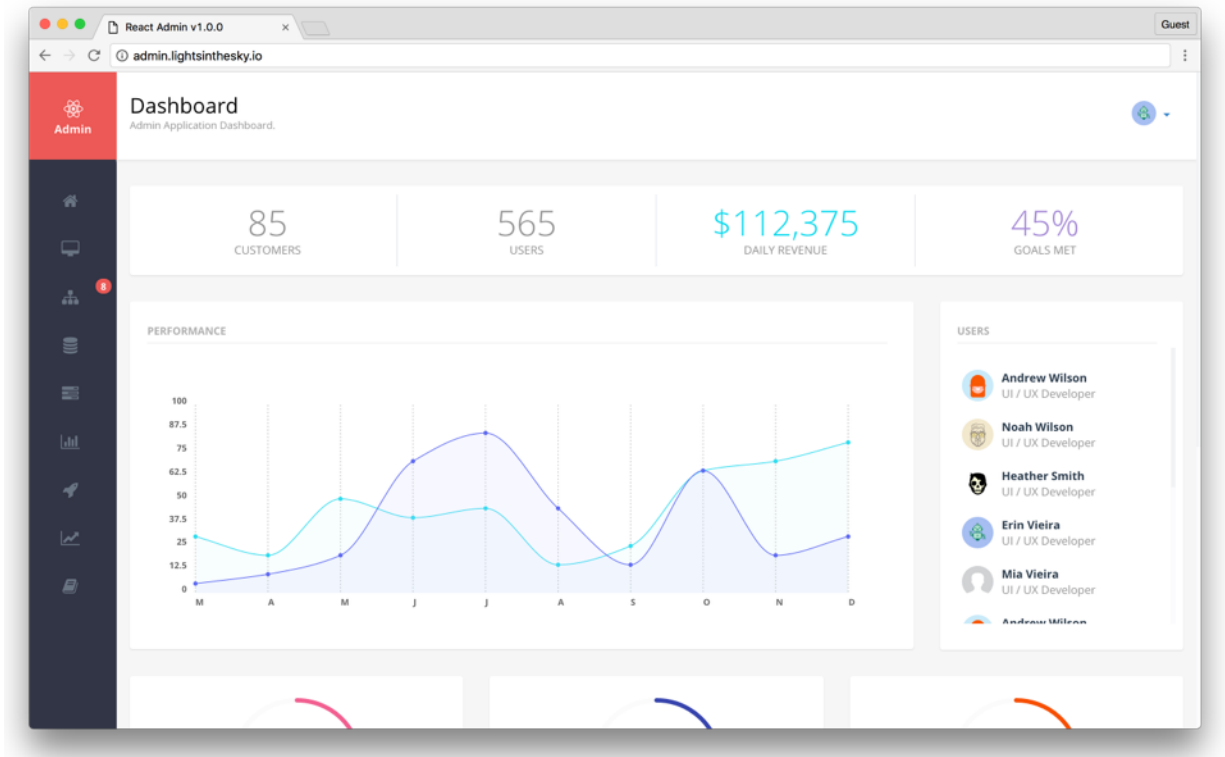


# React Admin

## *Documentation*



Version 1.0.0

# Getting Started

*This will help you get started with React Admin.*

React Admin is a javascript application that is powered by react, redux, react-router and a number of other libraries that will help you get up and running quickly. Once you open the zip file you can work within the project and expand and extend anything you wish.

1. Installing Node - The React Admin toolset requires node.js to run the commands to build , run and deploy the app. On Mac and Windows you can visit <https://nodejs.org/en/download/> and follow the instructions for your OS. On Linux you can install node using a package manager - visit this site <https://nodejs.org/en/download/package-manager/> for instructions for your OS.
2. Updating NPM - Once node.js has been installed you should update NPM to the latest version. At this point you will need to use the command shell (powershell on windows, terminal on OS X). Once you have the shell open navigate into the project folder and update NPM using : **\$npm install npm -g**
3. NPM Install - While in the command shell you can install the dependencies for React Admin by running : **\$sudo npm install**
4. Running the dev environment - While you are working on the project in development mode you can run **\$npm start** and the app will automatically build and reload whenever a source file is saved.
5. Compiling LESS to CSS - The CSS that dictates the look and feel of the project is generated by compiling LESS files. Compile your CSS from the LESS source **\$grunt**.

6. Building for production - Building for production is easy. Simply compile your CSS from the LESS source **\$grunt**. The build the javascript application **\$npm run build prod**.
7. Deployment example - Provided with the project is a sample AWS S3 upload config. If you want to deploy the application to S3 set up a bucket for static web hosting and enter the configuration in `aws-upload.conf.sample.js` and your credentials in `aws-credentials.sample.js` (removing sample from the file names). For instructions on static hosting on s3 visit <http://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html>

# Customizing Styles

*This will help you customize the look and feel of React Admin.*

1. Testing Grunt - If you are in the project folder (e.g. ReactAdmin) and run the command `$grunt` you should see the following:

```
Running "less:development" (less) task
>> 1 stylesheet created.

Running "watch" task
Waiting...
```

This implies the LESS has been compiled to css and is watching for new changes

2. Compiling with Grunt - If you wish to build a production version of the CSS, minified and compressed. Simply enter `$grunt build`
3. LESS structure - Like bootstrap, wholesale style changes can be accomplished through editing the variables.less file locations in `/dist/css/less/`. More granular style edits or extensions can be accomplished through editing the other .less files. Application (React-Admin) specific LESS styles are found in `react-admin.less`. See the LESS files for detailed documentation.

# Components

*This area will help with components*

1. React Components - The power of React is the ability for developers to create and composite components. To learn about React components visit here : <https://egghead.io/courses/react-fundamentals>
2. Edit components - Any component can be modified to meet a specific requirement by extending its functionality. This can be accomplished through changing the component internally, editing its style or changing its API (props) to provide new features.
3. Using redux in components and containers - Redux is a topic all in itself, we recommend visiting here : <https://egghead.io/courses/getting-started-with-redux> to learn how to use Redux. Briefly, you can connect any container or application in React to Redux using the **connect** method from **react-redux**.
4. A full manifest of components and their implementation can be seen here : <http://admin.lightsinthesky.io/docs#components>

# Pages and Apps

*This area will help with extending your app*

To build a new page or sub application you can review an existing page (e.g. /containers/Dashboard.js) or an existing app (e.g. /containers/Boards.js) to see how it all comes together. The following is an example of building a new container from scratch.

Create a new file Images.js in the /container folder. The template for this new container is :

```
import React, { Component } from 'react';
export default class Images extends Component {
  render() {
    return (
      <div></div>
    );
  }
}
```

This template will set up the container as a React component which renders an empty div.

Once the file is created we need to create a Route and Menu item such that the container can be accessible from the UI. First add the Route to /containers/Root.js

```
import Images from './Images';           // at the top
...
// in the Route declaration below
<Route path='/images' component={Images} pageName="Images"
pageDescription="A new Images container" />
```

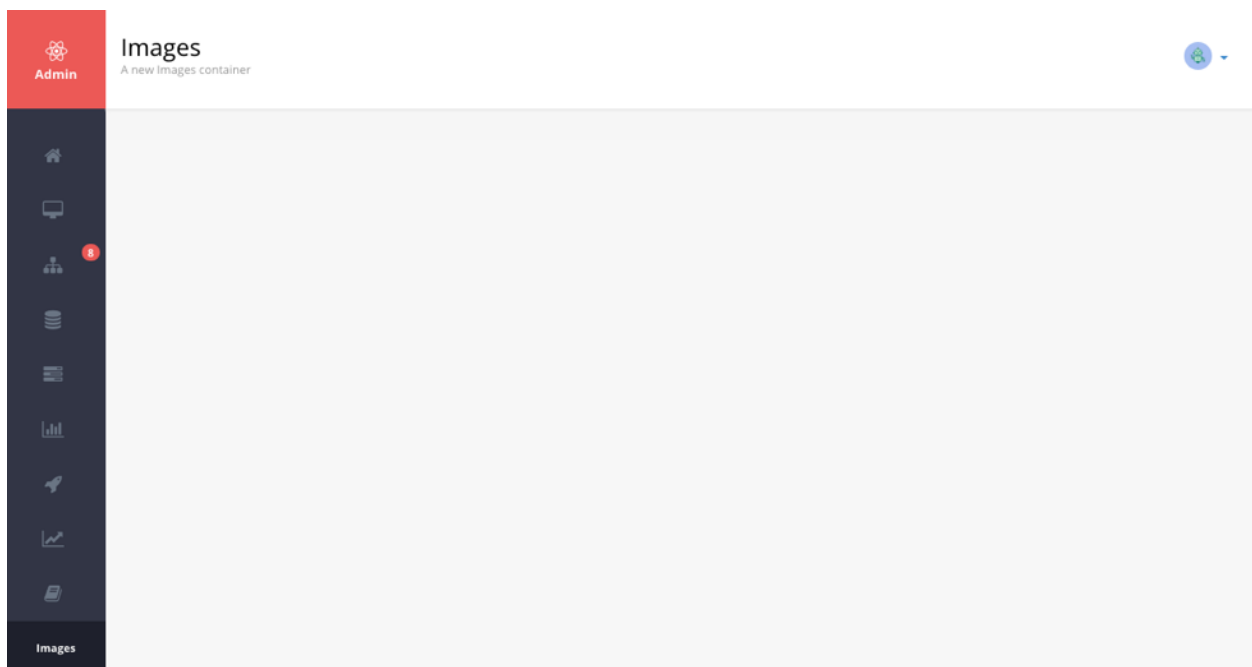
take a moment and ensure your development server is running by navigating to the project directory and running the command **\$npm start** and visiting <http://localhost:5050/> in your browser.

Next simply add a menu item to the menu for the new route by adding

```
<MenuItem
  link={'/images'}
  icon='fa-picture-o'
  color='bg-success'
  linkText='Images'
  currentPage={this.props.currentPage}
/>
```

at line 89 of /containers/Menu.js

Reloading your application in your browser should immediately show you your new menu item , if it doesn't make sure the server is running.



Now that there is a new container in your application you can add components and link the container to redux.

To connect the container to Redux update the source to look like this:

```
import React, { Component } from 'react';
import { connect } from 'react-redux'; // add connect

class Images extends Component {
  render() {
    return (
      <div>{this.props.user.email}</div> // from redux
    );
  }
}

function mapStateToProps(state) {
  return {
    token: state.app.token,
    user: state.user
  };
}

export default connect(mapStateToProps)(Images);
```

This will instantly provide the token and user object to the container via its props. This container now simply becomes a composition of data from Redux and react components. You read data through the props and mutate data by dispatching actions e.g.

```
deletePin(id) {
  let {token, dispatch} = this.props;
  dispatch(pinActions.delete(token, {_id:id}));
}
```



## Outlets and API

Outlets are CRUD based Redux store outlets (hence the name). They contain action consts, action creators and a reducer. Outlets also contain built in logic for fetching requirements. A local outlet is used for demoing the functionality of an outlet locally (with no persistence). In the case of React-Admin you can see that the initial state contains our demo data and our local outlet simply mutates our state without applying the changes to any backend. To connect to a real backend you must implement a ReduxOutlet, specifically you must write new action creators for fetch, fetchOne, create, update and delete - an example has been included to assist you. The only rule is that the action creators must return actions that implement promise (async) middleware.

React Admin will act as a the tool you use to start all your projects from prototyping through to production features.

## Credits

name	repo	license
axios	<a href="https://github.com/mzabriskie/axios">https://github.com/mzabriskie/axios</a>	<a href="https://github.com/mzabriskie/axios/blob/master/LICENSE">https://github.com/mzabriskie/axios/blob/master/LICENSE</a>
babel.js	<a href="https://github.com/babel/babel">https://github.com/babel/babel</a>	MIT
grunt	<a href="https://github.com/gruntjs/grunt">https://github.com/gruntjs/grunt</a>	<a href="https://github.com/gruntjs/grunt/blob/master/LICENSE">https://github.com/gruntjs/grunt/blob/master/LICENSE</a>
codemirror	<a href="https://github.com/codemirror/CodeMirror">https://github.com/codemirror/CodeMirror</a>	<a href="https://github.com/codemirror/CodeMirror/blob/master/LICENSE">https://github.com/codemirror/CodeMirror/blob/master/LICENSE</a>

name	repo	license
react-codemirror	<a href="https://github.com/JedWatson/react-codemirror">https://github.com/JedWatson/react-codemirror</a>	MIT
cookies-js	<a href="https://github.com/js-cookie/js-cookie">https://github.com/js-cookie/js-cookie</a>	MIT
draft-js	<a href="https://github.com/facebook/draft-js">https://github.com/facebook/draft-js</a>	BSD
draft-js-plugins-editor	<a href="https://github.com/draft-js-plugins/draft-js-plugins">https://github.com/draft-js-plugins/draft-js-plugins</a>	<a href="https://github.com/draft-js-plugins/draft-js-plugins/blob/master/LICENSE">https://github.com/draft-js-plugins/draft-js-plugins/blob/master/LICENSE</a>
googleapis	<a href="https://github.com/google/google-api-nodejs-client">https://github.com/google/google-api-nodejs-client</a>	Apache 2.0
moment	<a href="https://github.com/moment/moment">https://github.com/moment/moment</a>	<a href="https://github.com/moment/moment/blob/develop/LICENSE">https://github.com/moment/moment/blob/develop/LICENSE</a>
rc-slider	<a href="https://github.com/react-component/slider">https://github.com/react-component/slider</a>	MIT
react	<a href="https://github.com/facebook/react">https://github.com/facebook/react</a>	<a href="https://github.com/facebook/react/blob/master/LICENSE">https://github.com/facebook/react/blob/master/LICENSE</a>
react-chartist	<a href="https://github.com/fraserxu/react-chartist">https://github.com/fraserxu/react-chartist</a>	MIT
react-color	<a href="https://github.com/casesandberg/react-color/">https://github.com/casesandberg/react-color/</a>	MIT
react-datepicker	<a href="https://github.com/Hacker0x01/react-datepicker">https://github.com/Hacker0x01/react-datepicker</a>	MIT
bootstrap	<a href="https://github.com/twbs/bootstrap">https://github.com/twbs/bootstrap</a>	MIT
react-google-maps	<a href="https://github.com/tomchentw/react-google-maps">https://github.com/tomchentw/react-google-maps</a>	MIT
redux	<a href="https://github.com/reactjs/redux/">https://github.com/reactjs/redux/</a>	MIT
react-redux	<a href="https://github.com/reactjs/react-redux">https://github.com/reactjs/react-redux</a>	MIT
react-router	<a href="https://github.com/reactjs/react-router">https://github.com/reactjs/react-router</a>	MIT

name	repo	license
react-router-redux	<a href="https://github.com/reactjs/react-router-redux">https://github.com/reactjs/react-router-redux</a>	<a href="https://github.com/reactjs/react-router-redux/blob/master/LICENSE">https://github.com/reactjs/react-router-redux/blob/master/LICENSE</a>
webpack	<a href="https://github.com/webpack/webpack">https://github.com/webpack/webpack</a>	MIT
webpack-dev-server	<a href="https://github.com/webpack/webpack-dev-server">https://github.com/webpack/webpack-dev-server</a>	MIT
react-hot-loader	<a href="https://github.com/gaearon/react-hot-loader">https://github.com/gaearon/react-hot-loader</a>	MIT
font-awesome	<a href="https://github.com/FortAwesome/Font-Awesome">https://github.com/FortAwesome/Font-Awesome</a>	<a href="http://scripts.sil.org/OFL">http://scripts.sil.org/OFL</a>
material design icons	<a href="https://github.com/google/material-design-icons">https://github.com/google/material-design-icons</a>	<a href="https://github.com/google/material-design-icons/blob/master/LICENSE">https://github.com/google/material-design-icons/blob/master/LICENSE</a>