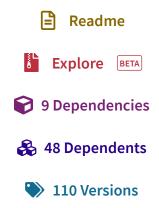


react-native-gifted-chat Is

0.16.3 • Public • Published a year ago





Gifted Chat

The most complete chat UI for React Native & Web



Sponsor



Coding Bootcamp in Paris co-founded by Farid Safi

Click to learn more



Scalable chat API/Server written in Go

API Tour | React Native Gifted tutorial



Delivering Personalized Healthcare

Find out more

Please vote

GiftedChat depends on other packages and some needs a boost, please vote for PRs will improve it, thanks:

- https://github.com/taskrabbit/react-native-parsed-text/pull/58
- https://github.com/watadarkstar/react-native-typing-animation/issues/18

Features

- * react-native-web able (since 0.10.0) web configuration
- Write with **TypeScript** (since 0.8.0)
- Fully customizable components
- Composer actions (to attach photos, etc.)
- Load earlier messages
- Copy messages to clipboard
- Touchable links using react-native-parsed-text
- Avatar as user's initials
- Localized dates
- Multi-line TextInput
- InputToolbar avoiding keyboard
- Redux support
- System message
- Quick Reply messages (bot)
- Typying indicatior react-native-typing-animation

Dependency

- Use version 0.2.x for RN >= 0.44.0
- Use version 0.1.x for RN $\geq 0.40.0$
- Use version 0.0.10 for RN < 0.40.0

Installation

- Using npm: npm install react-native-gifted-chat --save
- Using Yarn: yarn add react-native-gifted-chat

react-native-video and expo-av

- Both dependencies are removed since 0.11.0.
- You still be able to provide a video but you need to provide renderMessageVideo prop.

You have a question?

- 1. Please check this readme and may find a response
- 2. Please ask on StackOverflow first: https://stackoverflow.com/questions/tagged/react-native-gifted-chat
- 3. Find response on existing issues
- 4. Try to keep issues for issues

Example

```
import React, { useState, useCallback, useEffect } from 'rea
import { GiftedChat } from 'react-native-gifted-chat'
export function Example() {
  const [messages, setMessages] = useState([]);
  useEffect(() => {
    setMessages([
      {
        id: 1,
        text: 'Hello developer',
        createdAt: new Date(),
        user: {
          id: 2,
          name: 'React Native',
          avatar: 'https://placeimg.com/140/140/any',
        },
      },
    ])
  }, [])
```

Advanced example

See App.tsx for a working demo!

"Slack" example

See the files in **example-slack-message** for an example of how to override the default UI to make something that looks more like Slack -- with usernames displayed and all messages on the left.

Message object

e.g. Chat Message

```
export interface IMessage {
   _id: string | number
   text: string
   createdAt: Date | number
   user: User
```

{

```
image?: string
  video?: string
  audio?: string
  system?: boolean
  sent?: boolean
  received?: boolean
  pending?: boolean
  quickReplies?: QuickReplies
}
{
  id: 1,
  text: 'My message',
  createdAt: new Date(Date.UTC(2016, 5, 11, 17, 20, 0)),
  user: {
   _id: 2,
    name: 'React Native',
    avatar: 'https://facebook.github.io/react/img/logo og.pr
  },
  image: 'https://facebook.github.io/react/img/logo og.png',
  // You can also add a video prop:
  video: 'http://commondatastorage.googleapis.com/gtv-videos
  // Mark the message as sent, using one tick
  sent: true,
  // Mark the message as received, using two tick
  received: true,
  // Mark the message as pending with a clock loader
  pending: true,
  // Any additional custom parameters are passed through
}
 e.g. System Message
```

```
_id: 1,
   text: 'This is a system message',
   createdAt: new Date(Date.UTC(2016, 5, 11, 17, 20, 0)),
   system: true,
   // Any additional custom parameters are passed through
 }
  e.g. Chat Message with Quick Reply options
See PR #1211
 interface Reply {
   title: string
   value: string
   messageId?: any
 }
 interface QuickReplies {
   type: 'radio' | 'checkbox'
   values: Reply[]
   keepIt?: boolean
 }
   {
     id: 1,
     text: 'This is a quick reply. Do you love Gifted Chat? (
     createdAt: new Date(),
     quickReplies: {
       type: 'radio', // or 'checkbox',
       keepIt: true,
       values: [
         {
           title: 'ca Yes',
           value: 'yes',
```

```
},
      {
        title: ' Yes, let me show you with a picture!',
        value: 'yes_picture',
      },
      {
        title: 'Nope. What?',
        value: 'no',
      },
    ],
  },
  user: {
    _id: 2,
    name: 'React Native',
  },
},
{
 id: 2,
  text: 'This is a quick reply. Do you love Gifted Chat? (
  createdAt: new Date(),
  quickReplies: {
    type: 'checkbox', // or 'radio',
    values: [
      {
        title: 'Yes',
        value: 'yes',
      },
      {
        title: 'Yes, let me show you with a picture!',
        value: 'yes_picture',
      },
      {
        title: 'Nope. What?',
        value: 'no',
      },
```

```
],
},
user: {
    _id: 2,
    name: 'React Native',
},
}
```

Props

- messages (Array) Messages to display
- **isTyping** (Bool) Typing Indicator state; default false. If you use renderFooter it will override this.
- **text** (String) Input text; default is undefined, but if specified, it will override GiftedChat's internal state (e.g. for redux; see notes below)
- placeholder (String) Placeholder when text is empty; default is 'Type a message...'
- **messageIdGenerator** (Function) Generate an id for new messages. Defaults to UUID v4, generated by uuid
- **user** (Object) User sending the messages: { _id, name, avatar }
- onSend (Function) Callback when sending a message
- **alwaysShowSend** (Bool) Always show send button in input text composer; default false, show only when text input is not empty
- **locale** (String) Locale to localize the dates. You need first to import the locale you need (ie. require('dayjs/locale/de') or import 'dayjs/locale/fr')
- timeFormat (String) Format to use for rendering times; default is 'LT'
- dateFormat (String) Format to use for rendering dates; default is 'll'
- **loadEarlier** (Bool) Enables the "load earlier messages" button, required for infiniteScroll
- **isKeyboardInternallyHandled** (Bool) Determine whether to handle keyboard awareness inside the plugin. If you have your own keyboard handling outside the plugin set this to false; default is true
- onLoadEarlier (Function) Callback when loading earlier messages
- **isLoadingEarlier** (Bool) Display an ActivityIndicator when loading earlier messages

- renderLoading (Function) Render a loading view when initializing
- renderLoadEarlier (Function) Custom "Load earlier messages" button
- **renderAvatar** (Function) Custom message avatar; set to null to not render any avatar for the message
- **showUserAvatar** (Bool) Whether to render an avatar for the current user; default is false, only show avatars for other users
- **showAvatarForEveryMessage** (Bool) When false, avatars will only be displayed when a consecutive message is from the same user on the same day; default is false
- onPressAvatar (Function(user)) Callback when a message avatar is tapped
- onLongPressAvatar (Function(user)) Callback when a message avatar is long-pressed
- **renderAvatarOnTop** (Bool) Render the message avatar at the top of consecutive messages, rather than the bottom; default is false
- renderBubble (Function) Custom message bubble
- renderTicks (Function(message)) Custom ticks indicator to display message status
- renderSystemMessage (Function) Custom system message
- **onLongPress** (Function(context, message)) Callback when a message bubble is long-pressed; default is to show an ActionSheet with "Copy Text" (see example using showActionSheetWithOptions())
- inverted (Bool) Reverses display order of messages; default is true
- **renderUsernameOnMessage** (Bool) Indicate whether to show the user's username inside the message bubble; default is false
- renderMessage (Function) Custom message container
- renderMessageText (Function) Custom message text
- renderMessageImage (Function) Custom message image
- renderMessageVideo (Function) Custom message video
- **imageProps** (Object) Extra props to be passed to the <**Image**> component created by the default renderMessageImage
- **videoProps** (Object) Extra props to be passed to the video component created by the required renderMessageVideo
- **lightboxProps** (Object) Extra props to be passed to the MessageImage 's Lightbox
- **isCustomViewBottom** (Bool) Determine whether renderCustomView is displayed before or after the text, image and video views; default is false

- renderCustomView (Function) Custom view inside the bubble
- renderDay (Function) Custom day above a message
- renderTime (Function) Custom time inside a message
- renderFooter (Function) Custom footer component on the ListView, e.g.
 'User is typing...'; see example/App.js for an example. Overrides default typing indicator that triggers when isTyping is true.
- **renderChatEmpty** (Function) Custom component to render in the ListView when messages are empty
- **renderChatFooter** (Function) Custom component to render below the MessageContainer (separate from the ListView)
- renderInputToolbar (Function) Custom message composer container
- **renderComposer** (Function) Custom text input message composer
- **renderActions** (Function) Custom action button on the left of the message composer
- **renderSend** (Function) Custom send button; you can pass children to the original Send component quite easily, for example, to use a custom icon (example)
- **renderAccessory** (Function) Custom second line of actions below the message composer
- **onPressActionButton** (Function) Callback when the Action button is pressed (if set, the default actionSheet will not be used)
- **bottomOffset** (*Integer*) Distance of the chat from the bottom of the screen (e.g. useful if you display a tab bar)
- minInputToolbarHeight (Integer) Minimum height of the input toolbar; default is 44
- listViewProps (Object) Extra props to be passed to the messages
 ListView>; some props can't be overridden, see the code in
 MessageContainer.render() for details
- textInputProps (Object) Extra props to be passed to the <TextInput>
- **textInputStyle** (Object) Custom style to be passed to the <TextInput>
- **multiline** (Bool) Indicates whether to allow the <TextInput> to be multiple lines or not; default true.
- **keyboardShouldPersistTaps** (Enum) Determines whether the keyboard should stay visible after a tap; see **<ScrollView>** docs
- onInputTextChanged (Function) Callback when the input text changes
- maxInputLength (Integer) Max message composer TextInput length
- parsePatterns (Function) Custom parse patterns for react-native-parsed-text

used to linking message content (like URLs and phone numbers), e.g.:

- **extraData** (Object) Extra props for re-rendering FlatList on demand. This will be useful for rendering footer etc.
- minComposerHeight (Object) Custom min-height of the composer.
- maxComposerHeight (Object) Custom max height of the composer.
- **scrollToBottom** (Bool) Enables the scroll to bottom Component (Default is false)
- **scrollToBottomComponent** (Function) Custom Scroll To Bottom Component container
- **scrollToBottomOffset** (*Integer*) Custom Height Offset upon which to begin showing Scroll To Bottom Component (Default is 200)
- **scrollToBottomStyle** *(Object)* Custom style for Bottom Component container
- **alignTop** (Boolean) Controls whether or not the message bubbles appear at the top of the chat (Default is false bubbles align to bottom)
- onQuickReply (Function) Callback when sending a quick reply (to backend server)
- renderQuickReplies (Function) Custom all quick reply view
- quickReplyStyle (StyleProp) Custom quick reply view style
- renderQuickReplySend (Function) Custom quick reply send view
- **shouldUpdateMessage** (*Function*) Lets the message component know when to update outside of normal cases.
- **infiniteScroll** (Bool) infinite scroll up when reach the top of messages container, automatically call onLoadEarlier function if exist (not yet supported for the web). You need to add loadEarlier prop too.

Imperative methods

• focusTextInput() - Open the keyboard and focus the text input box

Notes for Redux

The messages prop should work out-of-the-box with Redux. In most cases, this is all you need.

If you decide to specify a text prop, GiftedChat will no longer manage its own internal text state and will defer entirely to your prop. This is great for using a tool like Redux, but there's one extra step you'll need to take: simply implement onInputTextChanged to receive typing events and reset events (e.g. to clear the text onSend):

```
<GiftedChat
  text={customText}
  onInputTextChanged={text => this.setCustomText(text)}
  /* ... */
/>
```

Notes for Android

If you are using Create React Native App / Expo, no Android specific installation steps are required -- you can skip this section. Otherwise, we recommend modifying your project configuration as follows.

Make sure you have android:windowSoftInputMode="adjustResize" in your
 AndroidManifest.xml:

```
<activity
  android:name=".MainActivity"
  android:label="@string/app_name"
  android:windowSoftInputMode="adjustResize"
  android:configChanges="keyboard|keyboardHidden|orientati</pre>
```

- For **Expo**, there are at least 2 solutions to fix it:
 - Append KeyboardAvoidingView after GiftedChat. This should only be

done for Android, as KeyboardAvoidingView may conflict with the iOS keyboard avoidance already built into GiftedChat, e.g.:

If you use React Navigation, additional handling may be required to account for navigation headers and tabs. KeyboardAvoidingView 's keyboardVerticalOffset property can be set to the height of the navigation header and tabBarOptions.keyboardHidesTabBar can be set to keep the tab bar from being shown when the keyboard is up. Due to a bug with calculating height on Android phones with notches, KeyboardAvoidingView is recommended over other solutions that involve calculating the height of the window.

- adding an opaque background status bar on app.json (even though android:windowSoftInputMode="adjustResize" is set internally on Expo's Android apps, the translucent status bar causes it not to work): https://docs.expo.io/versions/latest/guides/configuration.html#androidstatusbar
- If you plan to use GiftedChat inside a Modal, see #200.

Notes for local development

Native

- 1. Install yarn add -g expo-cli
- 2. expo start

react-native-web

With expo

```
1. Install yarn add -g expo-cli
```

2. expo start -w

Upgrade snack version

With create-react-app

```
    yarn add -D react-app-rewired

2. touch config-overrides.js
module.exports = function override(config, env) {
  config.module.rules.push({
    test: /\.js$/,
    exclude: /node_modules[/\\](?!react-native-gifted-chat|r
    use: {
      loader: 'babel-loader',
      options: {
        babelrc: false,
        configFile: false,
        presets: [
          ['@babel/preset-env', { useBuiltIns: 'usage' }],
          '@babel/preset-react',
        ],
        plugins: ['@babel/plugin-proposal-class-properties']
      },
    },
  })
  return config
}
```

You will find an example and a **web demo** here: **xcarpentier/gifted-chat-web-demo**

Another example with Gatsby: xcarpentier/clean-archi-boilerplate

Questions

- How can I set Bubble color for each user?
- How can I pass style props to InputToolbar design and customize its color and other styles properties?
- How can I change the color of the message box?
- Is there a way to manually dismiss the keyboard?
- I want to implement a popover that pops right after clicking on a specific avatar, what is the best implementation in this case and how?
- Why TextInput is hidden on Android?
- How to use renderLoading?
- Can I use MySql to save the message?

License

MIT

Author

Feel free to ask me questions on Twitter @FaridSafi! or @xcapetir!

Contributors

- Kevin Cooper cooperka
- Kfir Golan kfiroo
- Bruno Cascio brunocascio
- Xavier Carpentier xcarpentier
- more

Hire an expert!

Looking for a ReactNative freelance expert with more than 12 years of experience? Contact Xavier from his website!

Keywords

Install

android ios react-native react react-component messenger

> npm i react-native-gifted-chat

message chat

Repository

github.com/FaridSafi/react-native-gifted-chat

Homepage

𝚱 github.com/FaridSafi/react-native-gifted-chat#readme

★ Weekly Downloads

24,687

Version License

0.16.3 MIT

Unpacked Size Total Files

287 kB 103

Issues **Pull Requests**

78 22

Last publish

a year ago

Collaborators











>_ Try on RunKit

Report malware



Support

Help

Community

Advisories

Status

Contact npm

Company

About

Blog

Press

Terms & Policies

Policies

Terms of Use

Code of Conduct

Privacy