Version: 2.6.0 – 2.12.0

Handler State

As described in "About Gesture Handlers", gesture handlers can be treated as "state machines". At any given time, each handler instance has an assigned state that can change when new touch events occur or can be forced to change by the touch system in certain circumstances.

A gesture handler can be in one of the six possible states:

- UNDETERMINED
- FAILED
- BEGAN
- CANCELLED
- ACTIVE
- END

Each state has its own description below.

Accessing state

We can monitor a handler's state changes by using the `onHandlerStateChange` callback and the destructured `nativeEvent` argument passed to it. This can be done by comparing the `nativeEvent`'s `state` attribute to one of the constants exported under the `State` object (see example below).

```
import { State, LongPressGestureHandler } from 'react-native-gesture-handler';

class Demo extends Component {
  _handleStateChange = ({ nativeEvent }) => {
    if (nativeEvent.state === State.ACTIVE) {
      Alert.alert('Longpress');
    }
  };
  render() {
    return (
      <LongPressGestureHandler onHandlerStateChange={this._handleStateChange}>
        <Text style={styles.buttonText}>Longpress me</Text>
      </LongPressGestureHandler>
    );
  }
}
```

State flows

The most typical flow of state is when a gesture handler picks up on an initial touch event then recognizes it then acknowledges its ending then resets itself back to the initial state.

The flow looks as follows (longer arrows represent that there are possibly more touch events received before the state changes):

`UNDETERMINED` -> `BEGAN` ------> `ACTIVE` ------> `END` -> `UNDETERMINED`

Another possible flow is when a handler receives touches that cause a recognition failure:

`UNDETERMINED` -> `BEGAN` ------> `FAILED` -> `UNDETERMINED`

At last, when a handler does properly recognize the gesture but then is interrupted by the touch system. In that case, the gesture recognition is canceled and the flow looks as follows:

`UNDETERMINED` -> `BEGAN` ------> `ACTIVE` ------> `CANCELLED` -> `UNDETERMINED`

States

The section below describes all possible handler states:

UNDETERMINED

This is the initial state of each handler and it goes into this state after it's done recognizing a gesture.

FAILED

A handler received some touches but for some reason didn't recognize them. For example, if a finger travels more distance than a defined `maxDist` property allows, then the handler won't become active but will fail instead. Afterwards, it's state will be reset to `UNDETERMINED`.

BEGAN

Handler has started receiving touch stream but hasn't yet received enough data to either fail or activate.

CANCELLED

The gesture recognizer has received a signal (possibly new touches or a command from the touch system controller) resulting in the cancellation of a continuous gesture. The gesture's state will become `CANCELLED` until it is finally reset to the initial state, `UNDETERMINED`.

ACTIVE

Handler has recognized a gesture. It will become and stay in the `ACTIVE` state until the gesture finishes (e.g. when user lifts the finger) or gets cancelled by the touch system. Under normal circumstances the state will then turn into `END`. In the case that a gesture is cancelled by the touch system, its state would then become `CANCELLED`. Learn about discrete and continuous handlers here to understand how long a handler can be kept in the `ACTIVE` state.

END

The gesture recognizer has received touches signalling the end of a gesture. Its state will become `END` until it is reset to `UNDETERMINED`.