

Aufgabe 1 - Mini-Browser

a) WebView

Erstellt eine neue App, die ein Textfeld, einen ok-Button und eine WebView beinhaltet.

b) URL laden

Der Klick auf den Button soll die angegebene URL in die WebView laden, falls sich das Gerät im WLAN befindet (WifiManager!).

c) SharedPreferences

Mithilfe der SharedPreferences soll nun die zuletzt aufgerufene URL gespeichert und beim nächsten App-Start geladen werden. Was ist ein geeigneter Zeitpunkt, um die URL zu speichern?

Aufgabe 2 - Maps-Anbindung

a) Tabs

Macht die Activity nun zu einer SwipeView. Es gibt die grundsätzlichen Möglichkeiten, die Tabs in der ActionBar auswählen zu lassen, oder einen TitleStrip zu verwenden. <http://developer.android.com/training/implementing-navigation/lateral.html>.

b) Maps-Objekt einbinden

Das zweite Tab soll nun eine Map enthalten. Dazu ist etwas Vorbereitung nötig - das MapFragment benötigt einen API-Key von Google. Dieser wird aus dem Paketnamen (von euch im Projekt festgelegt) und dem Key abgeleitet, der zum signieren der App verwendet wird. Eclipse hat bereits einen Debug-Key angelegt, dieser ist dann anzugeben: <https://developers.google.com/maps/documentation/android/start>

(c) Server-Locations anzeigen

Die eingebundene Map soll jetzt die Positionen der Server anzeigen, zu denen man sich im Browser verbunden hat. Die Position lässt man sich z.B. über utrace.de als XML zurückgeben, ihr dürft auch zuerst Dummy-Positionen verwenden.