Using custom maps in native Android apps

21 January 2015 | **Software Engineering** | **Riina Pakarinen**, **Alexander Pacha**
Reading time: 6 minutes

In this post I will write about our experience using maps in native Android apps and how to make them available in offline scenarios. There are a few things to consider if you plan to implement such a feature. But before diving into the details, I first want to go through some design issues for native offline apps.

Web or native solution?

I was once asked, if a program using maps for mobile devices should be implemented as a web site or as a native app. The simple answer would be to ask: “Isn’t any new program nowadays implemented as a web solution?” The better answer is that both solutions have their advantages and drawbacks.

Main advantage of a web solution is that one solution works on many different device types. Even if there are platforms like Xamarin for the cross-platform development of native apps – the usage of third party libraries for map access can still be tricky. Another advantage of a web solution is that even complicated operations can be processed very fast via server-side map services. In native solutions every operation is usually processed locally on the device and depends on the device’s memory and process capabilities. Main advantage of a native solution is that when connectivity cannot be guaranteed, the application still runs. Another advantage is performance when loading tiles and features.

Offline or online usage?

An important question is: When is an offline solution needed? The answer is for example in locations, where the connectivity is low, or inside a construct, like a tunnel, where the connectivity is weak. It is also possible that the users choose to be offline or must be offline, like in an airplane.

When considering if maps should be used offline or online, there are some similar issues to solve as in common offline scenarios. When considering an offline solution, I usually clarify the following things up-front:

– Amount of data on the device

– Amount of data in memory

– Performance in cold start scenarios

– Need to merge the result of edit operations when online

– May the data be used offline

The amount of used data should be approximately known when choosing a solution, as the device has a limited amount of memory. If the app needs to be able to edit data in offline situations, then we need a design for merging the result of edit operations back to the main storage. Web solutions can perform complex operations fast, but this advantage is reduced, if the connection is slow. Both web and native solution can use caching, although the memory of a device can limit caching possibilities. When using public data, it is possible that there are limitations for offline or limited usage of the data.

Map libraries

Earlier I wrote an article about choosing a map provider. As mentioned also John Papa has a good criteria list for choosing a third party library. When choosing a map library for native usage I use the following criteria:

– Licensing model

– Support for data formats

– Ease of development

– Out of the box functionality

Licensing varies from free availability to limited amounts of data or usage. Used data format depends on source data and how the data will be stored. For developers it is important, that a third party is stable, has a living community and good documentation – especially regarding error messages or examples. Out of the box functionality can save the developer a lot of time for implementing performance critical algorithms and standard map functionality. Many libraries like for example OSMDroid; have extensions like osmbonuspack, which provide more functionality for handling different data formats and adding different layers to the map. Other good providers for Android map libraries are for example Nutiteq, Mapsforge, MapBox, Google and Esri. With Leaflet it is possible to build offline mobile web apps with map.

Development/design issues

When developing native offline apps for Android, I usually solve the following issues first:

– Data format

– Data storage

– Data update strategy

If only a small amount of data is needed, it can be simply available in one file. It is usually not the best idea to load all the data at once into memory. This might be unnecessary and decreases performance, when all the data is not visible at the same time. You have to think about a more scalable solution if:

– The amount of data is unknown or larger than for example 1000 geometries

– Your Klm-file is larger than 10 MB

– You need tiles in different zoom levels

There are different possibilities for optimized storage of the vector and raster data. Vector data can be divided into grid. Raster data can be divided per coordinates and zoom level e.g. into a SqlLite database. Many libraries already support custom tiles for raster data.

One possibility to increase performance is to preprocess the data into a format that is optimized for search. In an online application this is usually done by the map service on server-side, but in offline apps these performance critical operations need to be solved on the end device.

We also need to consider data updates, when data is updated from central storage to the device or when edits from the device are updated back to the central storage. In the simplest case data is copied manually to the SD-card when needed and no automation is used. It is also possible to synchronize data when the app goes online. If edits are made offline, a good merging strategy is needed, especially if geometry shapes are edited in addition to attributes.

When you are considering offline capabilities for native apps, I recommend implementing a worst case scenario as part of an early proof of concept. During the POC it is possible to proof, if the amount of data fits in to the limits of the device and if the core functionality is supported from the evaluated map library. With today’s connectivity and processing capabilities of the new devices, working with offline maps can be efficient.

What experiences do you have with offline apps or mobile maps development? Let me know by writing a comment. I am looking forward to discuss this with you.