



Codelab Flutter Devfest Nantes 2018

Horacio Gonzalez & Pierre Tibulle



@LostInBrittany



Codelab Flutter



@ptibulle

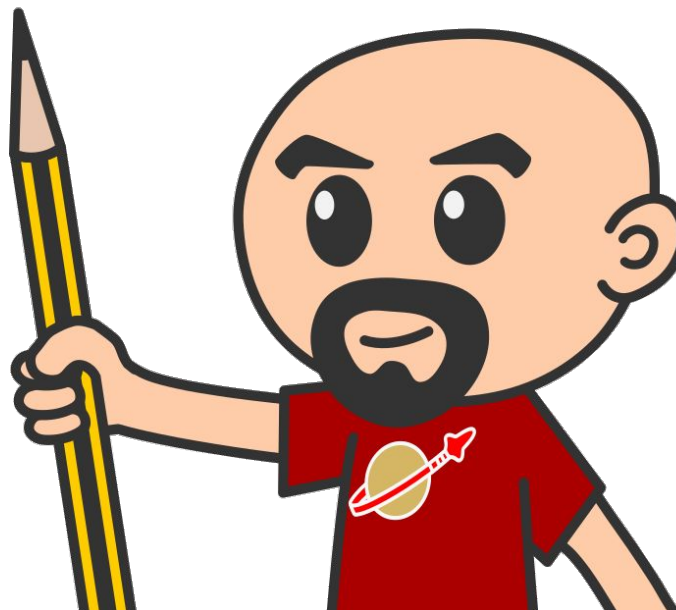


Horacio Gonzalez



@LostInBrittany

Spaniard lost in Brittany, developer,
dreamer and all-around geek



@LostInBrittany



Codelab Flutter

ASI @ptibulle

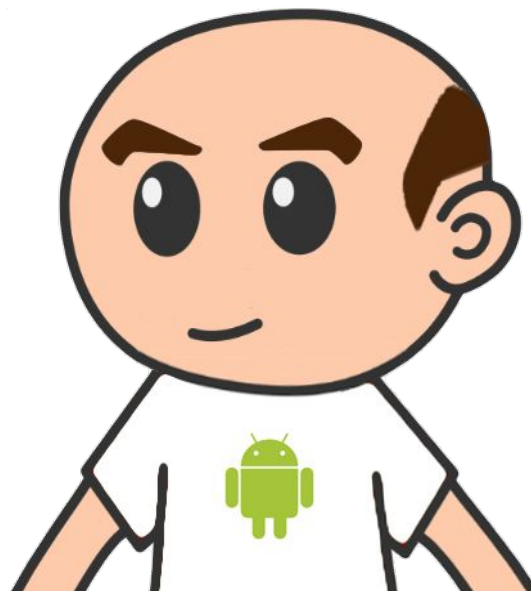


Pierre Tibulle



@ptibulle

Developer, Jobcrafter and Maker



@LostInBrittany



Codelab Flutter



@ptibulle



Before we begin



If you don't have these tools, download them:

- Flutter SDK

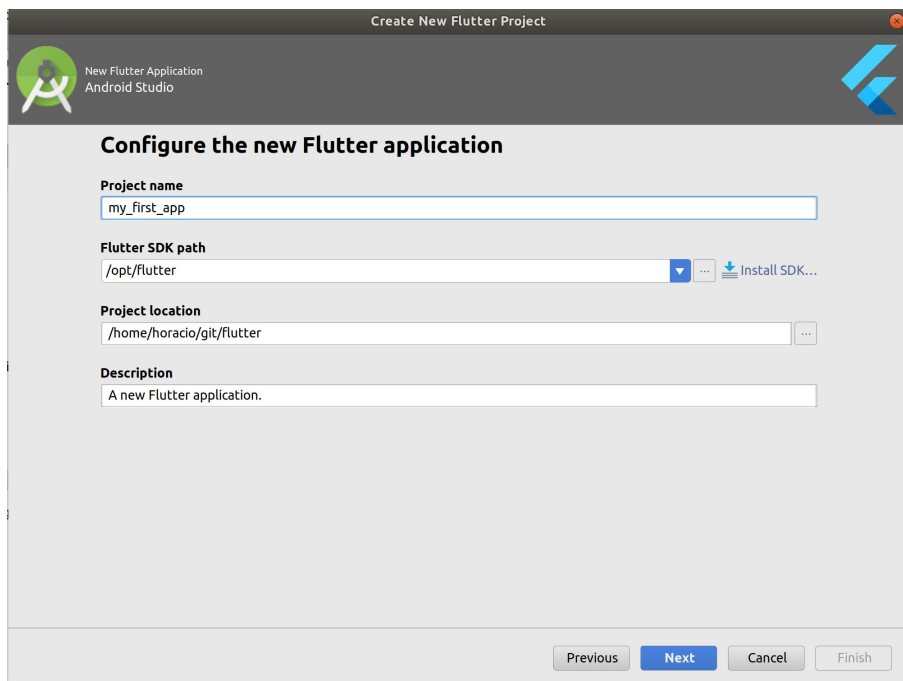
<https://flutter.io/get-started/install/>

- Android Studio or Visual Studio Code with Flutter plugins/extensions

<https://flutter.io/get-started/editor/#androidstudio>



Testing your install



Create and test your first app

<https://flutter.io/get-started/test-drive/>



@LostInBrittany



Codelab Flutter



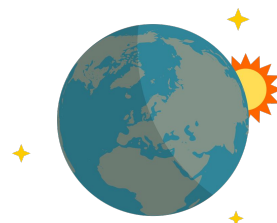
@ptibulle



URL to copy/paste the code



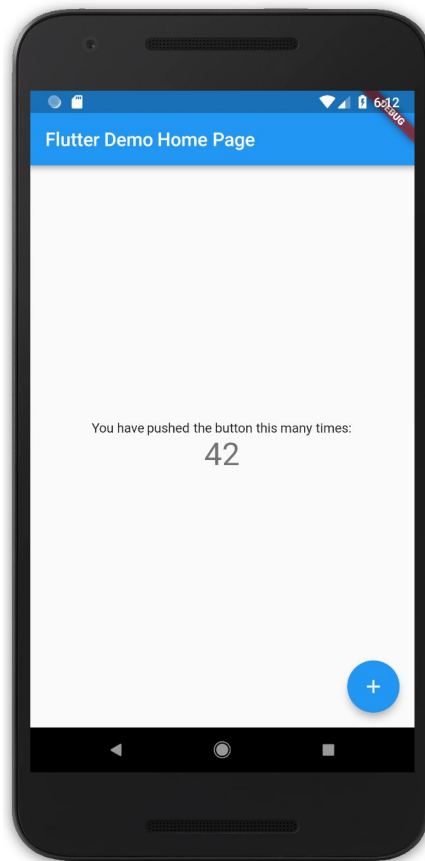
<https://goo.gl/Yqo3Am>



Let's look at the demo app



- The app is a stateless widget
- With a stateful widget inside
 - `createState()` method



Let's begin with a Hello World...



```
import 'package:flutter/material.dart';

void main() {
  runApp(
    new Center(
      child: new Text(
        'Hello, world!',
        textDirection: TextDirection.ltr,
      ),
    ),
  );
}
```

You will never do that...



Extending stateless widget



```
import 'package:flutter/material.dart';

class MyAppBar extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Center(
      child: new Text(
        'Hello, world!',
        textDirection: TextDirection.ltr,
      ),
    );
  }
}

void main() {
  runApp(new MyAppBar());
}
```

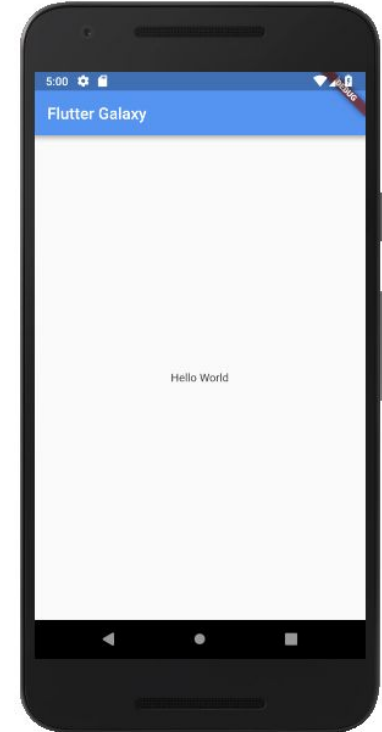
A widget's main job is to implement a build function



Material Design App



```
@override
Widget build(BuildContext context) {
  return new MaterialApp(
    title: 'Flutter Galaxy',
    home: new Scaffold(
      appBar: new AppBar(
        title: const Text('Flutter Galaxy'),
      ),
      body: const Center(
        child: const Text('Hello World'),
      ),
    ),
  );
}
```



Use the MaterialApp Widget



Add a Stateful Widget



Create a new widget in the main.dart

```
class GalaxiesState extends State<Galaxies> {  
  // TODO Add build method  
}  
  
class Galaxies extends StatefulWidget {  
  @override  
  GalaxiesState createState() => new GalaxiesState();  
}
```

Add the build method in the GalaxiesState class

```
@override  
Widget build(BuildContext context) {  
  final String galaxy = "Milky Way";  
  return new Text(galaxy, textDirection: TextDirection.ltr);  
}
```

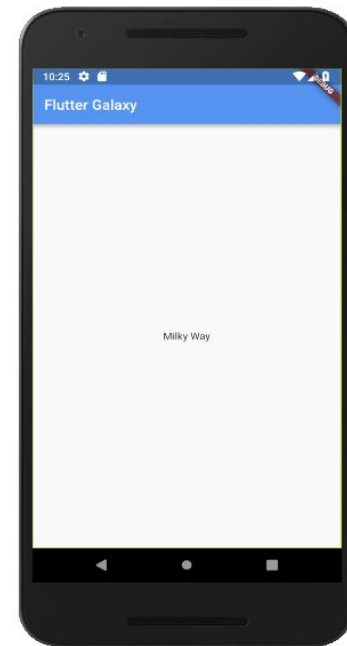


Use the new Widget



Update the build of MyApp to call the new widget

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return new MaterialApp(  
      title: 'Flutter Galaxy',  
      home: new Scaffold(  
        appBar: new AppBar(  
          title: const Text('Flutter Galaxy'),  
        ),  
        body: new Center( // const => new (no more a const)  
          child: new Galaxies(), // call the new widget.  
        ),  
      ),  
    );  
  }  
}
```



Create a scrolling listview 1/3



Create a list of galaxies and a TextStyle

```
class GalaxiesState extends State<Galaxies> {  
  final List<String> _galaxies = ["Milky Way Galaxy", "Andromeda Galaxy",  
                                   "Black Eye Galaxy", "Galaxy UGC10445"];  
  final TextStyle _biggerFont = const TextStyle(fontSize: 18.0);
```

Create a ListTile widget row as a method in GalaxyState

```
Widget _buildRow(String galaxy) {  
  return new ListTile(  
    title: new Text(  
      galaxy,  
      style: _biggerFont,  
    ),  
  );  
}
```



Create a scrolling listview 2/3



Then create another method to build the whole list:

```
Widget _buildGalaxies() {  
  return new ListView.builder(  
    padding: const EdgeInsets.all(16.0),  
    itemCount: _galaxies.length * 2,  
    itemBuilder: (BuildContext _context, int i) {  
      // Add a one-pixel-high divider on odd row  
      if (i.isOdd) return new Divider();  
      // Else build a galaxy row  
      final int index = i ~/ 2;  
      return _buildRow(_galaxies[index]);  
    }  
  );  
}
```



Create a scrolling listview 3/3

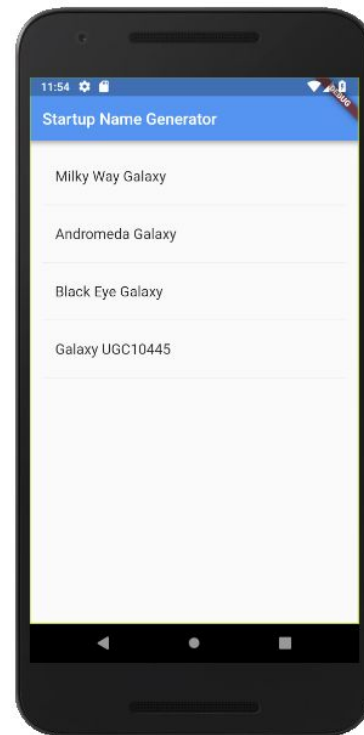


Update the build method of GalaxiesState

```
return new Scaffold (  
  appBar: new AppBar(  
    title: new Text('Flutter Galaxy'),  
  ),  
  body: _buildGalaxies(),  
);
```

Update the build method of MyApp

```
return new MaterialApp(  
  title: 'Flutter Galaxy',  
  home: new Galaxies(),  
);
```



Congratulation : Step 1 complete



Your first minimal flutter App is OK !

In case of emergency, you can download this step :

https://gitlab.com/ptibulle1/flutter_galaxy/tree/galaxy1



Import Real Data : Galaxies



Download the file “galaxies.json” :

https://gitlab.com/ptibulle1/flutter_galaxy/blob/master/assets/data/galaxies.json

and save it : ../assets/data/galaxies.json

Update the pubspec.yaml to add the new asset (**indent is important !**) :

```
...  
# To add assets to your application, add an assets section, like this:  
assets:  
  - assets/data/galaxies.json  
...
```



Create a galaxy class



Create a galaxy.dart file

```
class Galaxy {  
  final String id;  
  final String title;  
  final String previewHref;  
  final String description;  
  
  Galaxy({  
    this.id,  
    this.title,  
    this.previewHref,  
    this.description,  
  });  
}
```



Use Data : Galaxies 1/3



Import libraries and the new Galaxy class in the main.dart

```
import 'dart:convert';  
import 'dart:async';  
import 'galaxy.dart';
```

Load the JSON file in a new asynchronous function of GalaxiesState (Begin)

```
/// Retrieves a List of [galaxy]  
Future<void> _retrieveLocalGalaxies() async {  
  final json =  
    DefaultAssetBundle.of(context).loadString('assets/data/galaxies.json');  
  final data = JsonDecoder().convert(await json);  
  if (data is! Map) {  
    throw ('Data retrieved from API is not a Map');  
  }  
  var collection = data['collection'];  
  if (collection is! Map) {  
    throw ('Collection is not a Map');  
  }  
  ...  
}
```



Use Data : Galaxies 2/3



Load the json file in a new asynchronous function of GalaxiesState (end)

```
...  
var items = collection['items'];  
List<Galaxy> galaxies = [];  
  
for (var jsonMap in items) {  
  var links = jsonMap['links'][0];  
  var previewHref = links['href'];  
  var data = jsonMap['data'][0];  
  var title = data['title'];  
  var id = data['nasa_id'];  
  var description = data['description'];  
  var galaxy = new Galaxy(id: id, title: title, previewHref: previewHref,  
description: description);  
  galaxies.add(galaxy);  
}  
setState(() {  
  _galaxies.clear();  
  _galaxies.addAll(galaxies);  
});  
}
```



Use Data : Galaxies 3/3



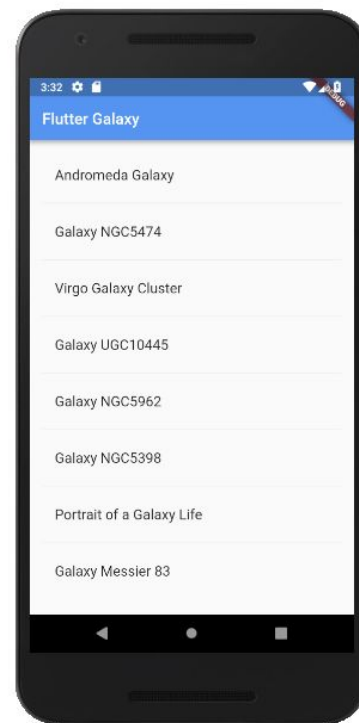
Use the Galaxy class instead of String

```
List<Galaxy> _galaxies = [];
```

```
Widget _buildRow(Galaxy galaxy) {  
  return new ListTile(  
    title: new Text(  
      galaxy.title,  
      style: _biggerFont,  
    ),  
  );  
}
```

Surcharge the lifecycle function

```
@override  
Future<void> didChangeDependencies() async {  
  super.didChangeDependencies();  
  if (_galaxies.isEmpty) {  
    await _retrieveLocalGalaxies();  
  }  
}
```



Congratulation : Step 2 complete



In case of emergency, you can download this step :

https://gitlab.com/ptibulle1/flutter_galaxy/tree/galaxy2

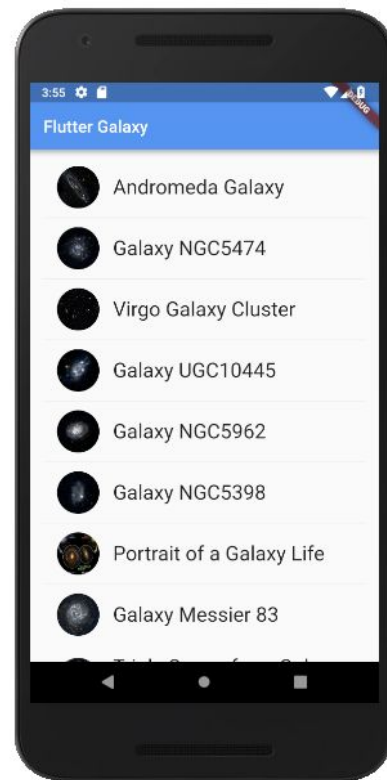


Add a preview image



Add a preview image in the leading part of the ListTile

```
Widget _buildRow(Galaxy galaxy) {  
  return new ListTile(  
    leading: Container(  
      width: 50.0,  
      height: 50.0,  
      decoration: new BoxDecoration(  
        shape: BoxShape.circle,  
        image: new DecorationImage(  
          fit: BoxFit.fill,  
          image: NetworkImage(galaxy.previewHref),  
        ),  
      ),  
    ),  
  ),  
  ...  
}
```



Make it “clickable”



Put the `ListTile` in an `InkWell` widget

```
Widget _buildRow(Galaxy galaxy) {  
  return InkWell(  
    borderRadius: BorderRadius.circular(8.0),  
    onTap: () => null,  
    child: new ListTile(  
      ...  
    ),  
  );  
}
```



Navigate to the galaxy !

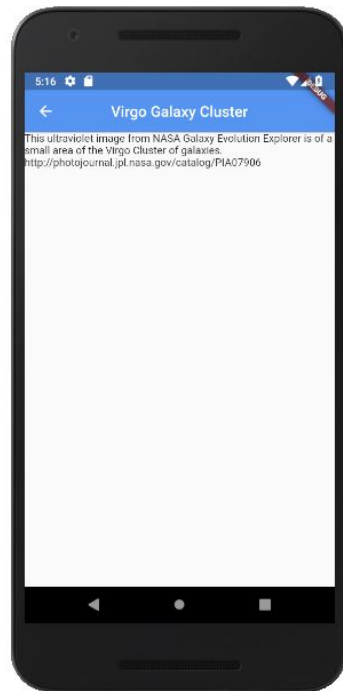


Use the “navigator” in a new function

```
void _navigateToGalaxy(BuildContext context, Galaxy galaxy) {  
  Navigator.of(context).push(MaterialPageRoute<Null>(  
    builder: (BuildContext context) {  
      return Scaffold(  
        appBar: AppBar(  
          elevation: 1.0,  
          title: Text(galaxy.title),  
          centerTitle: true,  
        ),  
        body: new Text(galaxy.description),  
      );  
    },  
  ));  
}
```

Call the function in the tap of the InkWell

```
onTap: () => _navigateToGalaxy(context, galaxy),
```



Create a galaxy route widget



Create a file `galaxy_route.dart`

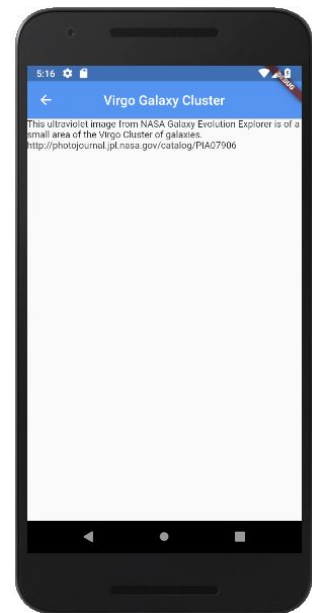
```
import 'package:flutter/material.dart';
import 'galaxy.dart';

class GalaxyRoute extends StatelessWidget {
  final Galaxy galaxy;
  const GalaxyRoute({
    @required this.galaxy,
  }) : assert(galaxy != null);
  @override
  Widget build(BuildContext context) {
    return new Text(galaxy.description);
  }
}
```

Import and call this class in `main.dart`

```
import 'galaxy_route.dart';

void _navigateToGalaxy(BuildContext context, Galaxy galaxy) {
  ...
  body: GalaxyRoute(galaxy: galaxy),
}
```



Improve the galaxy route widget



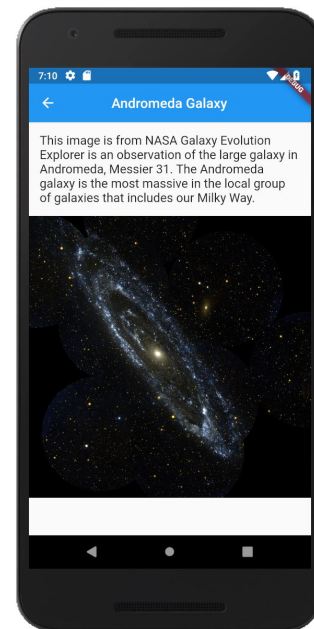
Create a bigger font

```
final TextStyle _biggerFont = const TextStyle(fontSize: 18.0);
```

And build the widget

```
@override
Widget build(BuildContext context) {
  String urlImage = galaxy.previewHref.replaceAll("thumb", "orig");

  return new ListView(
    children: <Widget>[
      new Padding(
        padding: const EdgeInsets.all(16.0),
        child: new Text(galaxy.description,
          style: _biggerFont,
        )),
      Image.network(urlImage),
    ],
  );
}
```



The image is loading...



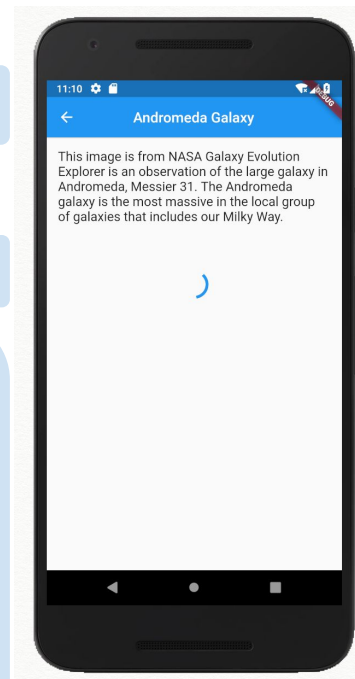
Import a package in pubspec.yaml

```
transparent_image: ^0.1.0
```

And build the widget with a progress indicator

```
import 'package:transparent_image/transparent_image.dart';
```

```
// Image.network(urlImage),
new Stack(
  children: <Widget>[
    new Padding(
      padding: const EdgeInsets.all(50.0),
      child: Center(child: CircularProgressIndicator())
    ),
    Center(
      child: FadeInImage.memoryNetwork(
        placeholder: kTransparentImage,
        image: urlImage
      ),
    ),
  ],
),
```



Congratulation : Step 3 complete



In case of emergency, you can download this step :

https://gitlab.com/ptibulle1/flutter_galaxy/tree/galaxy3



Fetch data from the internet



Import http package in pubspec.yaml

```
http: ^0.11.0
```

and in main.dart call the nasa api and use a spinner while the data is empty

```
import 'package:http/http.dart' as http;
```

```
Future<void> _retrieveLocalGalaxies() async {  
  final response = await http.get('https://images-api.nasa.gov/search?q=galaxy&media_type=image');  
  var json;  
  if (response.statusCode == 200) {  
    json = response.body;  
  } else {  
    json = DefaultAssetBundle.of(context).loadString('assets/data/galaxies.json');  
  }  
  ...  
}
```

```
Widget _buildGalaxies() {  
  if (_galaxies.isEmpty) return new Padding(padding: const EdgeInsets.all(50.0),  
    child: Center(child: CircularProgressIndicator()));  
  ...  
}
```



A zoomable image page 1/2



Import a package in pubspec.yaml

```
zoomable_image: ^1.2.1
```

Create a galaxy_zoom.dart file

```
import 'package:flutter/material.dart';
import 'package:zoomable_image/zoomable_image.dart';
import 'galaxy.dart';

class GalaxyZoom extends StatelessWidget {
  final Galaxy galaxy;

  const GalaxyZoom({
    @required this.galaxy,
  }) : assert(galaxy != null);

  @override
  Widget build(BuildContext context) {
    String urlImage = galaxy.previewHref.replaceAll("thumb", "orig");
    return new ZoomableImage(NetworkImage(urlImage));
  }
}
```



A zoomable image page 2/2



Update galaxy_route.dart

```
import 'galaxy_zoom.dart';
```

```
/// Navigates to the [GalaxyZoom].
void _navigateToGalaxyZoom(BuildContext context, Galaxy galaxy) {
  Navigator.of(context).push(MaterialPageRoute<Null>(
    builder: (BuildContext context) {
      return Scaffold(
        appBar: AppBar(elevation: 1.0, title: Text(galaxy.title), centerTitle: true, ),
        body: GalaxyZoom(galaxy: galaxy),
      );
    },
  ));
}
```

```
// Detect Tap on the image
new GestureDetector(
  onTap: () => _navigateToGalaxyZoom(context, galaxy),
  child: new Stack(
    ...
  ),
),
```



Create a dark galaxy theme

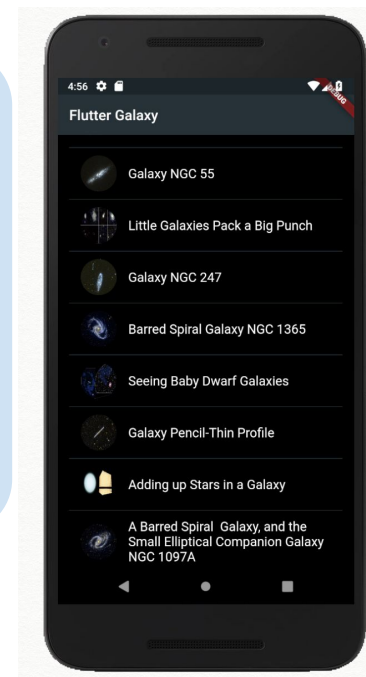


In the main.dart add a ThemeData to the MaterialApp

```
return new MaterialApp(  
  title: 'Flutter Galaxy',  
  theme: ThemeData(  
    brightness: Brightness.dark, // Default style for widgets  
    primaryColor: Colors.blueGrey[900], // AppBar  
    dividerColor: Colors.blueGrey[600], // Divider  
    scaffoldBackgroundColor: Colors.black, // Background  
    splashColor: Colors.blueGrey[200], // InkWell  
    accentColor: Colors.pink, // spinner  
  ),  
  home: new Galaxies(),  
);
```

All pages are impacted by the Theme ...

Modify the colors and play with the hot reload !



https://gitlab.com/ptibulle1/flutter_galaxy/tree/galaxy4



Congratulation : Step 4 complete



This is the end of the codelab !

In case of emergency, you can download this step :

https://gitlab.com/ptibulle1/flutter_galaxy/tree/galaxy4



