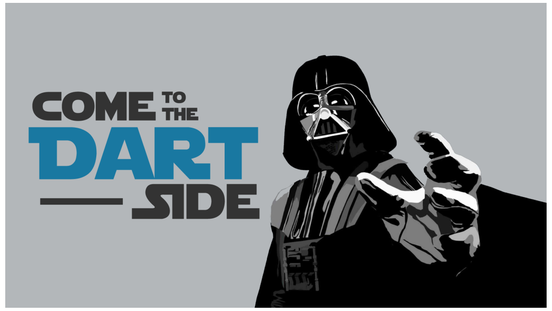
LET'S GO FLUTTER

4. NAVIGHIAMO IN NUOVE SCHERMATE

the Communities Bay

Federico Parezzan

Java Developer 👨‍💻



Sviluppatori Flutter

federico-parezzan

parez93

parezzan.com

thecmmbay.com

Sono un essere umano, nulla che sia umano mi è estraneo ~ Terenzio

Agenda

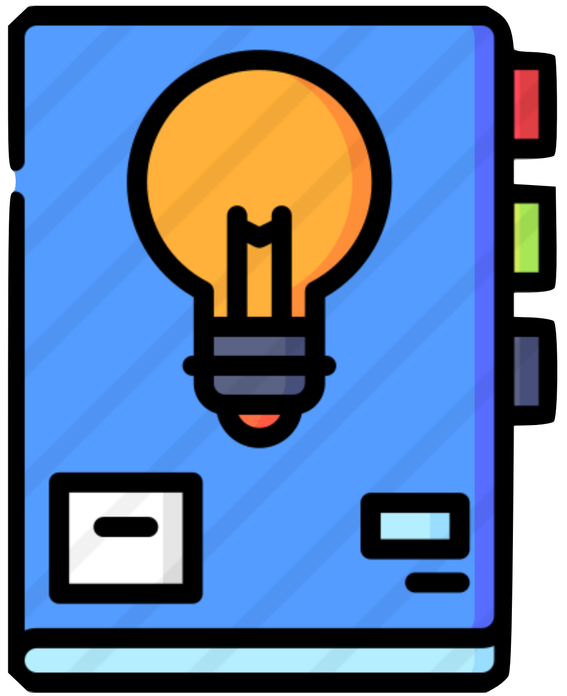
What is the navigation?

Introducing Navigator

Navigator 1.0

Navigation API

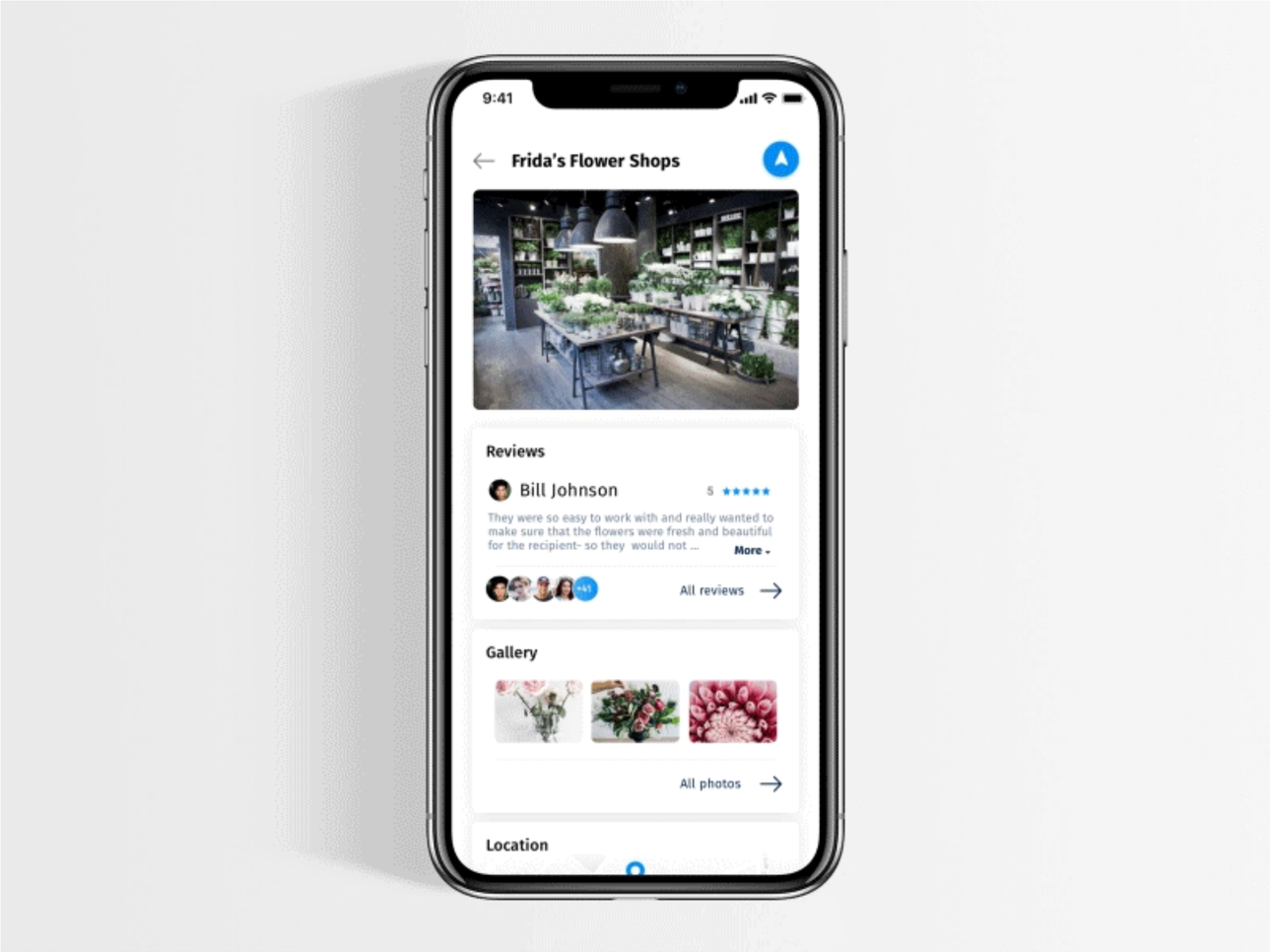
Over Navigator 1.0



What is the navigation?

What is the navigation?

How users switch between different screens



Introducing Navigator

Introducing Navigator

UINavigationController

Introducing Navigator

UINavigationController

Jetpack Navigation

Introducing Navigator

UINavigationController

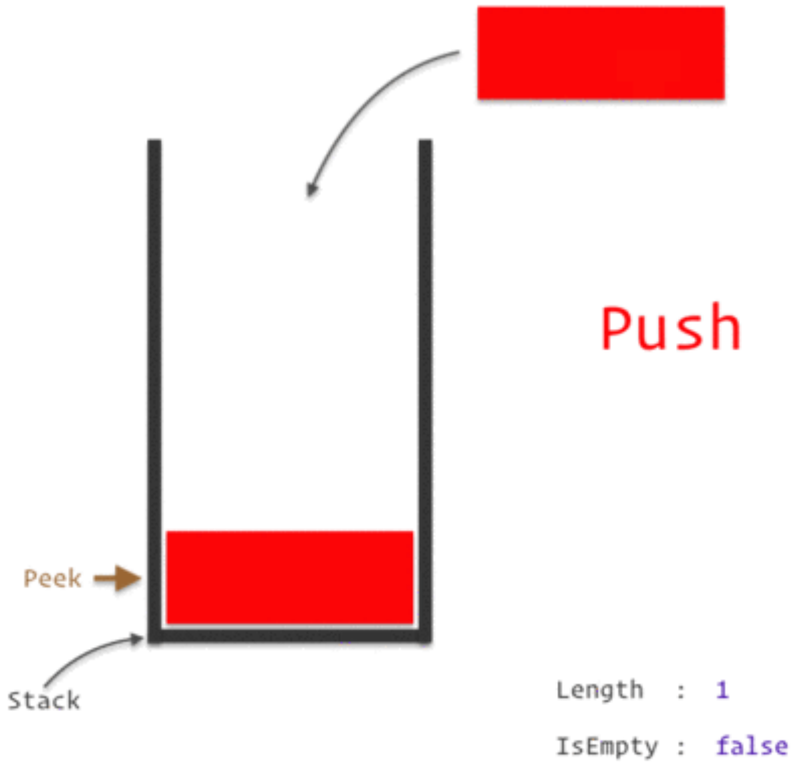
Jetpack Navigation

Navigator

Introducing Navigator

Stack

- Data structure
- LIFO
- Only the element at the top is visible to the user



Navigator 1.0

Navigator 1.0

Key points

Only way until 1.22 Flutter version

Navigator 1.0

Key points

Only way until 1.22 Flutter version

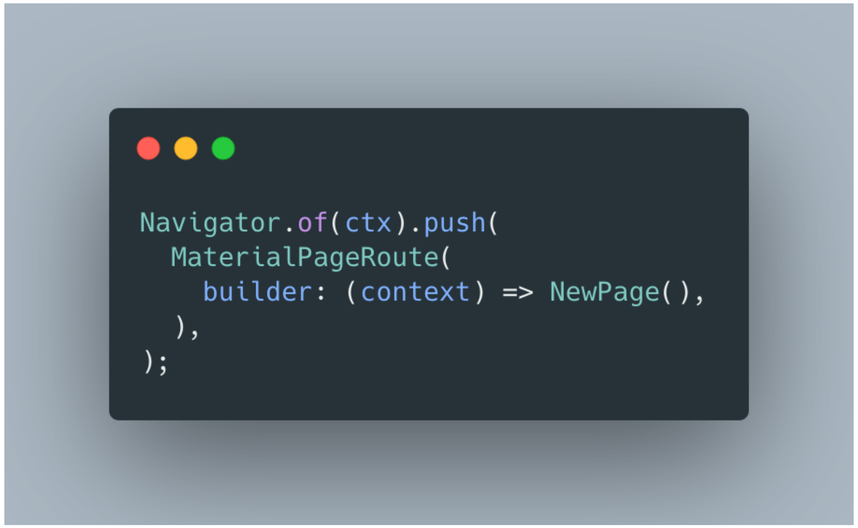
Provide a set of API to navigate between screens

Navigator 1.0

Key points

Only way until 1.22 Flutter version

Provide a set of API to navigate between screens



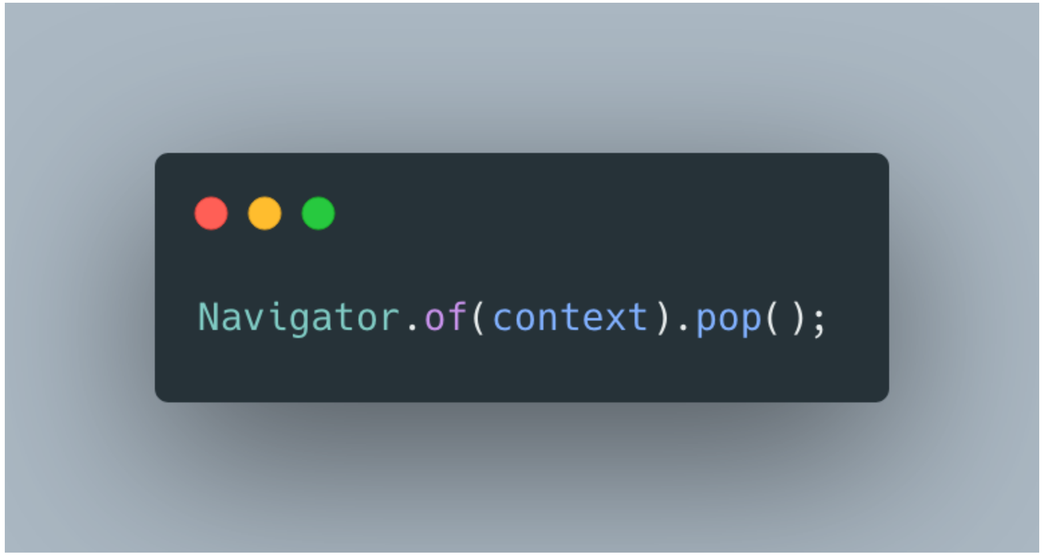
```
Navigator.of(ctx).push(
  MaterialPageRoute(
    builder: (context) => NewPage(),
  ),
);
```

Navigator 1.0

Key points

Only way until 1.22 Flutter version

Provide a set of API to navigate between screens



Navigator 1.0

Key points

Only way until 1.22 Flutter version

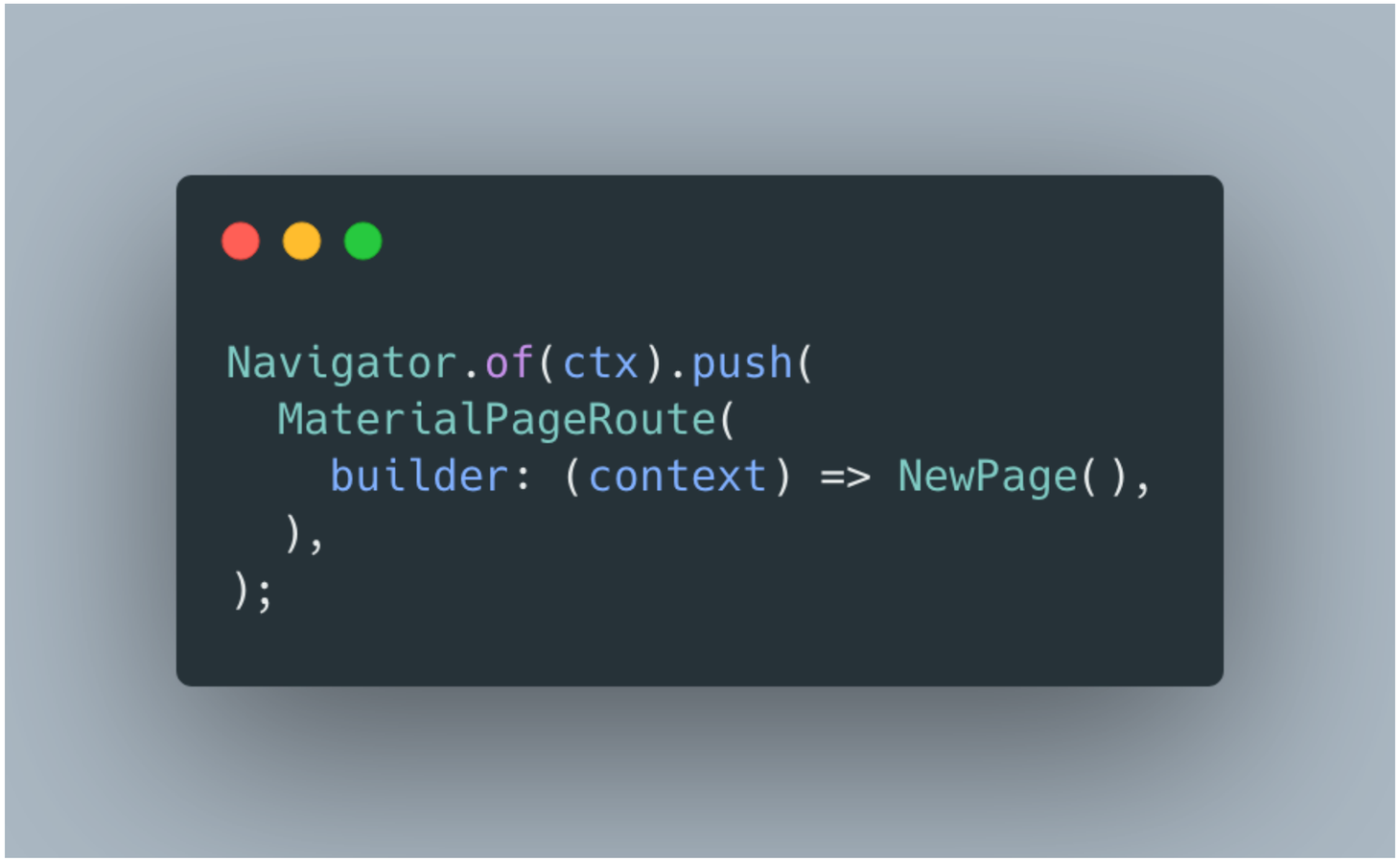
Provide a set of API to navigate between screens

It is provided by **WidgetApp** or its extensions

Navigation API

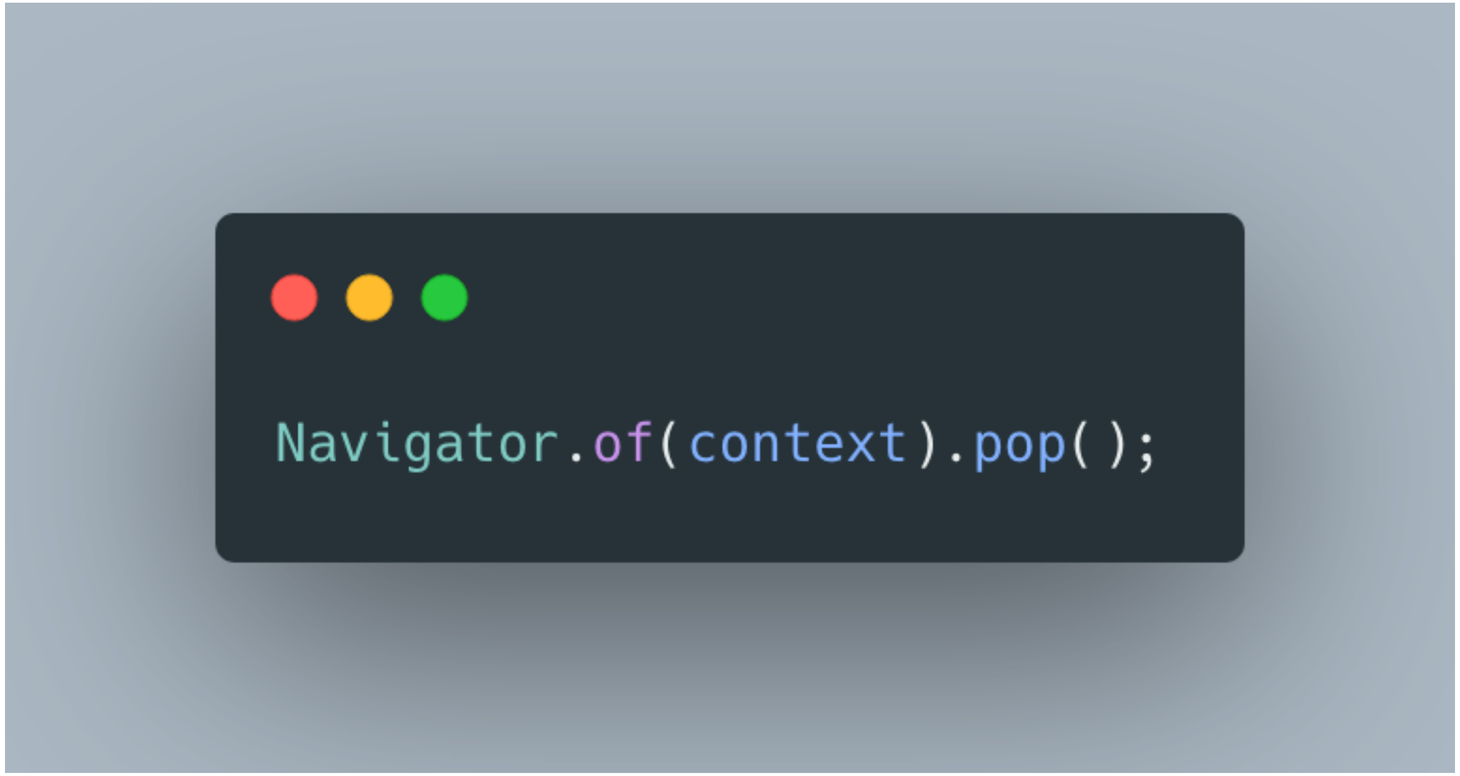
Navigation API

push()



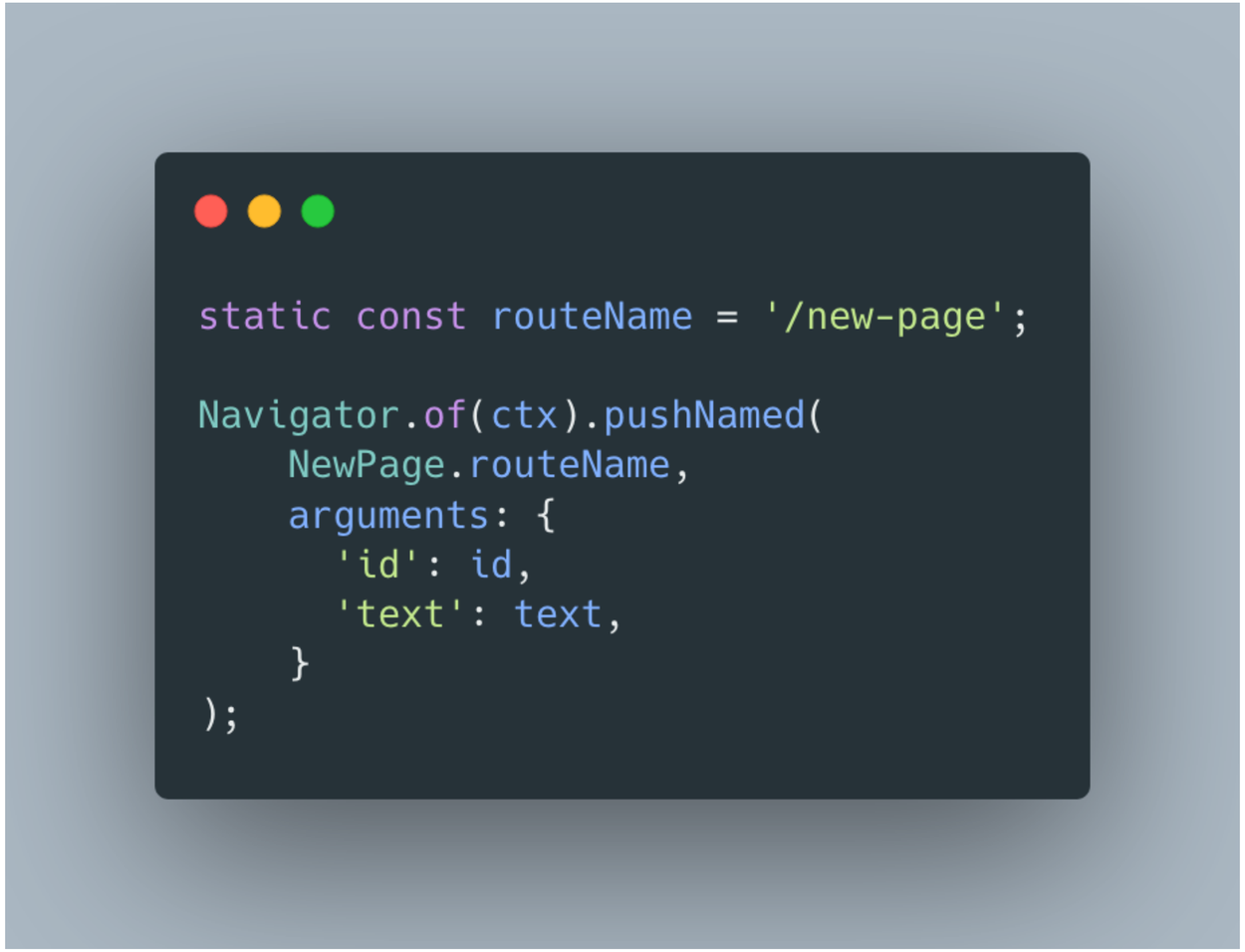
Navigation API

pop()



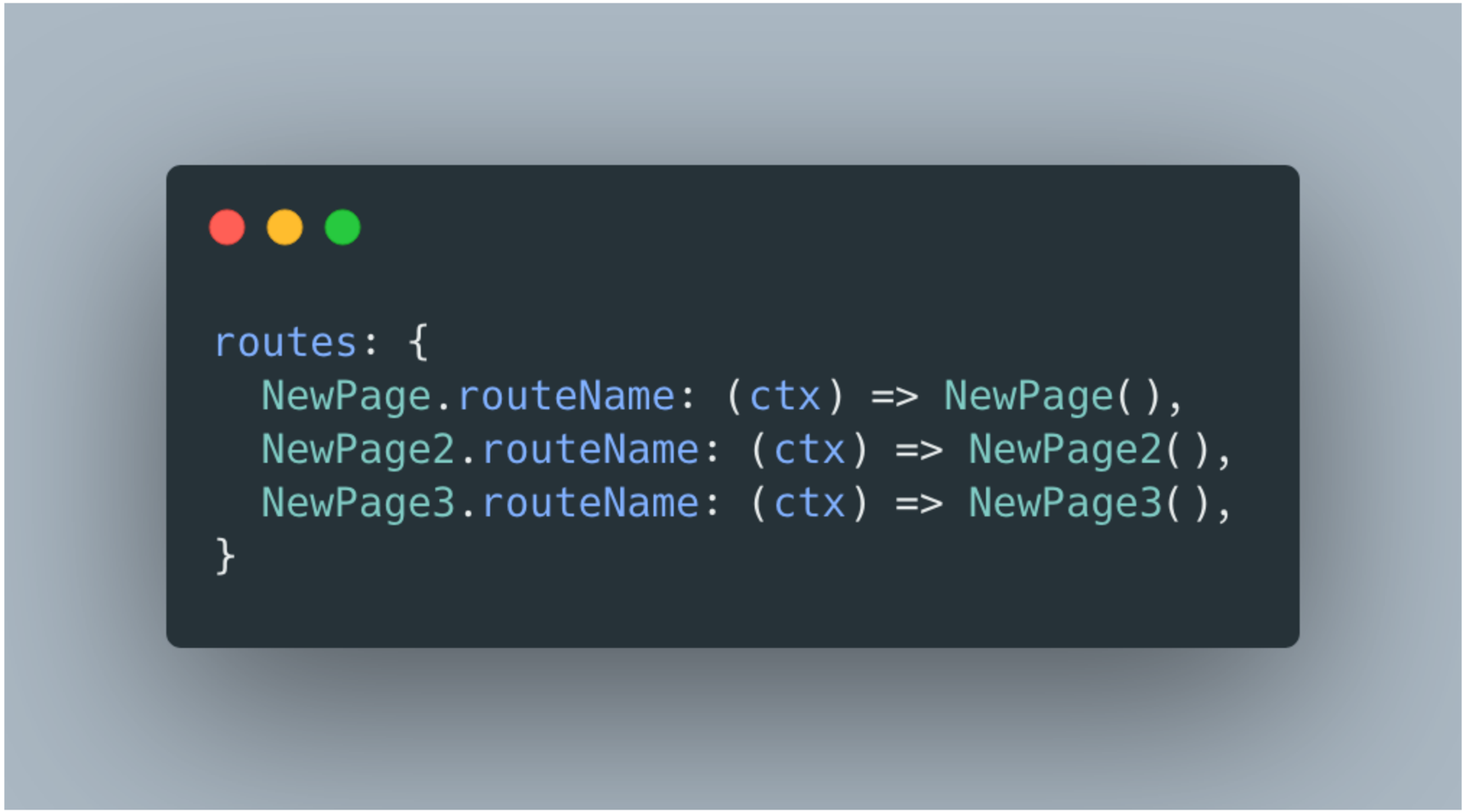
Navigation API

pushNamed()



Navigation API

pushNamed()

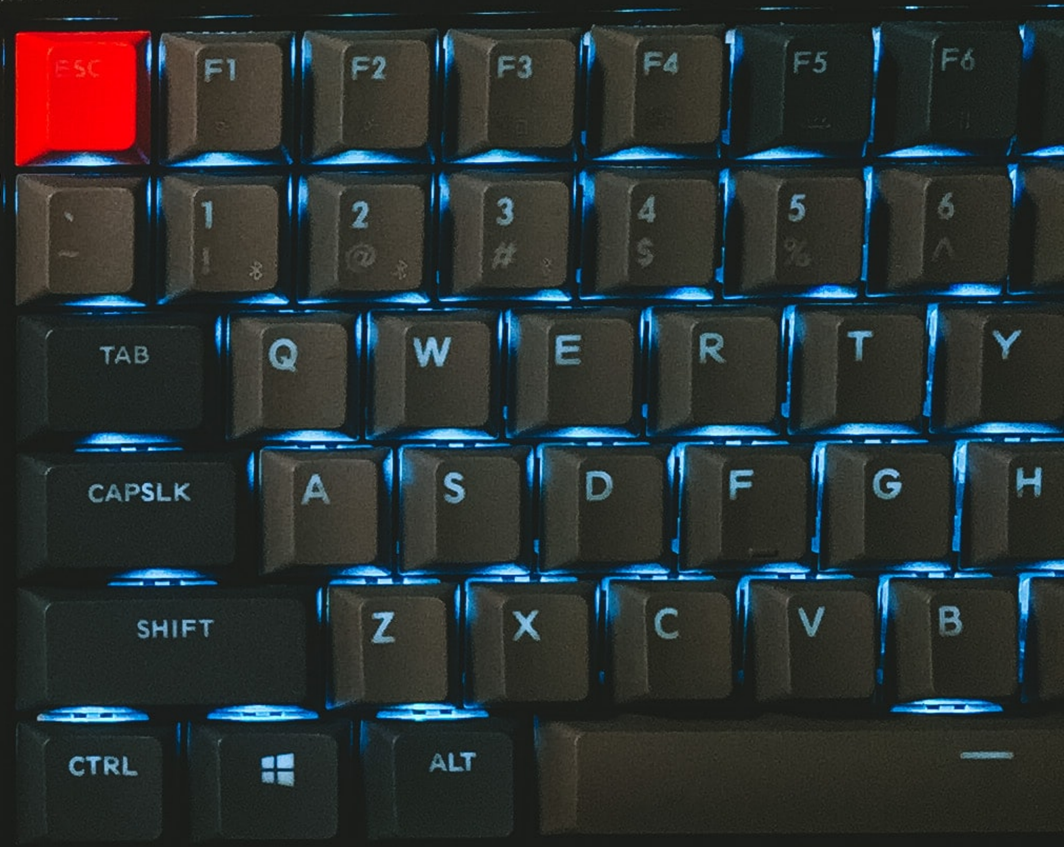


Navigation API

pushNamed()

```
final routeArgs = ModalRoute.of(ctx).settings.arguments as Map
```

DEMO


ESC
F1
F2
F3
F4
F5
F6
TAB
Q
W
E
R
T
Y
CAPSLK
A
S
D
F
G
H
SHIFT
Z
X
C
V
B
CTRL
ALT

Navigation API

Other push/pop methods

Method	Description
pushReplacement pushNamedReplacement	Replace the screen with a new one
pushAndRemoveUntil pushNamedAndRemoveUntil	It will push the page and remove all other routes as soon as the condition provided it met
popAndPushNamed	It will dispose of the current page and push another page according to the route provided.
popUntil	It will repeatedly pop pages until the condition is met.
canPop	It will check if the page can be popped.
restorablePopAndPushNamed	It will pop the current route of the navigator and push a named route in its place.
maybePop	It checks for the willPop() method and return result accordingly.

Disadvantages

Disadvantages

Hard to manage and scale (imperative API).

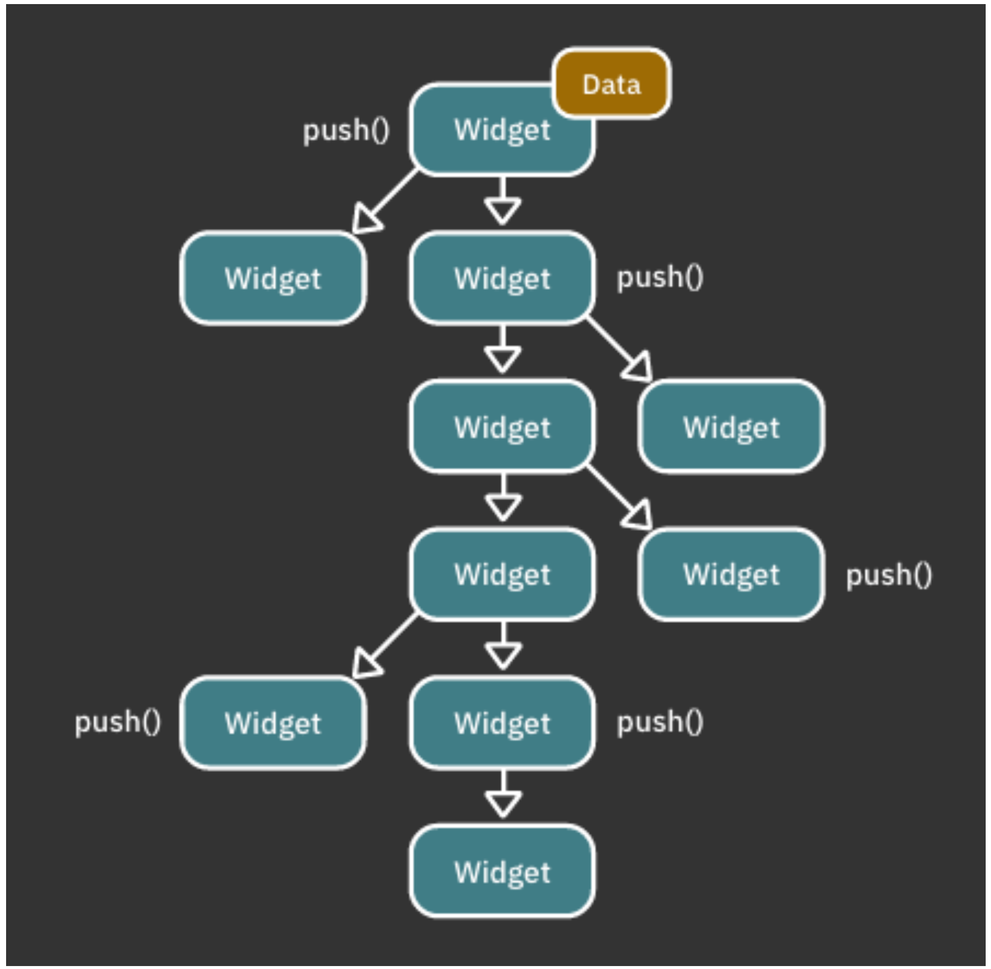
Disadvantages

Hard to manage

and scale (imperative API).

Need mental map of where

you push and pop a screen



Disadvantages

Hard to manage and scale (imperative API).

Need mental map of where you push and pop a screen

Doesn’t expose the route stack to developers

Disadvantages

Hard to manage and scale (imperative API).

Need mental map of where you push and pop a screen

Doesn’t expose the route stack to developers

Does not update the web URL path

Disadvantages

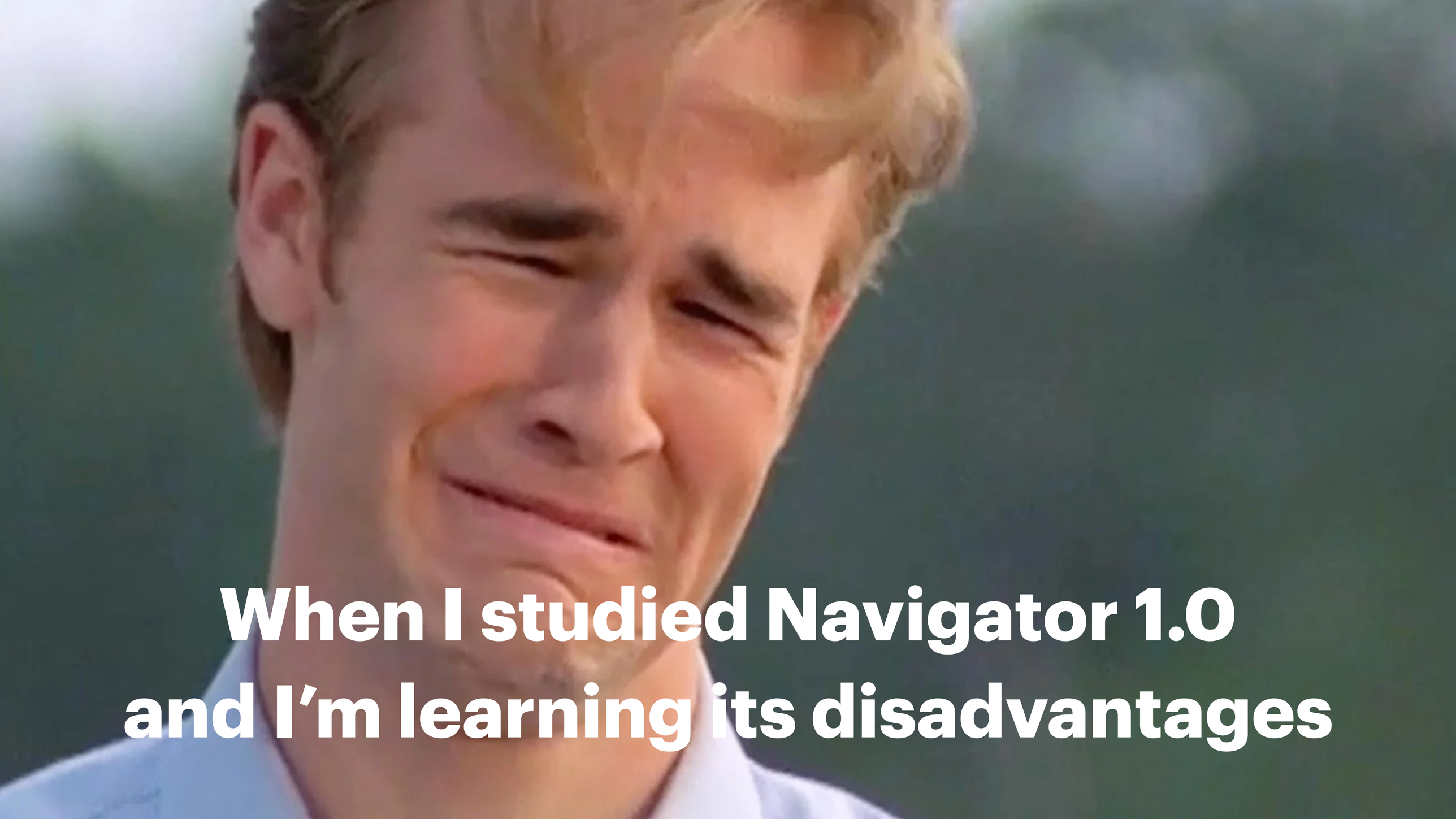
Hard to manage and scale (imperative API).

Need mental map of where you push and pop a screen

Doesn’t expose the route stack to developers

Does not update the web URL path

On Android devices, the Back button might not work


When I studied Navigator 1.0
and I'm learning its disadvantages


NAVIGATOR 2.0

Navigator 2.0

Navigator 2.0

Key points

Declarative API that allows you to take full control of your navigation stack

Navigator 2.0

Key points

Declarative API that allows you to take full control of your navigation stack

Exposing the navigator’s page stack

Navigator 2.0

Key points

Declarative API that allows you to take full control of your navigation stack

Exposing the navigator’s page stack

Backward-compatible with imperative API

Navigator 2.0

Key points

Declarative API that allows you to take full control of your navigation stack

Exposing the navigator’s page stack

Backward-compatible with imperative API

Handle operating system events

Navigator 2.0

Key points

Declarative API that allows you to take full control of your navigation stack

Exposing the navigator’s page stack

Backward-compatible with imperative API

Handle operating system events

Manage nested navigators

Navigator 2.0

Key points

Declarative API that allows you to take full control of your navigation stack

Exposing the navigator’s page stack

Backward-compatible with imperative API

Handle operating system events

Manage nested navigators

Manage navigation state

Navigator 2.0 vs 1.0

Navigator 2.0 vs 1.0

For medium to large apps

- You may have to manage a lot of your navigation state
- declarative API

Navigator 2.0

Navigator 2.0 vs 1.0

For small apps

- rapid prototyping and demos
- imperative API

Navigator 1.0

Recap

What we have learned

- What is navigation
- Intro Navigator 1.0
- Navigator 1.0 API
- Hints of Navigator 2.0

NAVIGATOR 2.0

NAVIGATOR 1.0

#Dashatar

https://dashatar-dev.web.app/


42
Q&A