

Ticketingservice Readme

1 Konfiguration der Ticketing App für ihren Servers

Ihr Server muss über https erreichbar sein. Eine Anleitung zur Konfiguration des Servers und zum Erzeugen eines Zertifikats finden sie unter:

<http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html#Configuration>

Greifen sie auf ihren Server mit einem Browser, z.b. Firefox zu. Nun sollte eine Sicherheitswarnung erscheinen das die Verbindung nicht vertrauenswürdig ist. Mit I Understand the Risks → Add Exception → im Dialog Certificate Status → View → unter Details → Export im Format X.509 Certificate(PEM) und speichern sie das Zertifikat unter

`/Ticketing/sec/servercrt.pem`

Starten sie anschließend das Skript

`/Ticketing/sec/create-bks.sh`

und bestätigen sie beim prompt "Trust this certificate" mit "yes".

1.1 Konfiguration der Communication Klasse

Passen sie in der Klasse

`/Ticketing/src/de/uni/passau/fim/chris/android/ticketing/Communication.java`

in der Methode

`postToServer(HttpClient client,JSONObject data)`

die Zeile

```
String url = protocol +  
"://" + getResources().getString(R.string.BASE_URI_TICKETSERVER);
```

an. Ändern sie dazu den String BASE_URI_TICKETSERVER sodass er die URL zu ihrem Server angibt. Hier können sie auch das Protokoll zwischen http und https wechseln. Die zugehörigen Strings findet man in der Datei

`/Ticketing/res/values/strings.xml`

in den folgenden Zeilen:

```

<!-- Replace with connection details of your application server -->
<string name="PROTOCOL">https</string>
<!-- string name="PROTOCOL">http</string -->
<string name="BASE_URI_LOCALHOST">
localhost:13448/TicketingService/rs/ticket/check
</string>
<string name="BASE_URI_TICKETSERVER">
sahel.fim.uni-passau.de:8443/TicketingService/rs/ticket/check
</string>

```

2 Ticketing App mit Emulator

1. In ihrem ADT IDE müssen die beiden Projekte android-filechooser und Ticketing vorhanden sein. Bei android-filechooser handelt es sich nur um eine Library die von dem Ticketing Projekt benötigt wird und es ist hier nicht weiter von Bedeutung.
2. Öffnen sie den Android Virtual Device Manager unter Window → Android Virtual Device Manager. Erzeugen sie mit New einen neuen Emulator, vergeben sie einen Namen für den Emulator, wählen sie als Device Nexus 4 und als Target Android 4.3 - API Level 18. Außerdem müssen sie unter dem Punkt SD Card Speicher für eine SD Karte (z.b. 200 MiB) freigeben, damit sie später Dateien auf ihrem Emulator speichern können.
3. Starten sie ihren Emulator im Virtual Device Manager mit einem Klick auf Start und schließen sie den Virtual Device Manager anschließend.
4. Laden sie sich die Datei BarcodeScanner4.4.apk von

<http://code.google.com/p/zxing/downloads/detail?name=BarcodeScanner4.4.apk>

Schritte 5 und 6 sind nur mit einem gestartetem Emulator möglich

5. In ihrem ADT Verzeichnis unter `pathToADT/sdk/platform-tools/` führen sie folgenden Befehl aus
`./adbinstall pathToBarcodeScanner4.4.apk/BarcodeScanner4.4.apk`
6. Unter Window → DDMS wählen sie ihren Emulator aus und wählen sie unter dem Tab File Explorer den Ordner sdcard aus. Mit "Push a file onto the device" (das icon ist rechts oben im File Explorer) laden sie die beiden Dateien `/Ticketing/test/qr-active.png` und `/Ticketing/test/qr-inactive.png` auf ihren Emulator
7. Mit einem rechtsklick auf das Ticketing Projekt → Run As → Android Application wird die App auf dem Emulator gestartet. Beim ersten Starten ihrer App, wählen sie beim Dialog Auto Monitor Logcat "Yes" aus.

3 Ticketing App mit Android fähigem Gerät

Ihr Android Gerät muss eine Internet Verbindung haben.

1. Aktivieren sie in den Einstellungen die Entwickleroptionen. Je nach Gerät kann das Vorgehen anders sein. Beispielsweise bei Android 4.2. muss man unter dem Punkt “Über das Tablet” 7 mal auf “Build-Nummer” klicken.
2. Aktivieren sie in den Entwickleroptionen unter Debugging den Punkt USB-Debugging
3. Installieren sie vom Google Play Store die App “SSH Tunnel” von Max Lv
4. Starten sie die SSH Tunnel App. Sie werden darauf hingewiesen dass sie ihr Gerät rooten müssen, dies ist allerdings nicht notwendig. Unter “SSH Tunnel Settings” geben sie unter “Host” “bender.fim.uni-passau.de” und als “Port” 22 and. Unter “Account Information” müssen sie ihre Zugangsdaten angeben (Cip-Kennung). Unter “Port Forwarding” müssen sie “Remote Adress” auf ihren Server setzen, “Remote Port” auf 8443 “Local Port” auf 13448 setzen. Starten sie den SSH Tunnel unter “Service Controller” mit einem Klick auf “Tunnel Switch”
5. Verbinden sie das Gerät per USB Kabel mit ihrem Rechner
6. In Menü von ADT unter Run → Run Configurations... das Ticketing Projekt auswählen (Dazu muss man es zumindest einmal per Emulator gestartet haben oder man muss eine neue Run Configuration definieren) und im Tab Target den Button auf Always prompt to pick device setzen
7. Starten sie die Ticketing App und sie können nun ihr Android Gerät auswählen, auf dem die App gestartet werden soll. Sie können die App auch auf ihrem Android Gerät starten und in ADT wie gewohnt debuggen

4 Aufbau und Anwendung der Ticketing App

Die Ticketing App besteht aus 3 Klassen:

- Communication.java: Stellt die Verbindung zum Ticket Server bereit
- MainActivity.java: Grundlegendes Verhalten der App, Reaktion auf Buttons etc.
- LocalhostVerifier: Bei der Anwendung eines Android fähigen Gerätes benötigt, hier nicht weiter von Belang

Die Ticketing App hat die zwei Funktionen:

- Check Ticket from File: ein Ticket gespeichert in einer Datei wird auf Gültigkeit überprüft

- Check Ticket: ein Ticket wird mit der Kamera gescannt und auf Gültigkeit überprüft

Die Funktion Check Ticket ist eher in Verbindung mit einem Android fähigem Gerät anzuwenden. Mit dem Emulator kann man zwar auch eine Kamera simulieren, dafür wird aber eine Webcam benötigt. Den Aufbau der GUI findet man unter Ticketing/res/layout_activity_main.xml. Ein Button ist in der GUI beispielsweise wie folgt definiert:

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_camera"
    android:onClick="checkTicketFromCamera"
/>
```

Mit `android:text="@string/button_camera"` wird der Text des Buttons konfiguriert, hier in der Datei Ticketing/res/values/strings.xml zu finden. Mit `android:onClick="checkTicketFromCamera"` wird definiert das bei einem Klick auf den Button die Methode `checkTicketFromCamera` in der `MainActivity` Klasse ausgeführt wird.