

Great Places

Widget (Stateless, Stateful)

MyApp() - Stateless

- `ChangeNotifierProvider.value()`
 - value: **GreatPlaces()**
 - child: `MaterialApp()`
 - title: 'Great Places'
 - home: `PlaceListScreen()`
 - routes:
 - `AddPlaceScreen()`
 - `PlaceDetailScreen()`

PlaceListScreen() - Stateless

- `Scaffold()`
 - `appBar: AppBar()`
 - title: `Text()` -> 'Your Places'
 - actions: `IconButton(+)` => navigator -> `AddPlaceScreen()`
 - `body: FutureBuilder()`
 - future: **Function** **GreatPlaces**.`fetchAndSetPlaces()`
 - builder: `ConnectionState.waiting`
 - **true**: `Center()`
 - `CircularProgressIndicator()`
 - **false**: `Consumer<GreatPlaces>()`
 - **GreatPlaces**.items.isEmpty
 - **true**: `Center()`
 - `Text()` -> 'Got no places yet'
 - **false**: `ListView.builder()`
 - itemCount: **GreatPlaces**.items.length
 - itemBuilder: `ListTile()` => navigate -> `PlaceDetailScreen(GreatPlaces.items.id)`
 - leading: `CircleAvatar()`
 - `FileImage()` - **GreatPlaces**.items.image

- title: Text() -> **GreatPlaces.items.title**
- subtitle: Text() -> **GreatPlaces.items.address**

AddPlaceScreen() - Stateful

- Scaffold()
 - appBar: AppBar()
 - title: Text() -> 'Add a new Place'
 - body: Column()
 - Expanded()
 - SingleChildScrollView()
 - Padding()
 - Column()
 - TextField() -> 'Title'
 - SizedBox()
 - ImageInput() => **Function** selectImage()
 - SizedBox()
 - LocationInput() => **Function** selectPlace()
 - ElevatedButton.icon(+) => **Function** savePlace()
 - Text() -> 'Add Place'

PlaceDetailScreen() - Stateless

- Scaffold()
 - appBar: Appbar()
 - title: Text() -> **selectedPlace.title**
 - body: Column()
 - SizedBox()
 - Image.file()
 - **selectedPlace.image**
 - SizedBox()
 - Text() -> **selectedPlace.address**
 - SizedBox()
 - TextButton() => navigate -> MaterialPageRoute()
 - MapScreen() -> **selectedPlace.location**
 - Text() -> 'View on Map'

ImageInput(Function selectImage()) - Stateful

- Row()
 - Container()
 - storedImageFound
 - **true**: Image.file() -> storedImage
 - **false**: Text() -> 'No Image Taken'
 - SizedBox()
 - Expanded()
 - TextButton.icon(camera) => **Function** takePicture()
 - Text() -> 'Take Picture'

LocationInput(Function selectPlace()) - Stateful

- Column()
 - Container()
 - previewImageAvailable
 - **true**: Image.network() -> previewImageUrl
 - **false**: Text() -> 'No Location'
 - Row()
 - TextButton.icon(location_on) => **Function** getCurrentUserLocation()
 - Text() -> 'Current Location'
 - TextButton.icon(map) => **Function** selectOnMap()
 - Text() -> 'Select on Map'

Functions

GreatPlaces.fetchAndSetPlaces()

- getData
- Items = **Place**(id, title, image, **PlaceLocation**(lat, long, address))
- notifyListeners()

selectPlace(lat, lng)

- pickedLocation -> **PlaceLocation**(lat, lng)

selectImage(pickedImage)

- setPickedImage

savePlace()

- **Function** GreatPlaces.addPlace(title, image, location)

- Navigate -> pop()

takePicture()

- takePicture
- **Function** selectImage(imageTaken)

getCurrentUserLocation()

- getLocationData
- **Function** showPreview(lat, lng)
- **Function** selectPlace(lat, lng)

selectOnMap()

- selectedLocation -> navigate -> MaterialPageRoute()
 - MapScreen() -> selectingTrue
- **Function** showPreview(lat, lng)
- **Function** selectPlace(lat, lng)

GreatPlaces.addPlace(title, image, location)

- address -> **LocationHelper**.getPlaceAddress(lat, lng)
- updatedLocation -> **PlaceLocation**(lat, lng, address)
- newPlace -> **Place**(id, image, title, updateLocation)
- items.add(newPlace)
- notifyListeners()

showPreview(lat, lng)

- imageUrl -> **Function LocationHelper**.generateLocationPreviewImage(lat, lng)
- previewImageUrl -> imageUrl

LocationHelper.getPlaceAddress(lat, lng)

- returnFormattedAddress

LocationHelper.generateLocationPreviewImage(lat, lng)

- returnLocationUrl