

## Atelier 5

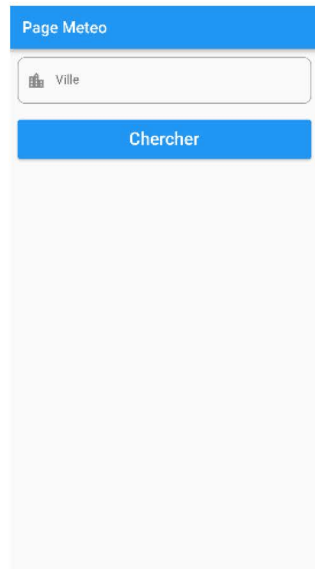
### Pages Meteo & Meteo-Details

**Matière : ATELIER FRAMEWORK CROSS-PLATFORM**

**DSI3**

**Enseignants : S. Hadhri & M. Hadiji**

L'objectif de cet atelier est la réalisation de 2 pages Meteo et Meteo-Details :

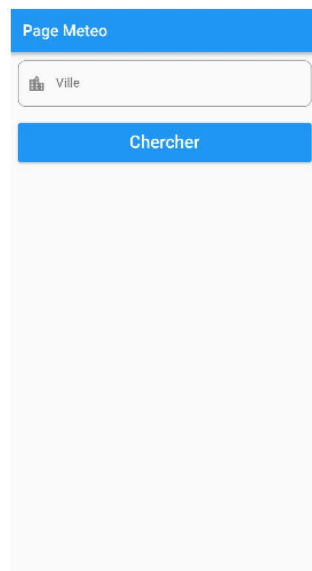


**Page Meteo**

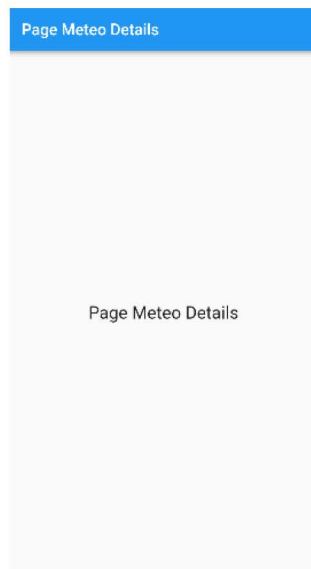


**Page Meteo-Details**

1. Créer dans la page Meteo un formulaire formé par une zone de texte et un bouton.



2. Sous le dossier « lib/pages », créer une nouvelle page « *meteo-details.page.dart* » contenant un texte centré « Page Meteo Details ».



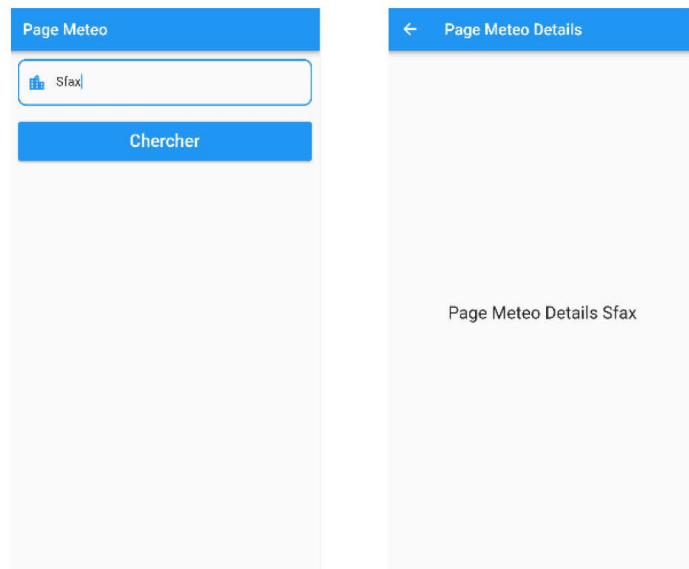
3. Développer la méthode privée `_onGetMeteoDetails()` appelée au clic sur le bouton « Chercher » qui permet de basculer vers la page Meteo-Details en lui passant la ville saisie. Vérifier que le passage s'est bien passé en affichant la ville saisie dans la page Meteo-Details.

#### Fichier meteo.page.dart

```
...  
void _onGetMeteoDetails(BuildContext context) {  
  String v=txt_ville.text;  
  Navigator.push(  
    context,  
    MaterialPageRoute(  
      builder: (context) => MeteoDetailsPage(v)),  
    txt_ville.text = "";  
  )  
}  
...
```

#### Fichier meteo-details.page.dart

```
import 'package:flutter/material.dart';  
  
class MeteoDetailsPage extends StatelessWidget {  
  String ville = "";  
  
  MeteoDetailsPage(this.ville);  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: Text('Page Meteo Details ${ville}')),  
      body: Center(  
        child: Text(  
          'Page Meteo Details ${ville}',  
          style: TextStyle(fontSize: 22),  
        )),  
    );  
  }  
}
```



- Convertir la page Meteo-Details en StatefulWidget puis redéfinir la méthode initState() pour qu'elle appelle une fonction getMeteoData(ville).

**Fichier meteo-details.page.dart**

```

...
void initState() {
  super.initState();
  getMeteoData(widget.ville);
}

void getMeteoData(String ville) {}
...
appBar: AppBar(title: Text('Page Meteo Details  ${widget.ville}')),
...
'Page Meteo Details ${widget.ville}',
...

```

## 5. Présentation du service OpenWeatherMap

OpenWeatherMap est un service en ligne qui fournit des données météorologiques actuelles, des prévisions et des données historiques.

Plus de vingt API météorologiques ont été développées pour obtenir différents types de données météorologiques. Ils prennent en charge plusieurs langues, unités de mesure et formats de données.

### Remarque :

La page de documentation de l'API est accessible <https://openweathermap.org/api>.

Utiliser openweathermap nécessite de s'enregistrer afin d'obtenir une clé API (créer votre compte)

➔ Le service OpenWeatherMap fournit une API de météo actuelle, ainsi que plusieurs types de prévisions avec différentes profondeurs et étapes de mesure. Dans cet atelier, nous allons utiliser l'API de prévision par étapes de 3 heures pour 5 jours :

<https://api.openweathermap.org/data/2.5/forecast?q={city name}&appid={your api key}>

Voici un exemple de réponse JSON de ce service :

```
{
  "cod": "200",
```

```

"message": 0,
"cnt": 40,
"list": [
  {
    "dt": 1659204000,
    "main": {
      "temp": 307.23,
      "feels_like": 308.72,
      "temp_min": 304.6,
      "temp_max": 307.23,
      "pressure": 1013,
      "sea_level": 1013,
      "grnd_level": 1012,
      "humidity": 40,
      "temp_kf": 2.63
    },
    "weather": [
      {
        "id": 800,
        "main": "Clear",
        "description": "clear sky",
        "icon": "01d"
      }
    ],
    "clouds": {
      "all": 0
    },
    "wind": {
      "speed": 4.84,
      "deg": 114,
      "gust": 5.5
    },
    "visibility": 10000,
    "pop": 0,
    "sys": {
      "pod": "d"
    },
    "dt_txt": "2022-07-30 18:00:00"
  },
  ...
],
"city": {
  "id": 2467454,
  "name": "Sfax",
  "coord": {
    "lat": 34.7406,
    "lon": 10.7603
  },
  "country": "TN",
  "population": 277278,
  "timezone": 3600,
  "sunrise": 1659155109,
  "sunset": 1659205287
}
}

```

6. Coder la fonction `getMeteoData(ville)` qui permet de lancer une requête http et récupérer l'état de la météo sous format JSON. Vérifier la réception de la réponse en l'affichant dans la console.

### **Rappel :**

Le package `http` (<https://pub.dev/packages/http>) fournit le moyen le plus simple de récupérer des données sur Internet. Pour pouvoir l'utiliser, il faudrait ajouter dans la partie « dependencies » du fichier « `pubspec.yaml` » la ligne suivante :

```
dependencies:
  shared_preferences:
  http:
  flutter:
  sdk: flutter
```

Une fois la dépendance ajoutée, importer le package dans le fichier Dart :

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

Ensuite, utiliser la méthode `get()` qui renvoie un objet de type `Future` ayant comme retour un `Response` représentant les données reçues d'un appel HTTP réussi.

```
http.get(Uri.parse(url)).then((resp) {
  });
}
```

Enfin, convertir la réponse en un objet Dart personnalisé grâce à la méthode `decode()`:

```
json.decode(resp.body);
```

#### **Fichier meteo-details.page.dart**

```
...
import 'package:http/http.dart' as http;
import 'dart:convert';
...
class _MeteoDetailsPageState extends State<MeteoDetailsPage> {
  var meteoData;
  ...
  void getMeteoData(String ville) {
    print("Météo de la ville de " + ville);
    String url =
    "https://api.openweathermap.org/data/2.5/forecast?q=${ville}&appid=c109c07bc4d
f77a88c923e6407aef864";
    http.get(Uri.parse(url)).then((resp) {
      this.meteoData = json.decode(resp.body);
      print(this.meteoData);
    }).catchError((err) {
      print(err);
    });
  }
  ...
}
```

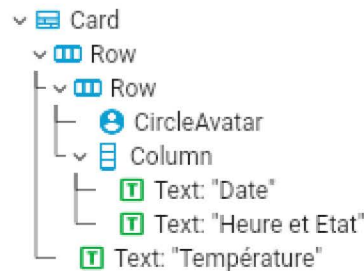
#### **Console**

```
Météo de la ville de sfax
{cod: 200, message: 0, cnt: 40, list: [{dt: 1659204000, main: {temp: 307.23,
feels_like: 308.72, temp_min: 304.6, temp_max: 307.23, pressure: 1013,
sea_level: 1013, grnd_level: 1012, humidity: 40, temp_kf: 2.63}, weather: [{id:
800, main: Clear, description: clear sky, icon: 01d}], clouds: {all: 0}, wind:
{speed: 4.84, deg: 114, gust: 5.5}, visibility: 10000, pop: 0, sys: {pod: d},
dt_txt: 2022-07-30 18:00:00}, {dt: 1659214800, main: {temp: 306.17, feels_like:
307.88, temp_min: 304.04, temp_max: 306.17, pressure: 1014, sea_level: 1014,
grnd_level: 1015, humidity: 44, temp_kf: 2.13}, weather: [{id: 800, main: Clear,
description: clear sky, icon: 01n}], clouds: {all: 0}, wind: {speed: 1.67, deg:
```



```
80, gust: 1.8}, visibility: 10000, pop: 0, sys: {pod: n}, dt_txt: 2022-07-30
21:00:00}, {dt: 1659225600, main: {temp: 304.79, feels_like: 306.22, temp_min:
303.57, temp_max: 304.79, pressure: 1014, sea_level: 1014, grnd_level: 1014,
humidity: 47, temp_kf: 1.22}, weather: [{id: 800, main: Clear, description:
clear sky,
...
```

7. Réaliser l'aperçu de la page Meteo-Details. Cette page est formée d'un ListView dont chaque élément représente l'état de la météo à une heure donnée. Chaque élément de ce ListView est représenté par un Card dont l'arbre des widgets est le suivant :



#### Fichier meteo-details.page.dart

```
...
class _MeteoDetailsPageState extends State<MeteoDetailsPage> {
  var meteoData;

  @override
  void initState() {
    ...
  }

  void getMeteoData(String ville) {
    ...
    setState(() {
      this.meteoData = json.decode(resp.body);
      print(this.meteoData);
    });
  }
  ...
  body: ListView.builder(
    itemCount: (meteoData == null ? 0 : meteoData['list'].length),
    itemBuilder: (context, index) {
      return Card(
        color: Colors.blue,
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: <Widget>[
            Row(
              children: [
                //Ce widget sera développé dans la question 8
                CircleAvatar(),
                Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    //Ce widget sera développé dans la question 9
                    Text("Date"),
                    //Ce widget sera développé dans la question 10
                    Text("Heure et Etat"),
                  ],
                ),
              ],
            ),
          ],
        ),
      ),
    ],
  ),
),
```

```

//Ce widget sera développé dans la question 11
Text("Température"),
],
),
});
...

```



8. Afficher dans le CircleAvatar l'image correspondant à l'état de la météo. Utiliser le widget AssetImage et insérer les images nécessaires dans votre projet.

**Fichier meteo-details.page.dart**

```

.....
CircleAvatar(
  backgroundImage: AssetImage(
    "images/${meteoData['list'][index]['weather'][0]['main'].toString().toLowerCase().png}"
  )
)
.....

```

9. Afficher la date de la météo sous format jj/mm/aaaa. Utiliser le package intl : <https://pub.dev/packages/intl>

**Fichier meteo-details.page.dart**

```

.....
Text(
  "${new DateFormat('E-dd/MM/yyyy').format(DateTime.fromMicrosecondsSinceEpoch(meteoData['list'][index]['dt'] * 1000000))}",
  style: TextStyle(
    fontSize: 22, fontWeight: FontWeight.bold
  )
)
.....

```

10. Afficher l'heure sous format HH:mm ainsi que l'état de la météo.

**Fichier meteo-details.page.dart**

```

.....
Text(...)
.....

```

11. Afficher la température en celcius.

## Fichier meteo-details.page.dart

```
.....
Text ( ... )
.....
```

← Page Meteo Details sfax		
	Mon-01/08/2022 06:00   Clear	28 °C
	Mon-01/08/2022 09:00   Clear	31 °C
	Mon-01/08/2022 12:00   Clear	32 °C
	Mon-01/08/2022 15:00   Clouds	31 °C
	Mon-01/08/2022 18:00   Clouds	30 °C
	Mon-01/08/2022 21:00   Clouds	30 °C
	Tue-02/08/2022 00:00   Clouds	29 °C
	Tue-02/08/2022 03:00   Clear	28 °C
	Tue-02/08/2022 06:00   Clear	28 °C
	Tue-02/08/2022 09:00   Clear	31 °C
	Tue-02/08/2022 12:00   Clouds	32 °C

12. Ajouter les espaces nécessaires (padding) entre les différents widgets afin de soigner l’affichage.

← Page Meteo Details sfax		
	Mon-01/08/2022 12:00   Clear	32 °C
	Mon-01/08/2022 15:00   Clear	31 °C
	Mon-01/08/2022 18:00   Clouds	30 °C
	Mon-01/08/2022 21:00   Clouds	30 °C
	Tue-02/08/2022 00:00   Clouds	29 °C
	Tue-02/08/2022 03:00   Clear	28 °C
	Tue-02/08/2022 06:00   Clouds	28 °C

13. Ajouter un Indicateur de progression circulaire qui s’affiche lors du chargement des données

## Fichier meteo-details.page.dart

```
.....
body: meteoData == null?
      Center(
        child: CircularProgressIndicator(),
      ):
      ListView.builder
.....
```



