

implementing the chart with

bit.ly/cannvas_spring

```
export default function App() {

  const { x } = useSpring({
    from: { x: FROM_X },
    to: async (next) => {
      while (1) {
        await next({ x: TO_X });
        await next({ x: FROM_X });
      }
    }
  });

  return (
    <div className="App">
      <animated.div>{x}</animated.div>

    </div>
  );
}
```




100.02


```
<AnimatedCanvas x={x} />
```



100.02

Re-implementing the chart with Canvas

```
export default function App() {  
  
  const { x } = useSpring({  
    from: { x: FROM_X },  
    to: async (next) => {  
      while (1) {  
        await next({ x: TO_X });  
        await next({ x: FROM_X });  
      }  
    }  
  });  
  
  return (  
    <div className="App">  
      <animated.div>{x}</animated.div>  
      <AnimatedCanvas x={x} />  
    </div>  
  );  
}
```



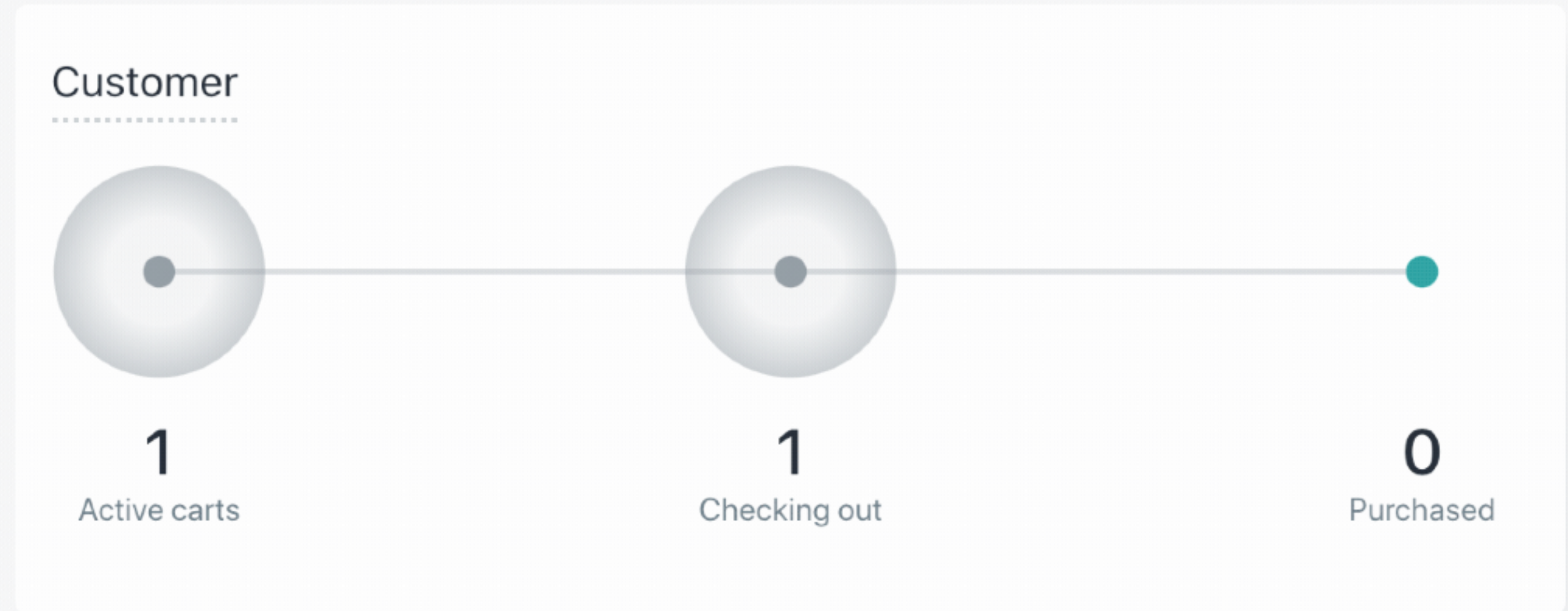
100.02

Re-implementing the chart with Canvas

```
const radiusSprings = useSprings(  
  // ...  
);
```

```
const ordersRunningDots = useSprings(  
  // ...  
);
```

```
<CustomerBehaviorCanvas  
  radiusSprings={bucketsRadius}  
  ordersRunningDots={ordersRunningDots}  
>
```



← Open custom settings