



Separating concerns with RubigoNavigator

an implementation using Flutter's Navigator 2.0

<https://yapb.dev/>

<https://twitter.com/jsroest>

<https://github.com/jsroest>

<https://www.linkedin.com/in/sanderroest/>

Table of contents

1. A sample app, to demonstrate the use case of this navigator.
2. The problem and a solution.
3. Navigator 2.0 and RubigoNavigator.
4. RubigoNavigator overview (classes and events).
5. Let me see the code.... live demo.
6. Package 'mason', to generate page/controller pairs.

Quality Control app

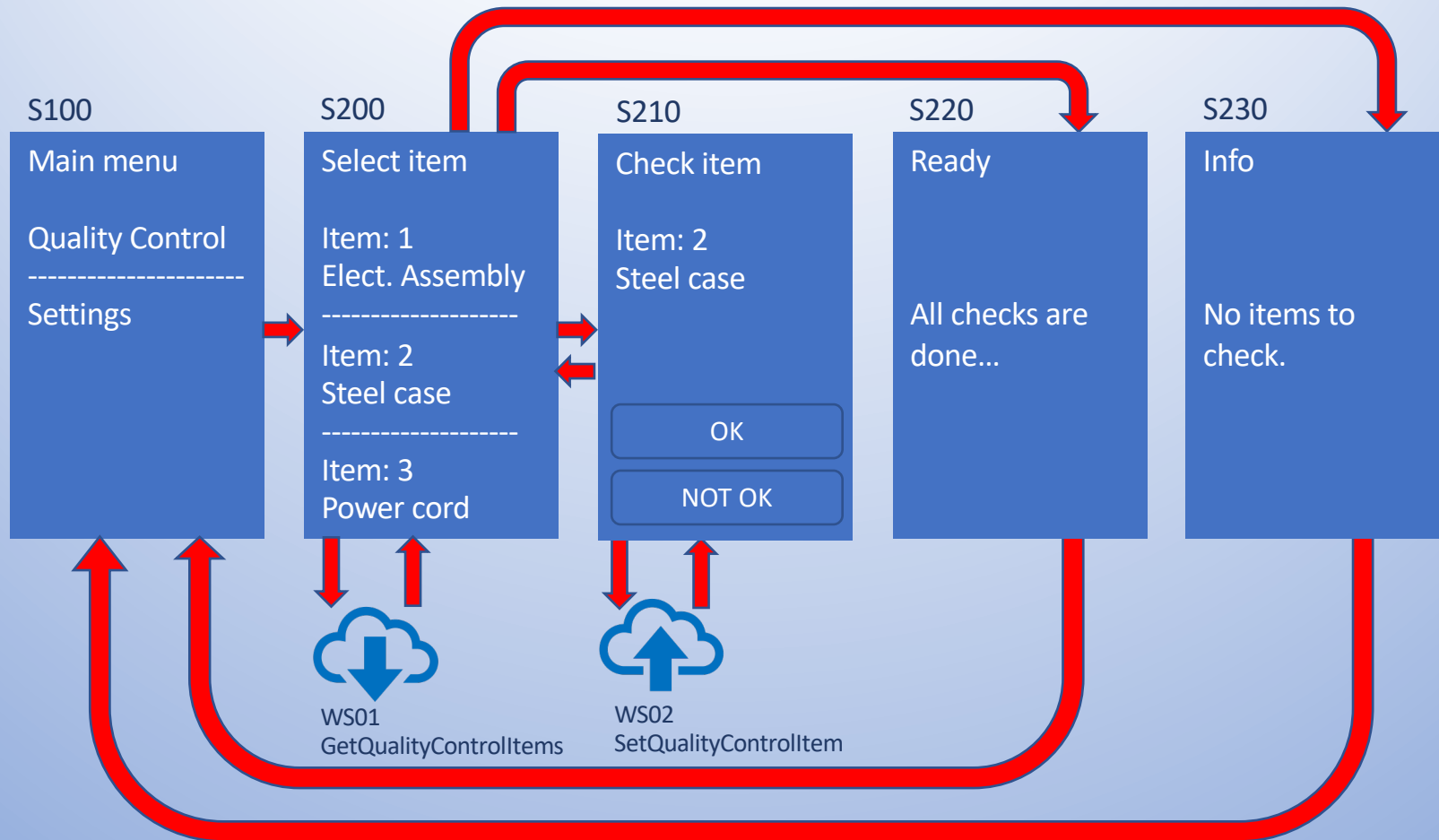
According the customer: "A really simple app"

Let's find out...

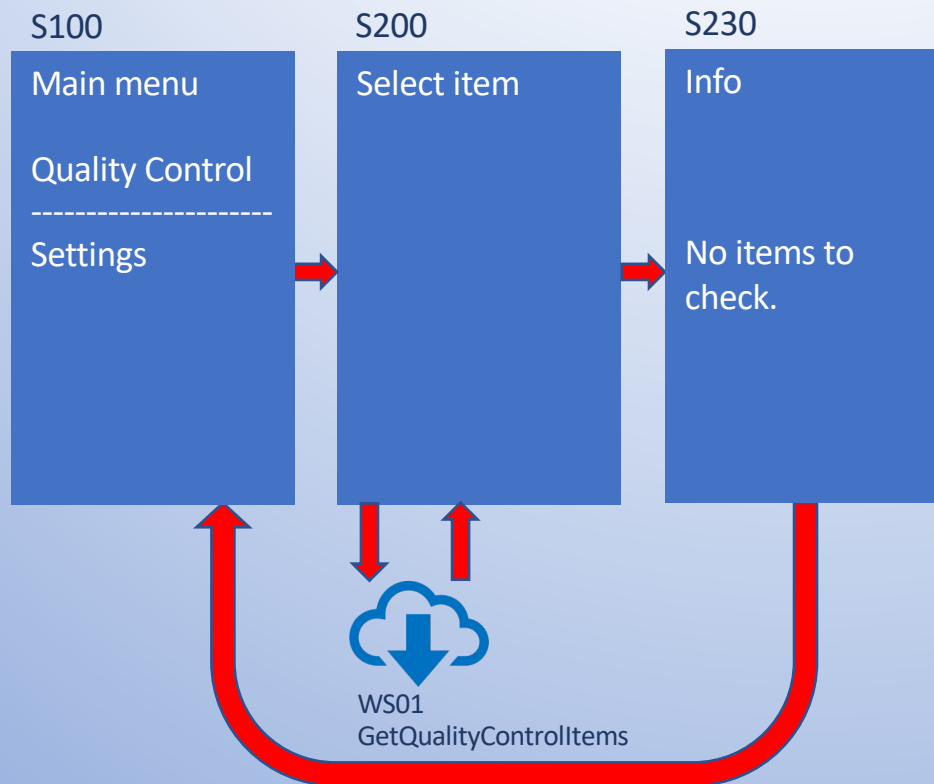
Requirements:

1. Get a list of 'items to check' from a webservice.
2. The user chooses an item from the list.
If just 1 item, continue automatically.
3. The user checks the item. Result is 'ok' or 'not ok'.
4. Send the result back via a webservice and remove the item from the list.
5. Repeat until all items are checked.

Design



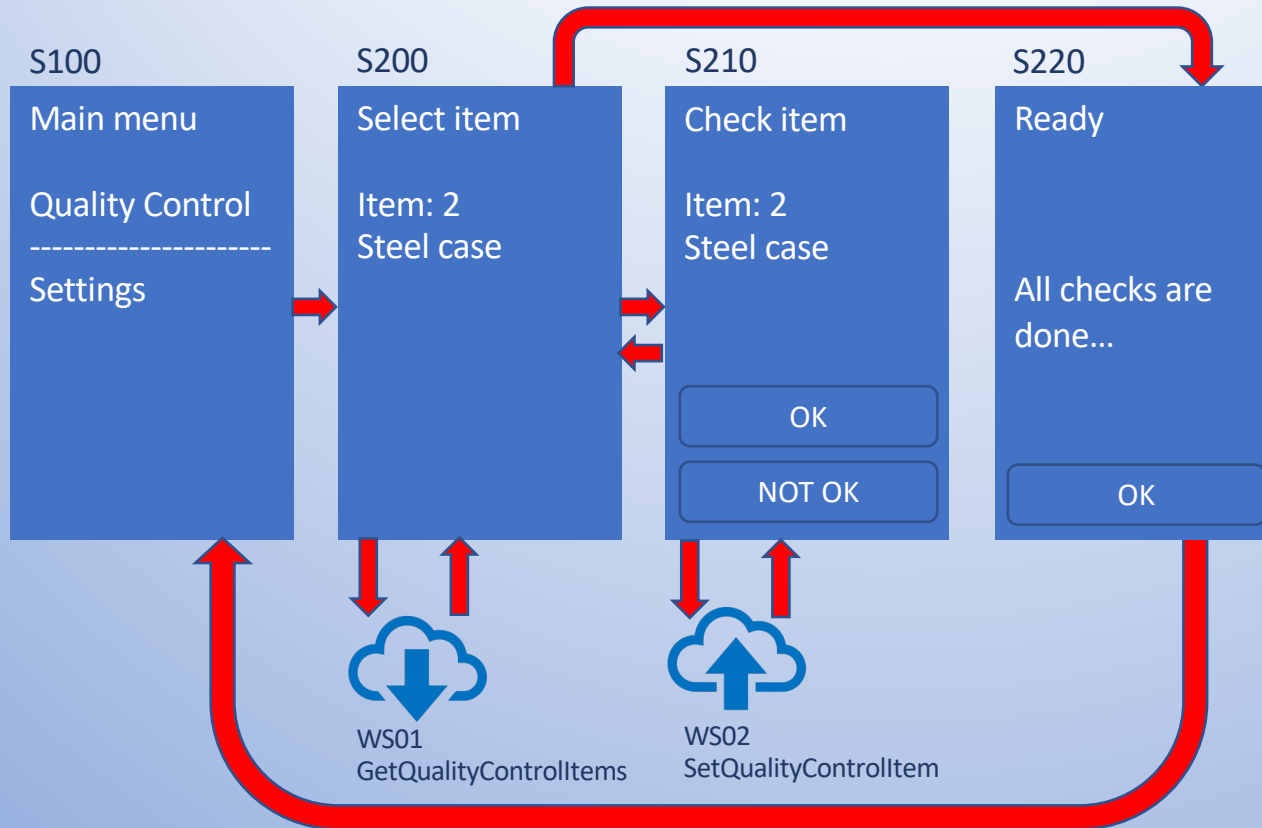
No quality check items found



1. Tap 'Quality Control'
2. Animate from S100 to S200
3. Execute WS01, show spinner
4. Animate from S200 to S230
5. Wait 2 seconds
6. Animate from S230 to S100

Combine step 2 and 4
Animate from S100 directly to S230

Only one item



1. Tap 'Quality Control'
2. Animate from S100 to S200
3. Execute WS01, show spinner
4. Animate from S200 to S210
5. Tap 'OK'
6. Execute WS02, show spinner
7. Animate from S210 to S200
8. Animate from S200 to S220
9. Wait 2 seconds
10. Animates from S220 to S100

Combine step 2 and step 4:
Animate from S100 directly to S210

Combine step 7 and step 8
Animates from S210 directly to
S220

Problem

If we want to separate concerns, we might have an issue with pages that animate quickly after each other.

If we want to solve that with Navigator 1.0, code becomes complicated and/or spaghetti and violates separation of concerns.

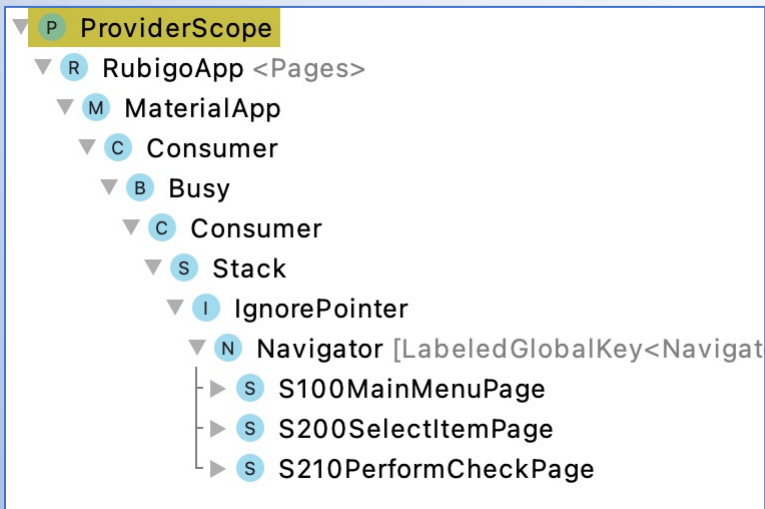
Solution

RubigoNavigator

A navigator, that informs Flutter's standard navigator about page changes when navigation is completed.

Made possible by Flutter's Navigator 2.0

Navigator 2.0

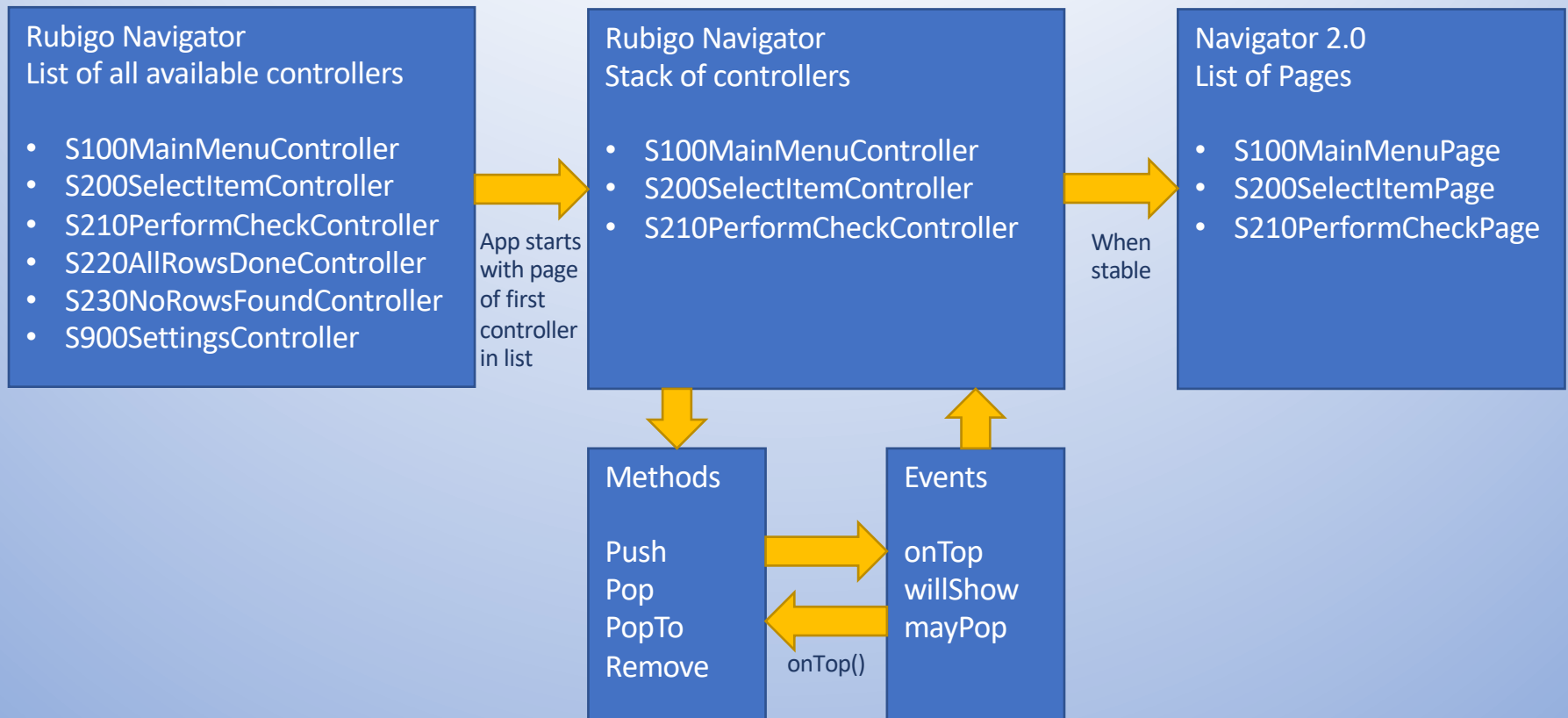


List of Pages

- Page(child: S100MainMenuPage());
- Page(child: S200SelectItemPage());
- Page(child: S210PerformCheckPage());

```
MaterialApp(  
  navigatorKey: _navigatorKey,  
  onGenerateRoute: (_) => null,  
  title: 'Flutter Navigator Demo',  
  builder: (context, _) {  
    return Consumer(  
      builder: (context, watch, child) {  
        return Busy(  
          child: Navigator(  
            key: _navigatorKey,  
            pages: watch(widget.navigatorProvider).pages,  
            onPopPage: context.read(widget.navigatorProvider).onPopPage,  
          ),  
        );  
      },  
    );  
  },  
);
```

Navigator 2.0 and RubigoNavigator



Implementation

Classes:

- **RubigoApp**
Convenience widget for easy initial app setup
- **RubigoNavigator<Pages>**
Navigator of type Pages. Strongly typed navigation
- **RubigoController**
Business logic and state
- **RubigoPage**
View

RubigoNavigator<Pages>

Methods:

- **pop()**
Pops the current page of the stack
- **push(enum id)**
Pushes the page with id on the stack
- **popTo(enum id)**
Pops directly to the specific page
- **remove(enum id)**
Removes page with id unconditionally from the stack

RubigoController (base class)

Properties:

- **rubigoNavigator**
Allows easy access to navigation

Events:

- **onTop(stackChange, previousPage)**
In this event the navigation can be continued (push/pop/popTo)
- **willShow()**
This event can be used to get data (also from async sources) for the page to display. The page transition will take place, right after this event.
- **bool mayPop()**
This function can be used to block a pop().

RubigoPage (base class)

Properties:

- **controllerProvider**
Allows easy access the corresponding Controller

Mason

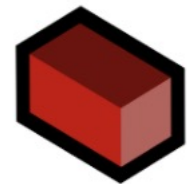
Use 'mason' to generate pairs of pages and controllers.

```
prompt> mason make rubigo_page

page_id: s300
page_name: item_search

✓ Made brick rubigo_page (5.6ms)
✓ Generated 2 file(s):

pages/s300_item_search/s300_item_search_controller.dart (new)
pages/s300_item_search/s300_item_search_page.dart (new)
```



mason

pub v0.0.1-dev.27 mason passing license MIT

Uploader

felangelov@gmail.com

Live coding

What can go wrong?

