

Settings

Settings serves as a wrapper for `NSUserDefaults`, a persistent key-value store available only on iOS.

Example

Settings Example

[^ Expo](#)

```
import React, {useState} from 'react';
import {Button, Settings, StyleSheet, Text, View} from
'react-native';

const App = () => {
  const [data, setData] = useState(() =>
Settings.get('data'));

  return (
    <View style={styles.container}>
      <Text>Stored value:</Text>
      <Text style={styles.value}>{data}</Text>
      <Button
        onPress={() => {
          Settings.set({data: 'React'});
          setData(Settings.get('data'));
        }}
        title="Store 'React'"
      />
      <Button
        onPress={() => {
          Settings.set({data: 'Native'});
          setData(Settings.get('data'));
        }}
        title="Store 'Native'"
      />
    </View>
  );
};
```

Download [Expo Go](#) and scan
the QR code to get started.



Connected devices 0

[Log in](#) or set a [Device ID](#) to open this Snack
from Expo Go on your device or simulator.

Preview



My Device

iOS

Reference

Methods

clearWatch()

```
static clearWatch(watchId: number);
```

watchId is the number returned by watchKeys() when the subscription was originally configured.

get()

```
static get(key: string): any;
```

Get the current value for a given key in NSUserDefaults.

set()

```
static set(settings: Record<string, any>);
```

Set one or more values in NSUserDefaults.


watchKeys()

```
static watchKeys(keys: string | array<string>, callback: () => void): number;
```

Subscribe to be notified when the value for any of the keys specified by the keys parameter has been changed in NSUserDefaults. Returns a watchId number that may be used with clearWatch() to unsubscribe.

Note: watchKeys() by design ignores internal set() calls and fires callback only on changes preformed outside of React Native code.

Is this page useful?  

 Edit this page

*Last updated on **Jun 21, 2023***