

[API reference](#)[Actions](#)[StackActions](#)

Version: 6.x

StackActions reference

`StackActions` is an object containing methods for generating actions specific to stack-based navigators. Its methods expand upon the actions available in `CommonActions`.

The following actions are supported:

replace

The `replace` action allows to replace a route in the navigation state. It takes the following arguments:

- `name` - *string* - A destination name of the route that has been registered somewhere.
- `params` - *object* - Params to pass to the destination route.

```
import { StackActions } from '@react-navigation/native';

navigation.dispatch(
  StackActions.replace('Profile', {
    user: 'jane',
  })
);
```

Try this example on Snack [↗](#)

If you want to replace a particular route, you can add a `source` property referring to the route key and `target` property referring to the navigation state key:

```
import { StackActions } from '@react-navigation/native';

navigation.dispatch({
  ...StackActions.replace('Profile', {
    user: 'jane',
  }),
  source: route.key,
```

```
target: navigation.getState().key,  
});
```

Try this example on Snack [↗](#)

If the `source` property is explicitly set to `undefined`, it'll replace the focused route.

push

The `push` action adds a route on top of the stack and navigates forward to it. This differs from `navigate` in that `navigate` will pop back to earlier in the stack if a route of the given name is already present there. `push` will always add on top, so a route can be present multiple times.

- `name` - *string* - Name of the route to push onto the stack.
- `params` - *object* - Screen params to pass to the destination route.

```
import { StackActions } from '@react-navigation/native';  
  
const pushAction = StackActions.push('Profile', { user: 'Wojtek' });  
  
navigation.dispatch(pushAction);
```

Try this example on Snack [↗](#)

pop

The `pop` action takes you back to a previous screen in the stack. It takes one optional argument (`count`), which allows you to specify how many screens to pop back by.

```
import { StackActions } from '@react-navigation/native';  
  
const popAction = StackActions.pop(1);  
  
navigation.dispatch(popAction);
```


Try this example on Snack [↗](#)

popToTop

The `popToTop` action takes you back to the first screen in the stack, dismissing all the others. It's functionally identical to `StackActions.pop({n: currentIndex})`.

```
import { StackActions } from '@react-navigation/native';  
  
navigation.dispatch(StackActions.popToTop());
```

Try this example on Snack [↗](#)

 [Edit this page](#)