app w/ `node init` then `kinvey-flex-init`
 - Install flex SDK w/ `npm install kinvey-flex-sdk`
 - Develop and test locally using `node .` and postman
 - When ready, upload to FSR using `kinvey flex deploy`
 - Check upload status with `kinvey flex status`
 - Back at console.kinvey.com, configure Kinvey app to use your flex service

References

- Sports app: <https://github.com/mdossant/Sports.git>
- NativeScript: <https://nativescript.org>
- Kinvey: <https://www.progress.com/kinvey>
- Kinvey console: <https://console.kinvey.com>
- Mongo docs: <https://docs.mongodb.com>
- How to add JFP support to a Business Entity: https://community.progress.com/community_groups/mobile/w/mobile/2933.how-to-add-jfp-support-to-a-business-entity



“simple not simpler”

–Albert Einstein