## Introduction

This sample app shows integration of the javascript framework Redux with Kony.
Redux was built for maintaining state of Single page applications.
It is based on three principles
1) Single source of truth
2) State is read-only
3) Changes are made with pure-functions

For more info on redux visit https://redux.js.org

## Prerequisites

Before using the sample app, ensure you have the following

1. Kony account
2. Kony Visualizer Enterprise V8 SP1 and above
3. Kony MobileFabric V8 SP1 and above
4. Node JS server (Required only for the Node dependent part of sample app)

## Importing the App to Visualizer

Before you start importing the application to Kony Visualizer, you must download the application from the Kony Marketplace website.

Below are the steps to import the Redux Integration Sample App:

- Open Kony Visualizer.
- In menu Click Marketplace option and tap on Browse, then Kony Marketplace screen opens.
- Search for Redux Integration sample application.
- Select Redux Integration sample application.
- Click on “Import to Workspace”, then project gets imported into Kony visualizer.

The app doesn’t use any Kony Fabric services. However creating a Fabric app lets the app initialization send metrics for reporting purposes. For that purpose once the app has been imported to Visualizer workspace.

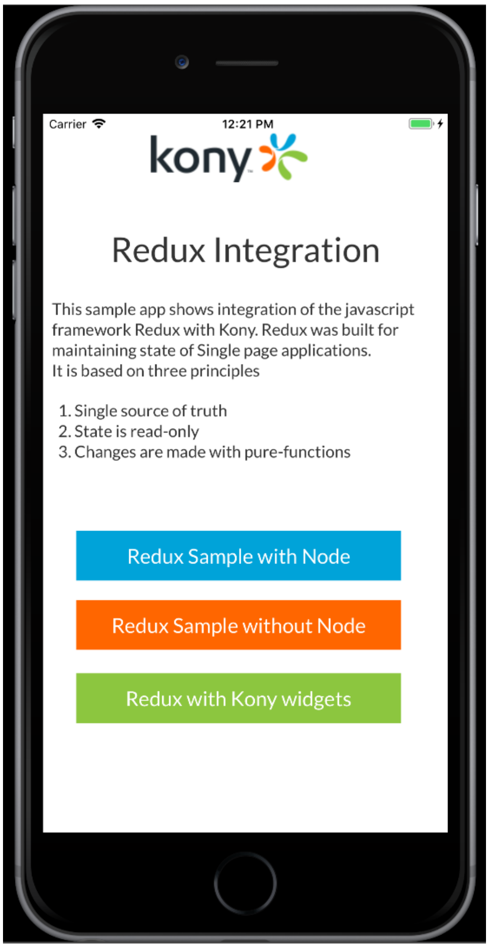
- Click on Kony Fabric in the left panel under the project tab in Kony Visualizer.
- Choose “Create New App”
- Login to the KonyFabric account by entering Kony Fabric account credentials in the popup
- Click on the Edit icon(pencil icon) in the top left corner and give this app a name like “Redux Integration”

- Click on the workspace icon (grid icon) in the top left and return back to the Visualizer codeview
- Click on File -> Publish to Kony Fabric from the top menu of Kony Visualizer
- Choose the Fabric server that you want to publish the app to and click on Publish

## App functionality

The sample app allows the user to add items to a todo list.
Enter the task and click on "Add Todo" to add the task.
Click on a task in the list below to mark the task as complete.
User can click "Show All" to see completed and todo tasks.
Click on "Active" to only see tasks that are todo (not completed)
Click on "Completed" to only see tasks that are done.
The sample app maintains its state using Redux JS.
It uses the redux actions, store, reducers, dispatch functionality.
You can learn more about redux basics from https://redux.js.org/basics

The app shows integration of Redux JS with Kony app in three different modes.
User lands on the landing page and chooses to view sample using one of the integration modes from the landing page.



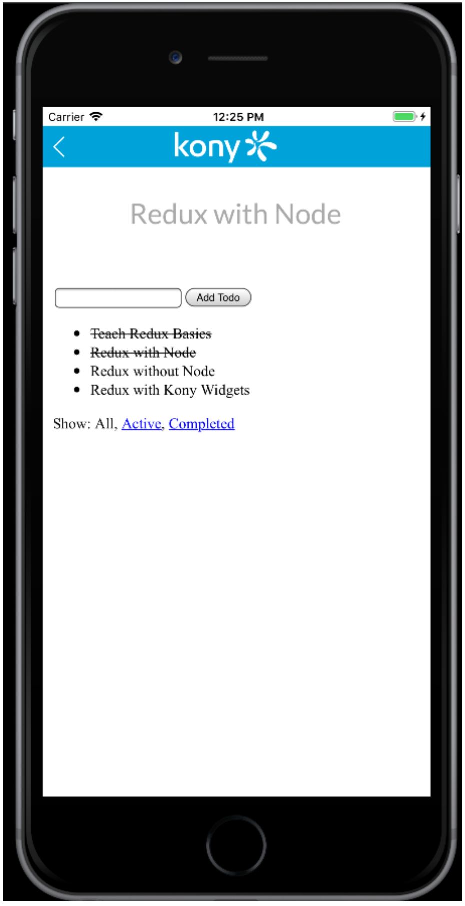
Redux Integration

This sample app shows integration of the javascript framework Redux with Kony. Redux was built for maintaining state of Single page applications.
It is based on three principles

1. Single source of truth
2. State is read-only
3. Changes are made with pure-functions

## 1) Redux with Node

This part of app requires app developer to download a sample app from https://redux.js.org
The sample shown in the screenshot below is the todo app from
https://github.com/reactjs/redux/tree/master/examples/todos



Redux with Node

- ~~Teach Redux Basics~~
- ~~Redux with Node~~
- Redux without Node
- Redux with Kony Widgets

Show: All, Active, Completed

## Configuration:

a. Install Node JS (if not already present). The code has been tested with v8.9.4. It might be compatible with earlier versions.

b. Download the source code from https://github.com/reactjs/redux/tree/master/examples/todos

c. Open a terminal or command prompt and navigate to the project directory.

d. Run
```
npm install
```
to install node dependencies for the project.

e. Run
```
npm start
```
to start the node server

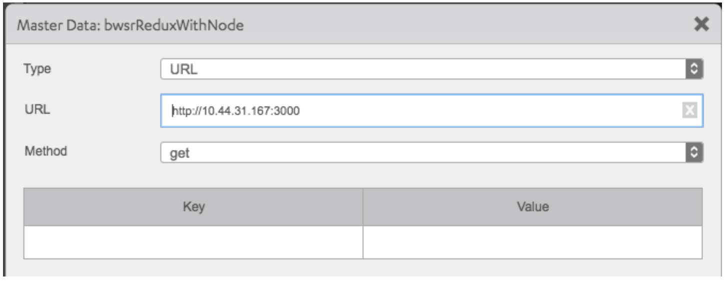
f. http://localhost:3000 will open in a browser with the sample app.

g. Note the IP address of the system that is running node

h. Open the client app in Kony Visualizer

i. Open frmReduxWithNode form in Mobile and edit the bwsrReduxWithNode
   - i. Right click the bwsrReduxWithNode in the left project panel
   - ii. Click on “Action: Edit Master Data”

iii. Choose type as URL and set the URL to the system that is running Node sample with the port and set method to get.



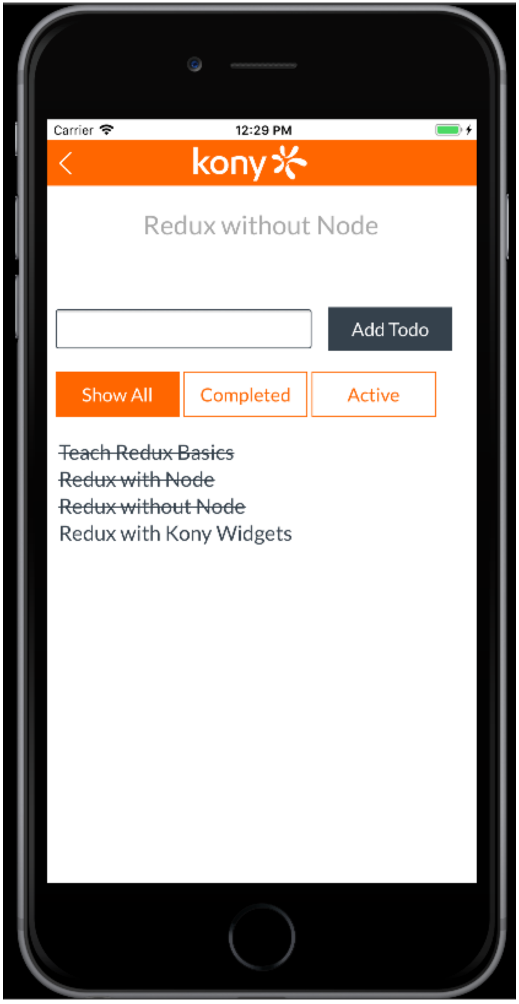
Master Data: bwsrReduxWithNode

| | |
|---|---|
| Type | URL |
| URL | http://10.44.31.167:3000 |
| Method | get |

| Key | Value |
|---|---|
| | |

## 2) Redux without Node

In this sample we have taken the Todo sample app and rewritten it completely in plain javascript removing the dependencies on Node JS and React JS frameworks.

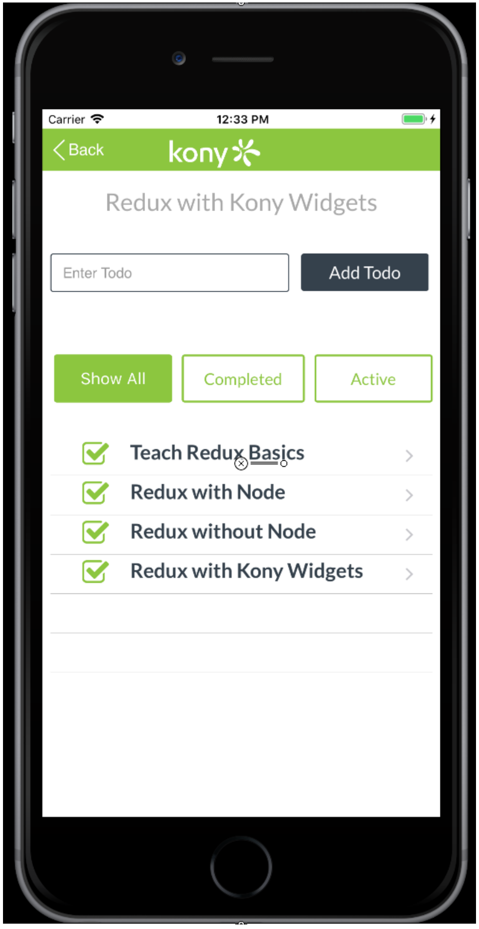


~~Teach Redux Basics~~
~~Redux with Node~~
~~Redux without Node~~
Redux with Kony Widgets

The code can be seen in the kony Visualizer project in the path
Web -> Localfiles.
redux.min.js is the minified version of the redux library.
The main.js file implements the Todo app using the redux library and simple webapp components of the html file.
The index.html is the UI for the Todo app.
The Kony app surfaces the index.html using a browser widget in the Redux without Node page.

## 3) Redux with Kony Widgets

This is the redux sample app that is fully integrated with Kony widgets.



In this sample the Todo sample app built for running on plain JS and web app, is rewritten to use Kony widgets like segments, button, textbox, checkbox, images and interact with them and use the redux JS library for maintaining the state of the app.
The code can be seen in the Kony Visualizer project in the Modules folder
Redux.js is the redux library code
Reduxlogic.js implements the Todo app using redux library and Kony widgets

Build and test the app in iOS/Android phone or simulators.