**React Native FlatList Pagination to Load More Data dynamically – Infinite List**

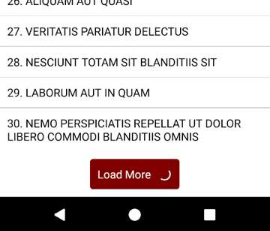
Here is an example of React Native FlatList Pagination to Load More Data dynamically – Infinite List. In this example, we will make a FlatList in which we will load the data in the form of pagination on a Click of a button. You can also visit Example of Infinite Loading Listview in React Native using FlatList for the infinite loading list view without load more button.

Let’s take a case where you can simply imagine what if Facebook loads all the Facebook posts from the Facebook database on your Facebook wall? Doesn’t it sound nasty? It will take hundreds of hours even more than that so what is the best way to load the data? Simply paging. When you open Facebook they just show you 10-20 post and while you scroll the Facebook they load the next 10-20 posts in the background. That is what the paging is.

**In an example,**

1. We are loading the first 10 posts from the web API call in *componentDidMount*.
2. We have added load more button on the footer of the list.
3. While clicking on the Load More Button we call the Web API again to get the next 10 posts.

We are using a variable offset to manage the index on upcoming Data. We will increase the offset by 1 after the successful call of the web API so that when we call the web API next time we will get the next dataset. So let’s get started with the example.

After Clicking on Load More It will load the data with the loader in Button

//This is an example of React Native

//FlatList Pagination to Load More Data dynamically - Infinite List

import React, { Component } from 'react';

//import react in our code.

import {

View,

Text,

TouchableOpacity,

StyleSheet,

FlatList,

ActivityIndicator,

} from 'react-native';

//import all the components we are going to use.

export default class App extends Component {

constructor() {

super();

this.state = {

loading: true,

//Loading state used while loading the data for the first time

serverData: [],

//Data Source for the FlatList

fetching\_from\_server: false,

//Loading state used while loading more data

};

this.offset = 1;

//Index of the offset to load from web API

}

componentDidMount() {

fetch('https://aboutreact.herokuapp.com/getpost.php?offset=' + this.offset)

//Sending the currect offset with get request

.then(response => response.json())

.then(responseJson => {

//Successful response from the API Call

this.offset = this.offset + 1;

//After the response increasing the offset for the next API call.

this.setState({

serverData: [...this.state.serverData, ...responseJson.results],

//adding the new data with old one available in Data Source of the List

loading: false,

//updating the loading state to false

});

})

.catch(error => {

console.error(error);

});

}

loadMoreData = () => {

//On click of Load More button We will call the web API again

this.setState({ fetching\_from\_server: true }, () => {

fetch('https://aboutreact.herokuapp.com/getpost.php?offset=' + this.offset)

//Sending the currect offset with get request

.then(response => response.json())

.then(responseJson => {

//Successful response from the API Call

this.offset = this.offset + 1;

//After the response increasing the offset for the next API call.

this.setState({

serverData: [...this.state.serverData, ...responseJson.results],

//adding the new data with old one available

fetching\_from\_server: false,

//updating the loading state to false

});

})

.catch(error => {

console.error(error);

});

});

};

renderFooter() {

return (

//Footer View with Load More button

<View style={styles.footer}>

<TouchableOpacity

activeOpacity={0.9}

onPress={this.loadMoreData}

//On Click of button calling loadMoreData function to load more data

style={styles.loadMoreBtn}>

<Text style={styles.btnText}>Load More</Text>

{this.state.fetching\_from\_server ? (

<ActivityIndicator color="white" style={{ marginLeft: 8 }} />

) : null}

</TouchableOpacity>

</View>

);

}

render() {

return (

<View style={styles.container}>

{this.state.loading ? (

<ActivityIndicator size="large" />

) : (

<FlatList

style={{ width: '100%' }}

keyExtractor={(item, index) => index}

data={this.state.serverData}

renderItem={({ item, index }) => (

<View style={styles.item}>

<Text style={styles.text}>

{item.id}

{'.'}

{item.title.toUpperCase()}

</Text>

</View>

)}

ItemSeparatorComponent={() => <View style={styles.separator} />}

ListFooterComponent={this.renderFooter.bind(this)}

//Adding Load More button as footer component

/>

)}

</View>

);

}

}

const styles = StyleSheet.create({

container: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

paddingTop: 30,

},

item: {

padding: 10,

},

separator: {

height: 0.5,

backgroundColor: 'rgba(0,0,0,0.4)',

},

text: {

fontSize: 15,

color: 'black',

},

footer: {

padding: 10,

justifyContent: 'center',

alignItems: 'center',

flexDirection: 'row',

},

loadMoreBtn: {

padding: 10,

backgroundColor: '#800000',

borderRadius: 4,

flexDirection: 'row',

justifyContent: 'center',

alignItems: 'center',

},

btnText: {

color: 'white',

fontSize: 15,

textAlign: 'center',

},

});