MapLibre Configuration

This guide will explain various ways to create a map.

When working with maps, you likely want to configure the `MapView`.

There are several ways to build a `MapView`:

1. Using existing XML namespace tags for`MapView` in the layout.
2. Creating `MapLibreMapOptions` and passing builder function values into the `MapView`.
3. Creating a `SupportMapFragment` with the help of `MapLibreMapOptions`.

Before diving into `MapView` configurations, let's understand the capabilities of both XML namespaces and `MapLibreMapOptions`.

Here are some common configurations you can set:

- Map base URI
- Camera settings
- Zoom level
- Pitch
- Gestures
- Compass
- Logo
- Attribution
- Placement of the above elements on the map and more

We will explore how to achieve these configurations in XML layout and programmatically in Activity code, step by step.

`MapView` Configuration with an XML layout

To configure `MapView` within an XML layout, you need to use the right namespace and provide the necessary data in the layout file.

```
{{#include
../../../../platform/android/MapLibreAndroidTestApp/src/main/res/layout/
activity_map_options_xml.xml}}
```

This can be found in `activity_map_options_xml.xml`.

You can assign any other existing values to the `maplibre...` tags. Then, you only need to create `MapView` and `MapLibreMap` objects with a simple setup in the Activity.

```
{{#include
../../../../platform/android/MapLibreAndroidTestApp/src/main/java/org/ma
plibre/android/testapp/activity/options/MapOptionsXmlActivity.kt}}
```

This can be found in `MapOptionsXmlActivity.kt`.

`MapView` configuration with `MapLibreMapOptions`

Here we don't have to create MapView from XML since we want to create it programmatically.

```
{{#include
../../../../platform/android/MapLibreAndroidTestApp/src/main/res/layout/
activity_map_options_xml.xml}}
```

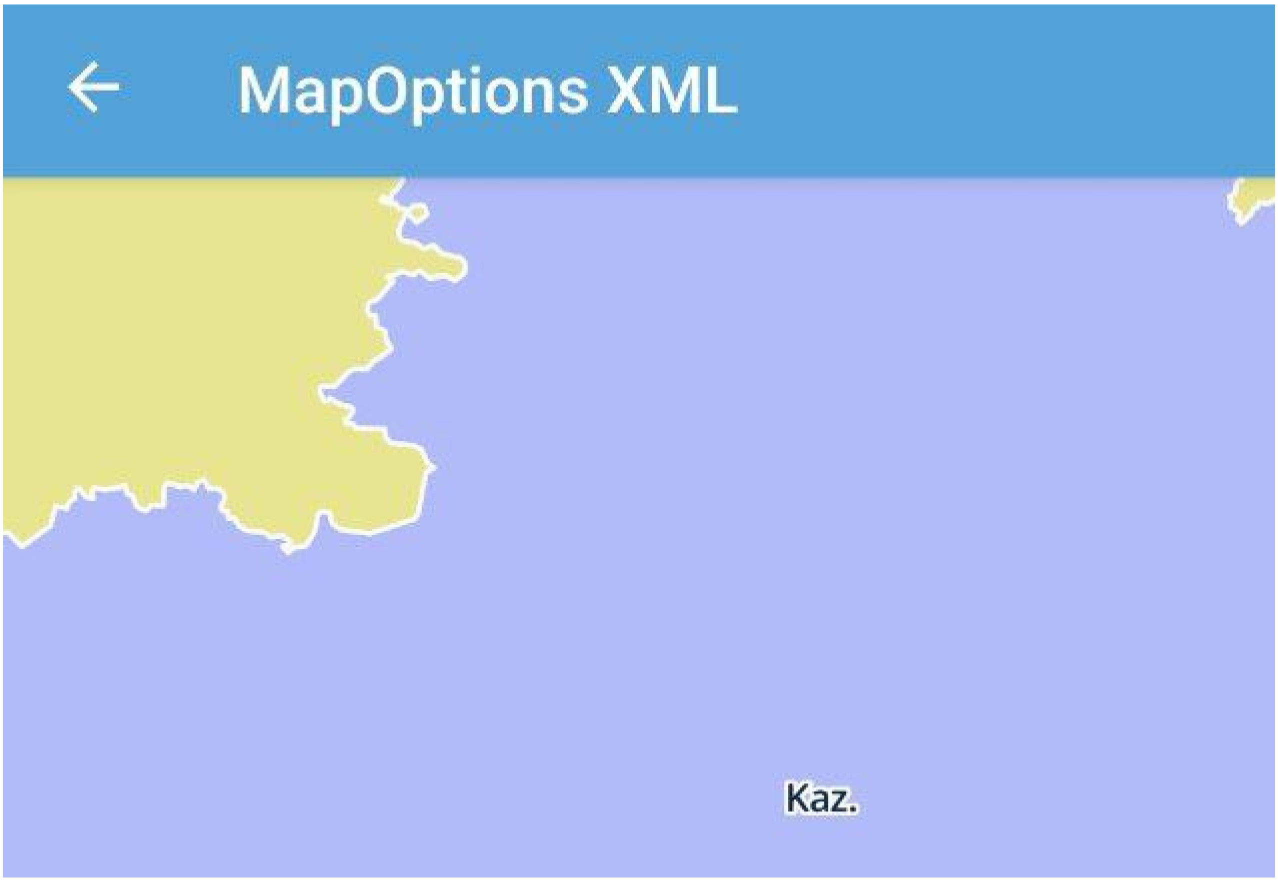
This can be found in `activity_map_options_runtime.xml`.

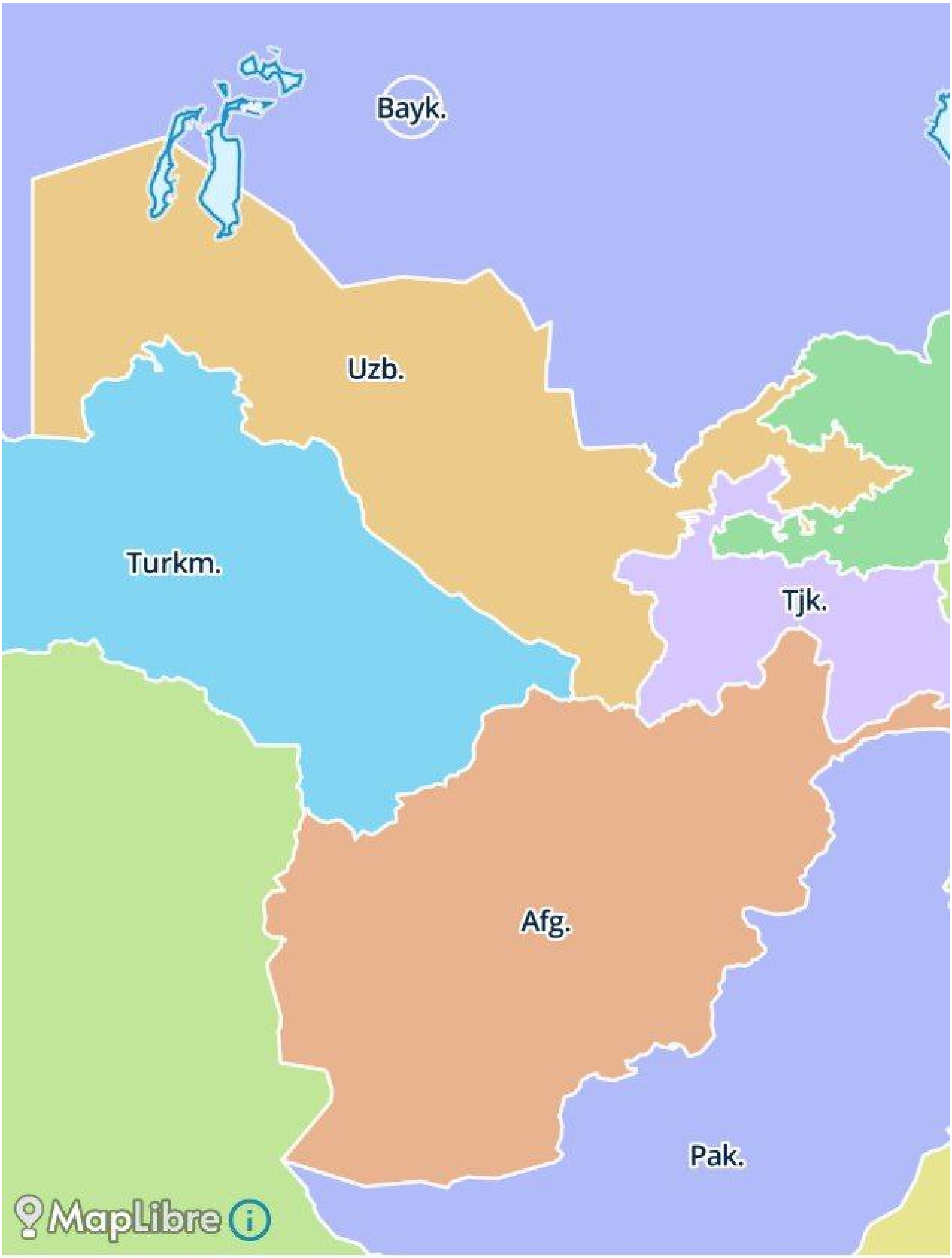
A `MapLibreMapOptions` object must be created and passed to the MapView constructor. All setup is done in the Activity code:

```
{{#include
../../../../platform/android/MapLibreAndroidTestApp/src/main/java/org/ma
plibre/android/testapp/activity/options/MapOptionsRuntimeActivity.kt}}
```

This can be found in `MapOptionsRuntimeActivity.kt`.

Finally you will see a result similar to this:





For the full contents of `MapOptionsRuntimeActivity` and `MapOptionsXmlActivity`, please take a look at the source code of MapLibreAndroidTestApp.

You can read more about `MapLibreMapOptions` in the Android API documentation.

`SupportMapFragment` with the help of `MapLibreMapOptions`.

If you are using MapFragment in your project, it is also easy to provide initial values to the `newInstance()` static method of `SupportMapFragment`, which requires a `MapLibreMapOptions` parameter.

Let's see how this can be done in a sample activity:

```
{{#include ../../../../platform/android/MapLibreAndroidTestApp/src/main/java/org/maplibre/android/testapp/activity/fragment/SupportMapFragmentActivity.kt}}
```

You can also find the full contents of `SupportMapFragmentActivity` in the MapLibreAndroidTestApp.

To learn more about `SupportMapFragment`, please visit the Android API documentation.