

React Native Notes

REALM

Realm is a library for saving data that is persistent on reloads and restarts. Realm Database makes it easy to save and manipulate data in a database for your React Native app. Other ways to handle data in such a way is using AsyncStorage or SQLite. Realm JavaScript enables you to efficiently write your app's model layer in a safe, persisted and fast way. It's designed to work with React Native and Node.js.

Installation

```
npm install --save realm
```

Next, link your project to the realm native module.

```
react-native link realm
```

Warning for Android : Depending on the version, react-native link may generate an invalid configuration, updating Gradle correctly (android/settings.gradle and android/app/build.gradle) but failing to add the Realm module. Confirm that react-native link has added the Realm module; if it has not, link manually to the library with the following steps:

Add the following lines to android/settings.gradle:

```
include ':realm'  
  
project(':realm').projectDir = new File(rootProject.projectDir, '../node_modules/  
realm/android')
```

Add Realm to dependencies in android/app/build.gradle:

```
// When using Android Gradle plugin 3.0 or higher  
dependencies {
```

React Native Notes

```
implementation project(':realm')  
  
}  
  
// When using Android Gradle plugin lower than 3.0  
  
dependencies {  
  
    compile project(':realm')  
  
}
```

Add the import and link the package in MainApplication.java:

```
import io.realm.react.RealmReactPackage; // add this import  
  
public class MainApplication extends Application implements ReactApplication {  
  
    @Override  
  
    protected List<ReactPackage> getPackages() {  
  
        return Arrays.<ReactPackage>asList(  
  
            new MainReactPackage(),  
  
            new RealmReactPackage() // add this line  
  
21.         );  
  
22.     }  
  
}
```

You're now ready to go. To see Realm in action, replace the definition of class `<project-name>` with the following in App.js:

```
const Realm = require('realm');  
  
class <project-name> extends Component {  
  
    constructor(props) {  
  
        super(props);  
  
        this.state = { realm: null };  
  
    }  
  
}
```

React Native Notes

```
}

componentDidMount() {
  Realm.open({
    schema: [{name: 'Dog', properties: {name: 'string'}}]
  }).then(realms => {
    realm.write(() => {
      realm.create('Dog', {name: 'Rex'});
    });
    this.setState({ realm });
  });
}

componentWillUnmount() {
  // Close the realm if there is one open.
  const {realm} = this.state;
  if (realm !== null && !realm.isClosed) {
    realm.close();
  }
}

render() {
  const info = this.state.realm
    ? 'Number of dogs in this Realm: ' + this.state.realm.objects('Dog').length
    : 'Loading...';
  return (
```

React Native Notes

```
<View style={styles.container}>  
  <Text style={styles.welcome}>  
    {info}  
  </Text>  
</View>  
);  
}  
}
```

Code

```
const Realm = require('realm');  
  
// Define your models and their properties  
  
const CarSchema = {  
  name: 'Car',  
  properties: {  
    make: 'string',  
    model: 'string',  
    miles: {type: 'int', default: 0},  
  }  
};  
  
const PersonSchema = {  
  name: 'Person',  
  properties: {  
    name: 'string',
```

React Native Notes

```
    birthday: 'date',  
    cars: 'Car[]', // a list of Cars  
    picture: 'data?' // optional property  
  }  
};
```

```
Realm.open({schema: [CarSchema, PersonSchema]})  
  .then(real => {  
    // Create Realm objects and write to local storage  
    realm.write(() => {  
      const myCar = realm.create('Car', {  
        make: 'Honda',  
        model: 'Civic',  
        miles: 1000,  
      });  
      myCar.miles += 20; // Update a property value  
    });  
  
    // Query Realm for all cars with a high mileage  
    const cars = realm.objects('Car').filtered('miles > 1000');  
  
    // Will return a Results object with our 1 car  
    cars.length // => 1
```

React Native Notes

```
// Add another car

realm.write(() => {

  const myCar = realm.create('Car', {

    make: 'Ford',

    model: 'Focus',

    miles: 2000,

  });

});

// Query results are updated in realtime

cars.length // => 2

// Remember to close the realm when finished.

realm.close();

})

.catch(error => {

  console.log(error);

});
```