leecade / react-native-swiper

The best Swiper component for React Native.

#swipe #react-native #react

309 commits | 26 branches | 22 releases | 56 contributors | MIT

Branch: master ▾ | New pull request | Create new file | Upload files | Find File | Clone or download ▾

arribbar Merge pull request #897 from leecade/release1.5.14 … Latest commit e4dbde6 on 26 Nov 2018

.github	update issue template to get more info	2 years ago
examples	#684 to fix onScrollBeginDrag not being called on Android	8 months ago
src	resolve styling conflict	5 months ago
.editorconfig	first ci	4 years ago
.gitignore	Ignore dist/ directory.	3 years ago
.npmignore	first ci	4 years ago
CHANGELOG.md	Release 1.5.14	5 months ago
LICENSE	Initial commit	4 years ago
README.md	Merge pull request #838 from thealiilman/patch-1	7 months ago
ROADMAP.md	bump versiom 1.5.2	3 years ago
index.d.ts	Correcting type `autoplayTimeout: number` prop	a year ago
index.js	Improve develop workflow prepare for 1.5.0	3 years ago
index.js.flow	Add flow definitions	7 months ago
package.json	Release 1.5.14	5 months ago
yarn.lock	bump version 1.5.4	2 years ago

README.md

> Hi there:

[头条财经前端团队急招] 了解团队 / 投简历请联系:

微信: 103024979 / 邮箱: leecade@163.com

「全新团队, 全新产品, 全新技术栈, 不设技术边界, 不设管理, 不装」

要求: 参考头条面试标准 (重度: `Node / RN / Vue / React / 微服务 等技术向`)

字节跳动 ByteDance

react-native-swiper

The best Swiper component for React Native.

code style standard | npm v1.5.14 | downloads 141k/m | issue stats no longer available | issue stats no longer available | chat on gitter

react-native-swiper

Roadmap

> see: ROADMAP.md

Changelogs

- **[1.5.6]**
  - Fix #16, #36, #371, #410, #411, #422, #468 Fix landscape orientation auto resize! (thanks @ahmed3mar, @timmywil)
  - Add containerStyle prop to customize the view container.
- [1.5.5]
  - Update: using PropTypes from prop-types and Change View.propTypes to ViewPropTypes
- [1.5.4]
  - Added easily accessible pagination point manipulation: use `dotColor` / `activeDotColor` and `dotStyle` / `activeDotStyle` (thanks @denizs)
  - Added scrollEnabled prop to documentation (thanks @ibandominguez)
- [1.5.3]
  - Add loadMinimalLoader prop to customize `<ActivityIndicator />` (thanks @Exilz)
  - Disable autoplay timer when prop changes to false (thanks @dizlexik)
  - Special thanks to @hypatiah for fixed some grammatical errors in README
- [1.5.2]
  - Add yarn lock
  - Fix jitter when quickly swiping back and forth between pages (iOS) (thanks @nemophrost)
  - The first webview always reloaded when injecting the rest of the children (thanks @eosterberg)

> see more: CHANGELOG.md

Show Cases

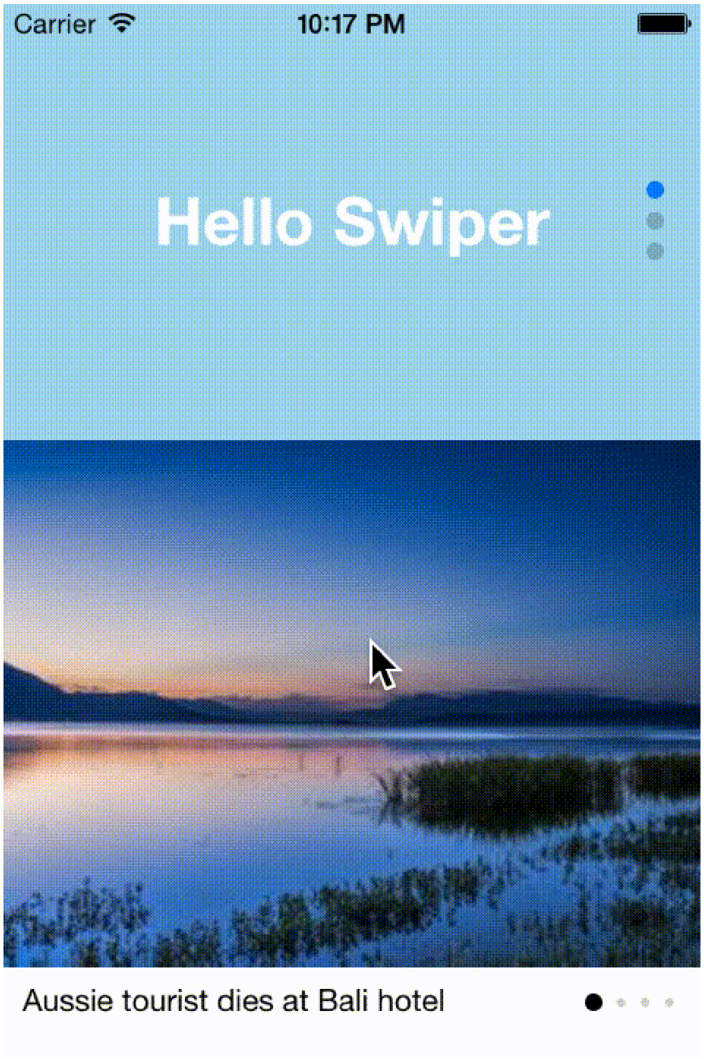
> Try these cases by yourself very easy, Just open `examples/ios/swiper.xcodeproj` in Xcode, then press `Cmd + R`; you may edit `examples/index.ios.js` for switch cases.

examples/components/Basic

The image you are requesting does not exist or is no longer available.

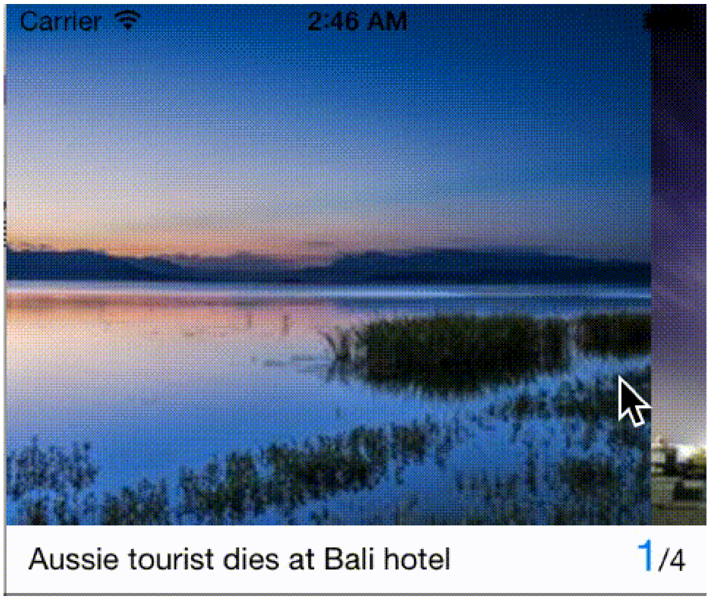
imgur.com

examples/components/Swiper



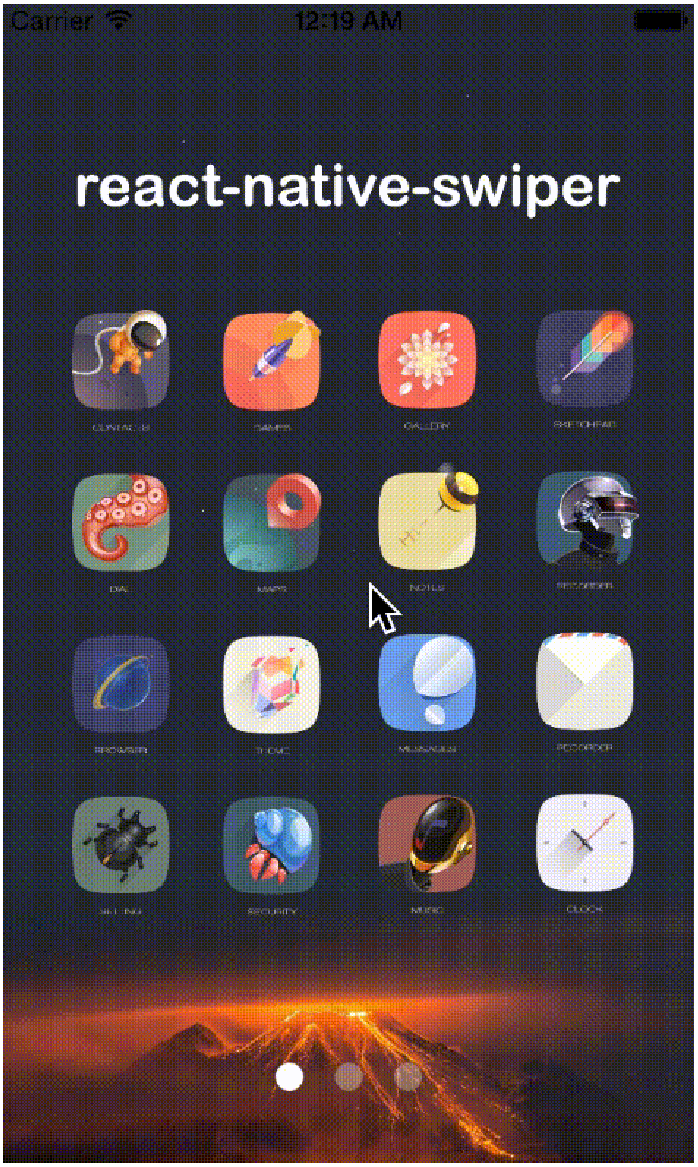
Aussie tourist dies at Bali hotel

examples/components/SwiperNumber

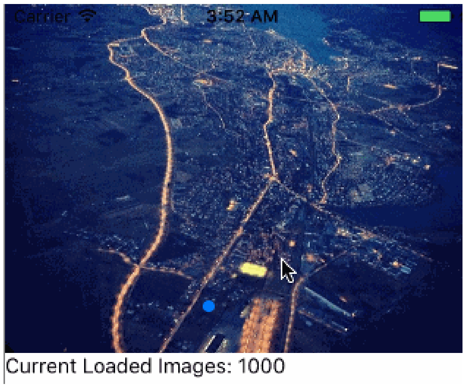


Aussie tourist dies at Bali hotel

examples/components/Phone



examples/components/LoadMinimal



Current Loaded Images: 1000

Getting Started

Installation

```
$ npm i react-native-swiper --save
```

Basic Usage

- Install `react-native` first

```
$ npm i react-native-cli -g
```

- Initialization of a react-native project

```
$ react-native init myproject
```

- Then, edit `myproject/index.ios.js` , like this:

```
import React, { Component } from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  View
} from 'react-native';

import Swiper from 'react-native-swiper';
```

```
const styles = StyleSheet.create({
  wrapper: {
  },
  slide1: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#9DD6EB',
  },
  slide2: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#97CAE5',
  },
  slide3: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#92BBD9',
  },
  text: {
    color: '#fff',
    fontSize: 30,
    fontWeight: 'bold',
  }
})

export default class Swiper extends Component {
  render(){
    return (
      <Swiper style={styles.wrapper} showsButtons={true}>
        <View style={styles.slide1}>
          <Text style={styles.text}>Hello Swiper</Text>
        </View>
        <View style={styles.slide2}>
          <Text style={styles.text}>Beautiful</Text>
        </View>
        <View style={styles.slide3}>
          <Text style={styles.text}>And simple</Text>
        </View>
      </Swiper>
    );
  }
}

AppRegistry.registerComponent('myproject', () => Swiper);
```

Properties

Basic

Prop	Default	Type	Description
horizontal	true	`bool`	If `true`, the scroll view's children are arranged horizontally in a row instead of vertically in a column.
loop	true	`bool`	Set to `false` to disable continuous loop mode.
index	0	`number`	Index number of initial slide.
showsButtons	false	`bool`	Set to `true` make control buttons visible.
autoplay	false	`bool`	Set to `true` enable auto play mode.

Prop	Default	Type	Description
onIndexChanged	(index) => null	`func`	Called with the new index when the user swiped

Custom basic style & content

Prop	Default	Type	Description
width	-	`number`	If no specify default enable fullscreen mode by `flex: 1` .
height	-	`number`	If no specify default fullscreen mode by `flex: 1` .
style	{...}	`style`	See default style in source.
containerStyle	{...}	`style`	See default container style in source.
loadMinimal	false	`bool`	Only load current index slide , `loadMinimalSize` slides before and after.
loadMinimalSize	1	`number`	see `loadMinimal`
loadMinimalLoader	`<ActivityIndicator />`	`element`	Custom loader to display when slides aren't loaded

Pagination

Prop	Default	Type	Description
showsPagination	true	`bool`	Set to `true` make pagination visible.
paginationStyle	{...}	`style`	Custom styles will merge with the default styles.
renderPagination	-	`function`	Complete control how to render pagination with three params (`index` , `total` , `context`) ref to `this.state.index` / `this.state.total` / `this` , For example: show numbers instead of dots.
dot	`<View style={{backgroundColor:'rgba(0,0,0,.2)', width: 8, height: 8,borderRadius: 4, marginLeft: 3, marginRight: 3, marginTop: 3, marginBottom: 3,}} />`	`element`	Allow custom the dot element.
activeDot	`<View style={{backgroundColor: '#007aff', width: 8, height: 8, borderRadius: 4, marginLeft: 3, marginRight: 3, marginTop: 3, marginBottom: 3,}} />`	`element`	Allow custom the active-dot element.
dotStyle	-	`object`	Allow custom the active-dot element.
dotColor	-	`string`	Allow custom the active-dot element.
activeDotColor	-	`string`	Allow custom the active-dot element.
activeDotStyle	-	`object`	Allow custom the active-dot element.

Autoplay

Prop	Default	Type	Description
autoplay	true	`bool`	Set to `true` enable auto play mode.
autoplayTimeout	2.5	`number`	Delay between auto play transitions (in second).
autoplayDirection	true	`bool`	Cycle direction control.

Control buttons

Prop	Default	Type	Description
showsButtons	true	`bool`	Set to `true` make control buttons visible.
buttonWrapperStyle	`{backgroundColor: 'transparent', flexDirection: 'row', position: 'absolute', top: 0, left: 0, flex: 1, paddingHorizontal: 10, paddingVertical: 10, justifyContent: 'space-between', alignItems: 'center'}`	`style`	Custom styles.
nextButton	`<Text style={styles.buttonText}>›</Text>`	`element`	Allow custom the next button.
prevButton	`<Text style={styles.buttonText}>‹</Text>`	`element`	Allow custom the prev button.

Props of Children

Prop	Default	Type	Description
style	{...}	`style`	Custom styles will merge with the default styles.
title	{...}	`element`	If this parameter is not specified, will not render the title.

Basic props of `<ScrollView />`

Prop	Default	Type	Description
horizontal	true	`bool`	If `true` , the scroll view's children are arranged horizontally in a row instead of vertically in a column.
pagingEnabled	true	`bool`	If true, the scroll view stops on multiples of the scroll view's size when scrolling. This can be used for horizontal pagination.
showsHorizontalScrollIndicator	false	`bool`	Set to `true` if you want to show horizontal scroll bar.
showsVerticalScrollIndicator	false	`bool`	Set to `true` if you want to show vertical scroll bar.
bounces	false	`bool`	If `true` , the scroll view bounces when it reaches the end of the content if the content is larger then the scroll view along the axis of the scroll direction. If `false` , it disables all bouncing even if the alwaysBounce* props are true.

Prop	Default	Type	Description
scrollsToTop	false	`bool`	If true, the scroll view scrolls to top when the status bar is tapped.
removeClippedSubviews	true	`bool`	If true, offscreen child views (whose overflow value is hidden) are removed from their native backing superview when offscreen. This canimprove scrolling performance on long lists.
automaticallyAdjustContentInsets	false	`bool`	Set to `true` if you need adjust content insets automation.
scrollEnabled	true	`bool`	Enables/Disables swiping

> @see: http://facebook.github.io/react-native/docs/scrollview.html

Supported ScrollResponder

Prop	Params	Type	Description
onScrollBeginDrag	`e / state / context`	`function`	When animation begins after letting up
onMomentumScrollEnd	`e / state / context`	`function`	Makes no sense why this occurs first during bounce
onTouchStartCapture	`e / state / context`	`function`	Immediately after `onMomentumScrollEnd`
onTouchStart	`e / state / context`	`function`	Same, but bubble phase
onTouchEnd	`e / state / context`	`function`	You could hold the touch start for a long time
onResponderRelease	`e / state / context`	`function`	When lifting up - you could pause forever before * lifting

> Note: each ScrollResponder be injected with two params: `state` and `context`, you can get `state` and `context` (ref to swiper's `this`) from params, for example:

```
var swiper = React.createClass({
  _onMomentumScrollEnd: function (e, state, context) {
    console.log(state, context.state)
  },
  render: function() {
    return (
      <Swiper style={styles.wrapper}
      onMomentumScrollEnd ={this._onMomentumScrollEnd}
     ...
      </Swiper>
    )
  }
})
```

> More ScrollResponder info, see: https://github.com/facebook/react-native/blob/master/Libraries/Components/ScrollResponder.js

Methods

scrollBy(index, animated)

Scroll by relative index.

Parameters:

Name	Type	default	Description
index	`number`	`undefined`	offset index
animated	`bool`	`true`	offset index

Examples

```
$ cd examples
$ npm i
$ react-native run-ios
```

> Quick start with examples.

Development

```
$ cd examples
$ npm i
$ npm run dev
$ react-native run-ios
```

Then launch simulator to preview. Note that you just need to edit the source file `src/index.js` , the change will auto sync to examples.

And now that this project follows the standard code style, you'd better prepare it for IDE.

Contribution

- @leecade The main author.
- @rajkissu The secondary contributor.

Questions

Feel free to contact me or create an issue

> Inspired by nolimits4web/Swiper & Design material from Dribbble & made with ♥.