Version: 3.x

Customizing animations

Previous section not only taught you how to use shared values in practice but also you used `withSpring` and `withTiming` functions to create animations. We think now you're more than ready to dive deeper into customizing animations!

Reanimated comes with three built-in animation functions: `withTiming`, `withSpring`, and `withDecay`. For now, let's focus on the first two, and we'll come back to `withDecay` in the Handling gestures section.

It's very easy to customize the behavior of animation functions in Reanimated. You can do this by passing a `config` object to the second parameter of either `withTiming` or `withSpring` function.

Configuring `withTiming`

The `config` parameter of `withTiming` comes with two properties: `duration` and `easing`.

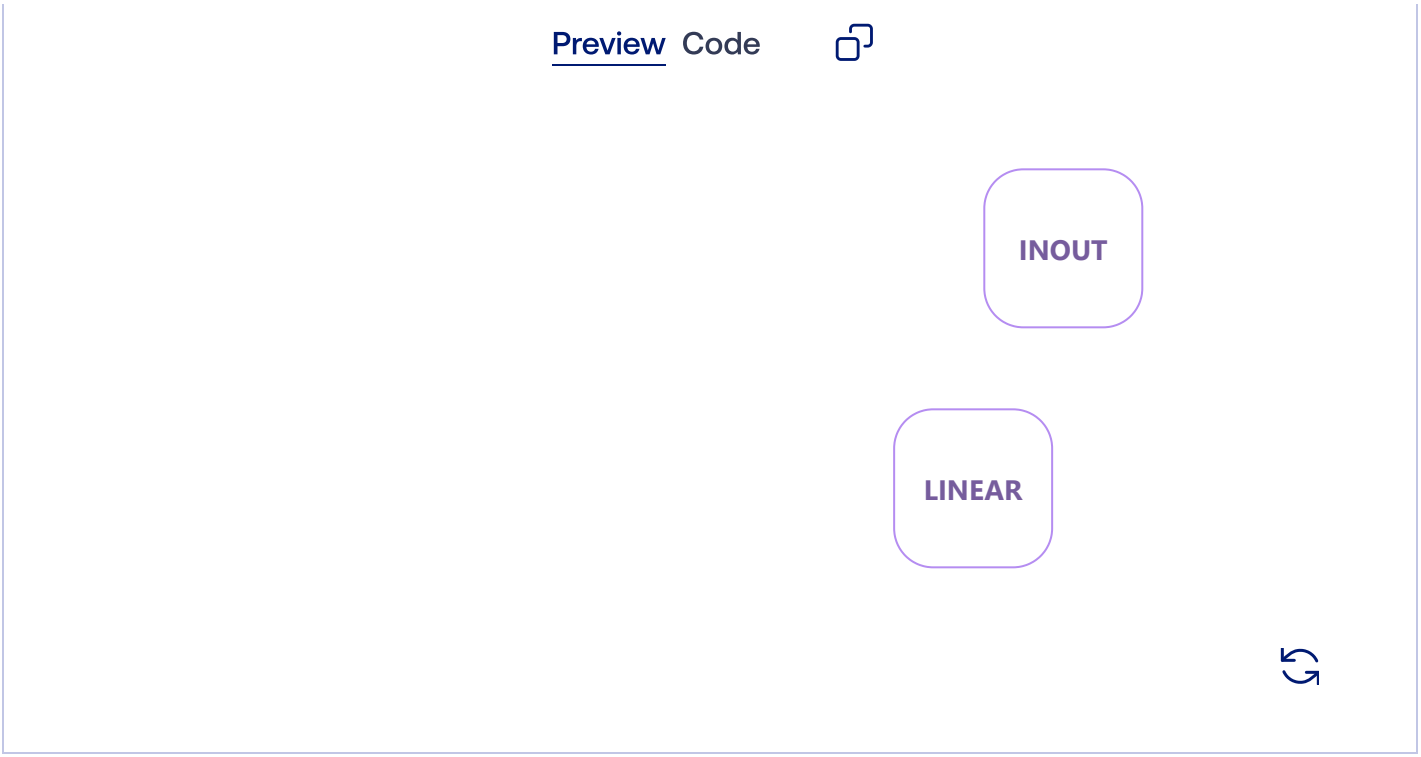
```
import { withTiming, Easing } from 'react-native-reanimated';

withTiming(sv.value, {
  duration: 300,
  easing: Easing.inOut(Easing.quad),
});
```

Simple enough, the `duration` parameter defines how long in milliseconds the animation should take to reach the assigned `toValue`. The duration by default is set to `300` milliseconds.

The `easing` parameter lets you fine-tune the animation over the specified time. For example, you can make the animation start slowly then pickup some speed and slow it down again towards the end. This value defaults to `Easing.inOut(Easing.quad)`.

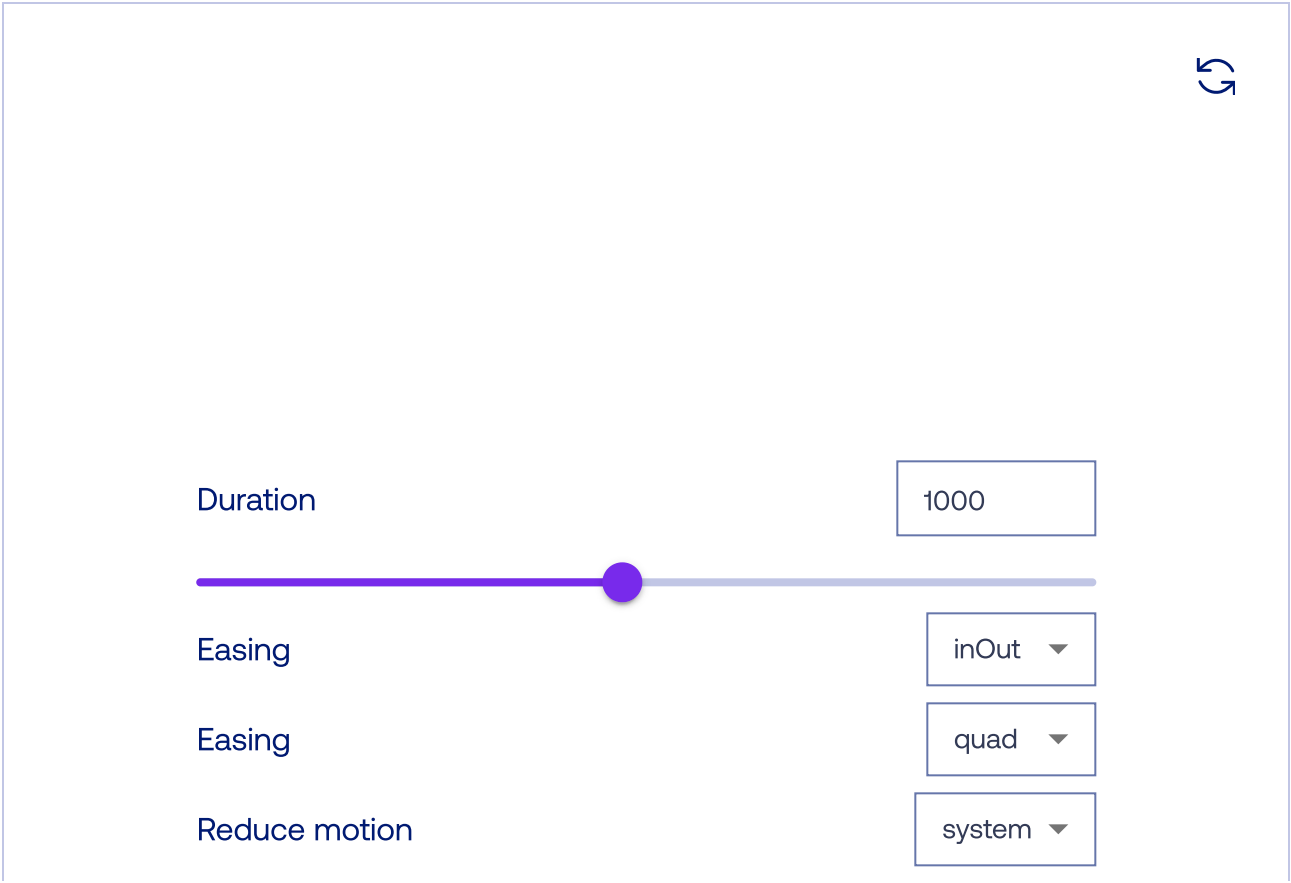
It all will start to make sense when you compare side by side a `linear` easing with the default easing.

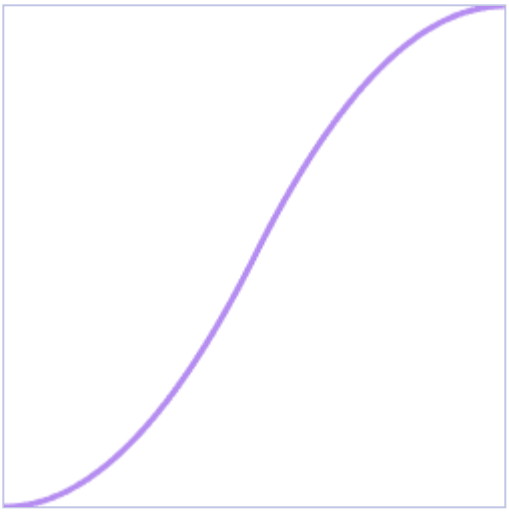
Preview Code



Reanimated comes with a handful of predefined easing functions. You can play around with them in the interactive playground below or check the full `withTiming` **API reference**.

`withTiming` interactive playground





```
withTiming(sv.value, {
  duration: 1000,
  easing: Easing.inOut(Easing.quad),
  reduceMotion: ReduceMotion.System,
})
```

Configuring `withSpring`

`withSpring` is a physics-based animation function which works way differently from `withTiming`. It makes it look like the object you're animating is connected to a real spring. The physics-based approach makes the animations look *believable*.

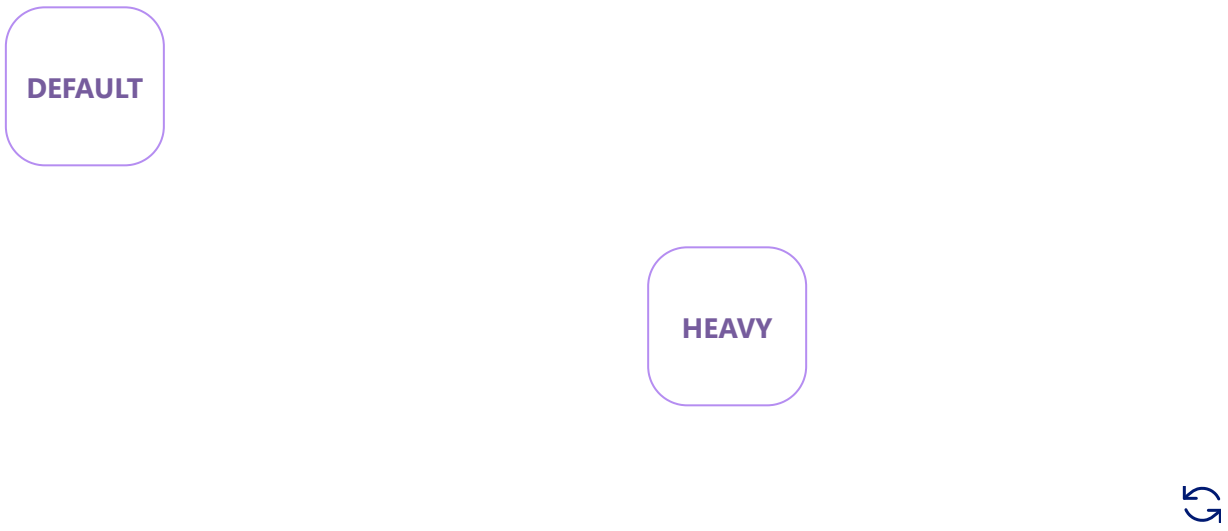
Most of the time when tinkering with springs you'll be adjusting one of these three properties: `mass`, `stiffness` (also known as *tension*), and `damping` (also known as *friction*).

```
import { withSpring } from 'react-native-reanimated';

withSpring(sv.value, {
  mass: 1,
  stiffness: 100,
```

```
  damping: 10,
});
```

The mass of a spring influences how hard is it to make an object move and to bring it to a stop. Mass adds a feeling of ***inertia*** to the object you're trying to move. You can see in the playground that the spring with greater mass moves more "sluggish" compared to the default configuration.

Preview Code

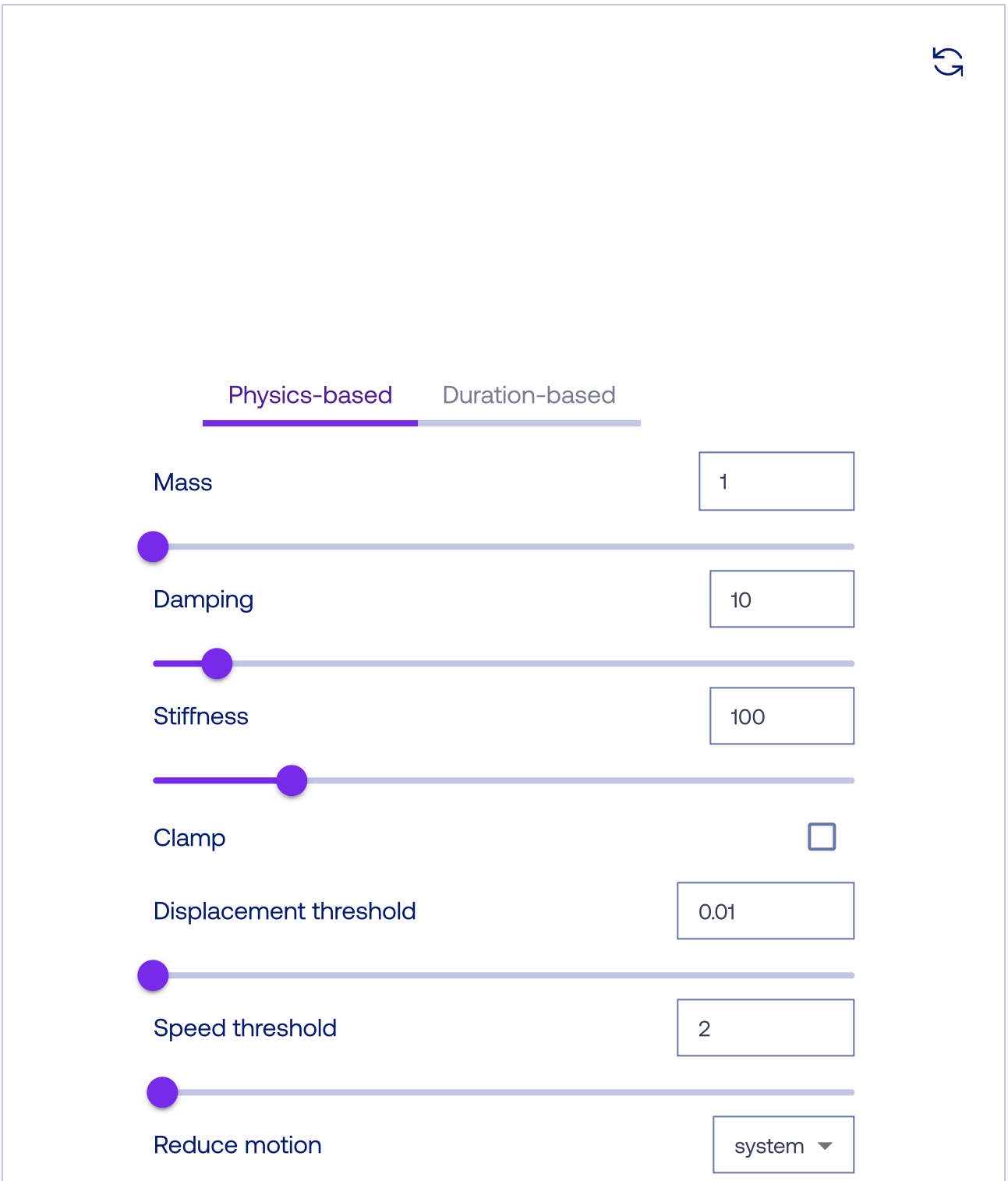


Stiffness affects how bouncy the spring is. As an example, think about the difference between a steel spring (with very high stiffness) and a spring made out of soft plastic (with low stiffness).

Damping describes how quickly the spring animation finishes. Higher damping means the spring will come to rest faster. In the real world, you could think about the same spring bouncing in the air and underwater. For example, a spring in a vacuum would have zero friction and thus zero damping.

Reanimated comes with a handful of other properties used to customize your spring animation. You can play around with them in our interactive playground or check the full `withSpring` **API reference**.

`withSpring` interactive playground




```
withSpring(sv.value, {
  mass: 1,
  damping: 10,
  stiffness: 100,
  overshootClamping: false,
  restDisplacementThreshold: 0.01,
```

```
    restSpeedThreshold: 2,
    reduceMotion: ReduceMotion.System,
  })
```

Summary

In this section, we learned how to customize the `withTiming` and `withSpring` animation functions. To sum up:

- Both `withTiming` and `withSpring` functions take the `config` object as a second parameter.
- You can adjust `withTiming` behavior with `duration` and `easing` properties. For your convenience Reanimated comes with a built-in `Easing` module.
- Some of the properties which adjust the behavior of `withSpring` are `mass`, `stiffness` and `damping`.

What's next?

In the upcoming section, you'll discover how to use animation modifiers such as `withSequence` and `withRepeat`. These modifiers we'll allow us to create more complex and engaging animations.

Edit this page