

# Handbuch zu ERP-Connect

(Ausführlichere Informationen/Beschreibungen: Siehe ERP-Connect.pdf)

## Inhalt

Installation der Anwendung .....	2
Tipps für die Bedienung der Anwendung.....	5
Fehlermeldungen/Benachrichtigungen der Anwendung.....	7
Source-Code .....	8
Öffnen des Projektes. ....	8
Aufbau des Sourcecodes .....	9
Libraries .....	11

# Installation der Anwendung

1. Verbinden des Android-Gerätes mit dem Computer.
2. erp\_connect.apk auf das Android-Gerät kopieren. (Kann natürlich auch durch andere Wege auf dem Gerät kopiert werden!)

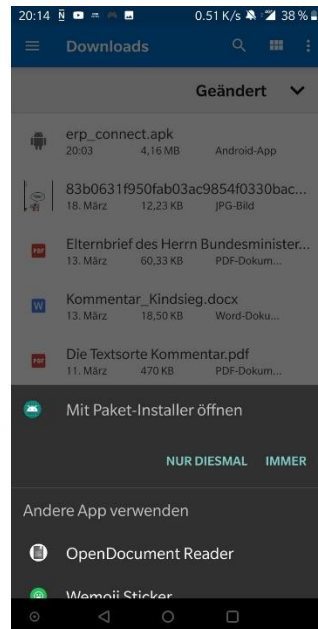


Abbildung 1: Filemanger -> App auswählen



Abbildung 2: Apps aus unbekannter Herkunft erlauben!

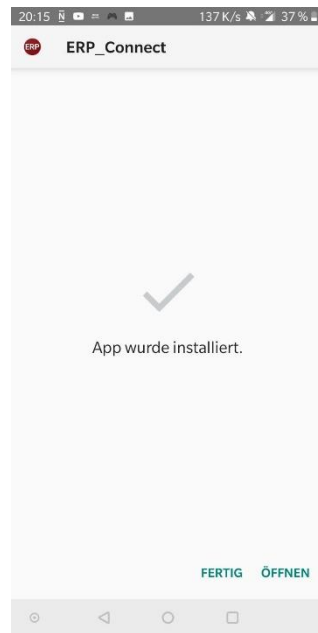


Abbildung 3: Installieren.

Der Schritt (unbekannte Herkunft erlauben) ist notwendig, da diese Anwendung nicht mit den Google Play Services signiert wurde.



# Tipps für die Bedienung der Anwendung

Das Einstellungsmenü schreibt URL-Eingaben automatisch in das https-Format um! Das ist notwendig, da sonst keine Sicherheit gewährleistet werden kann.

Auch ist es nicht möglich ohne Passwort und Benutzername eine Verbindung zu einem Webservice herzustellen.

Der Test-Webservice verwendet ein HTTPS Zertifikat für die Domain Sysco.at, jedoch hat der Webservice eine IP-Adresse als „Domain“ – dies erkennt der Http-Client und würde die Verbindung nicht erlauben. Deswegen wurde eine Ausnahme hinzugefügt, welche prüft ob der Ziel-Host die IP-Adresse des Test-Webservices besitzt – damit wird die Verbindung für gültig erklärt. Für andere Ziel-Webservices funktioniert das nicht so. Bei diesen muss sichergestellt werden, dass die HTTPS-Authentifizierung richtig abläuft.

Manche Funktionen auf den Detail-Ansichten sind nicht ausführbar, dies liegt dann vor, wenn zu wenige Informationen in dem ERP-System gespeichert sind. (Beispiel: Nur Vorwahl und Ort-Nummer -> Keine vollständige Telefonnummer) Dies wird mit einer Meldung kenntlich gemacht.

Die Suche funktioniert primär als Filter. Durch Auswahl der Enter-Taste/Such-Symbol kann jedoch eine absolute Suche durchgeführt werden.

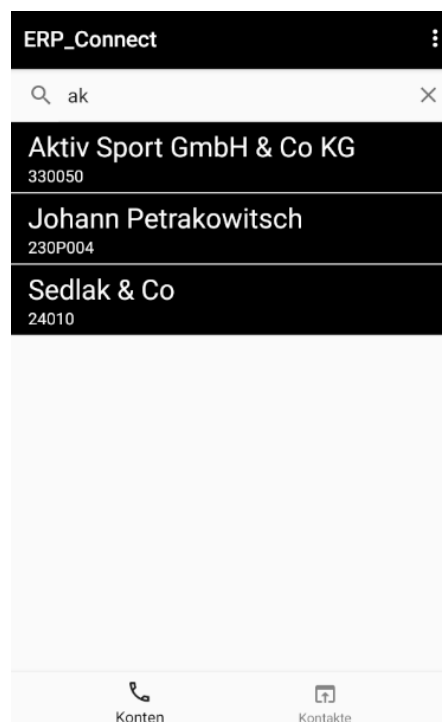


Abbildung 1: Relative Suche

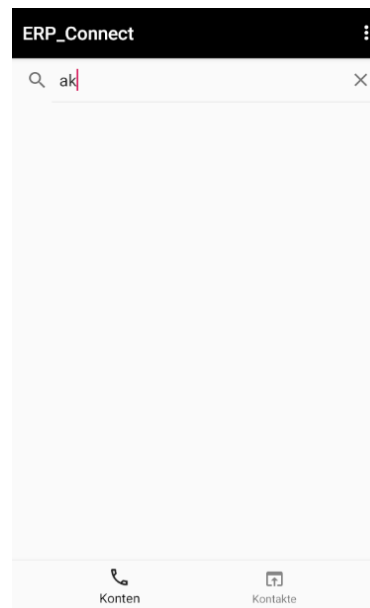


Abbildung 2: Absolute Suche

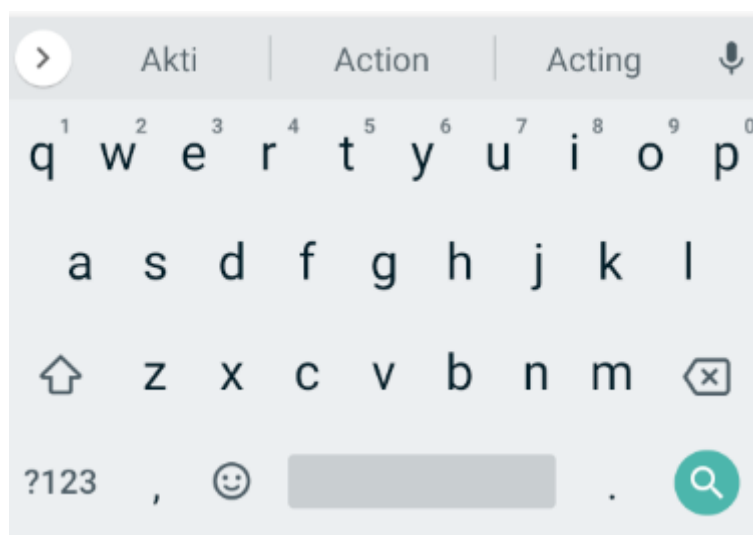


Abbildung 3: Tastatur mit Suchsymbol

# Fehlermeldungen/Benachrichtigungen der Anwendung

Die Anwendung stellt Fehler in einer Snackbar dar. Dafür gibt es verschiedene Fehlercodes/Finishcodes welche darin dargestellt werden.

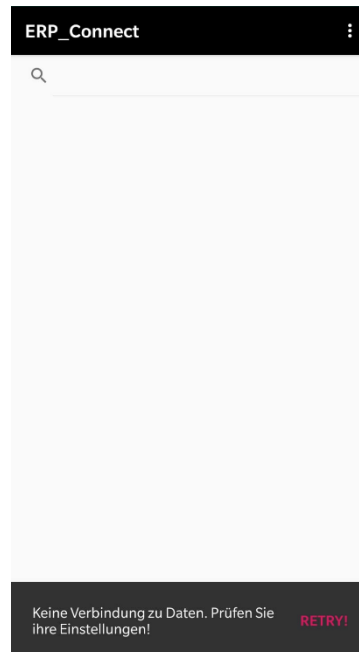


Abbildung 5: Snackbar mit Fehlermeldung

- Leider nicht gespeichert. Bitte erneut starten! -> Wenn das Speichern fehlgeschlagen ist. Bitte schauen ob genug Speicher vorhanden ist.
- Keine Daten vorhanden! Bitte starten Sie das Internet! -> Keine Internetverbindung vorhanden, und keine Offline-Daten vorhanden.
- Die lokal gespeicherten Daten konnten nicht geladen werden. -> Fehler beim Laden der lokalen Daten, bei Auftreten dieses Fehlers werden die gespeicherten Daten gelöscht.
- Keine Verbindung zu Daten. Prüfen Sie ihre Einstellungen! -> Keine Verbindung zu dem Webservice möglich (Falsches Passwort/Benutzer/URL)
- Fehler beim Starten, bitte kehre zurück! -> Tretet auf wenn eine Detail-Ansicht ohne Kontonummer oder Kontakt Nummer gestartet wird.
- Lokale Dateien geladen. Vielleicht nicht aktuell. -> Keine Verbindung möglich, jedoch alte Daten geladen
- Alles gespeichert! (Wenn Konten und Kontakte gespeichert und geladen wurden)
- Konten geladen und gespeichert!
- Kontakte geladen und gespeichert!

# Source-Code

Hier wird beschrieben wie der Source-Code aufgebaut ist und wie dieser mit Android Studio geöffnet werden kann.

## Öffnen des Projektes.

**Achtung!** Best möglichst Android-Studio mit Administratoren-Rechte starten. Falls nach einer Android-SDK gefragt wird (dieses Projekt wurde mit der Android API 29 entwickelt, auch diese verwenden oder nach installieren!).

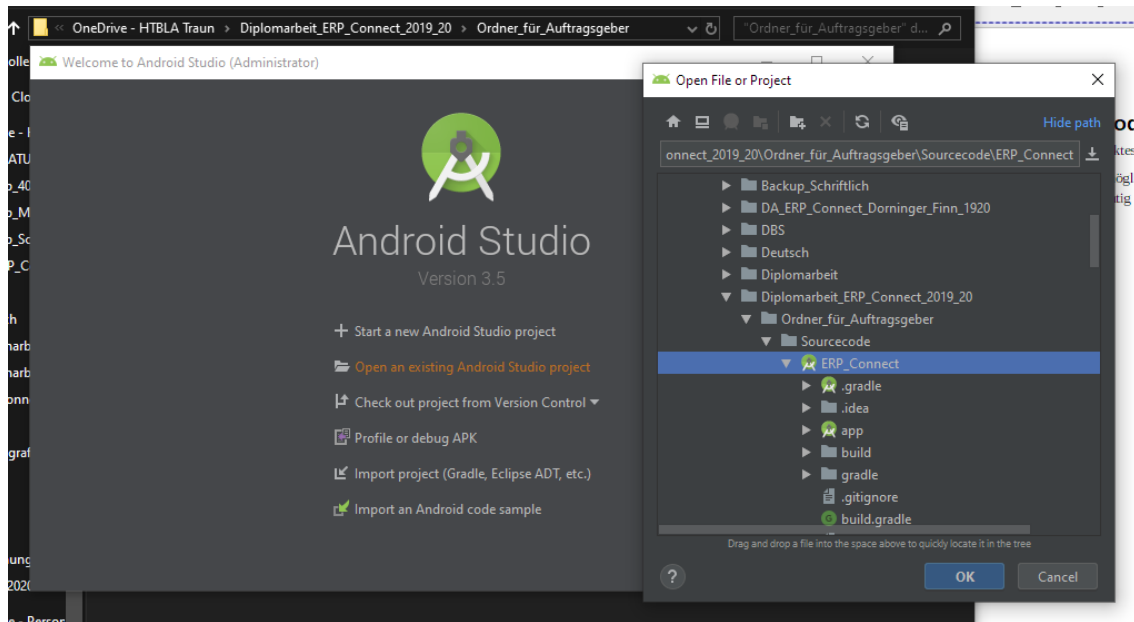


Abbildung 6: Öffnen des Projektes (Ordner ERP\_Connect)



## Aufbau des Sourcecodes

Der Source-Code ist in drei wichtige Packages aufgeteilt. Der Code ist vollständig kommentiert. Bei Unklarheiten kann man sich gerne bei mir melden. Es wurde für den größten Teil der Anwendung das MVP-Architekturmuster verwendet.

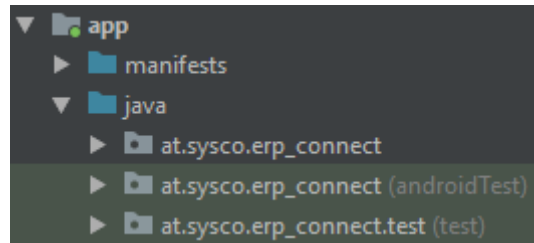


Abbildung 7: Wichtige Source-Code Packages

- `at.sysco.erp_connect (androidTest)` – Testfälle welche ein echtes/emuliertes Gerät benötigen, weil auf Systemressourcen zugegriffen wird.
- `At.sysco.erp_connect (test)` – JUnit Tests welche keine Systemressourcen benötigen.
- `at.sysco.erp_connect` – Source Code der Anwendung, dieser ist weiter unterteilt:

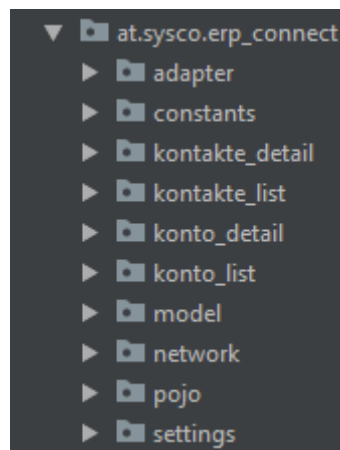


Abbildung 8: Einteilung des Sourcecodes

- `adapter`: Beinhaltet Adapter-Klassen welche für die RecyclerViews notwendig sind.
- `Constants`: Definierte Fehlermeldungen
- `kontakte_detail`: Umfasst Klassen, welche für die Kontakte-Detail Activity zuständig sind. Da mit dem MVP-Architekturmuster gearbeitet wurde (Presenter, View (Activity) und der Contract)
- `kontakte_list`: Gleich wie Kontakte\_Detail nur für eine Listenansicht.
- `konto_detail`: Gleich wie Kontakte\_detail nur für Konten.
- `konto_list`: Gleich wie Konto\_Detail nur für Konto\_list
- `model`: Beinhaltet für jede Ansicht eine Model-Klasse. (Dort werden die zugehörigen Daten aus dem Webservice oder lokal geladen und gespeichert.)

- network: Http-Client-Klasse, welcher für den Webservice-Zugriff benötigt wird. Und WebserviceApi-Klasse in der ein Retrofit-Interface für den Webservice-Zugriff definiert wird. Dieses wird in den Model-Klassen verwendet.
- pojo: Beinhaltet POJOS (auch Entitäten)
- Settings: Klassen für die Verschlüsselung, Einstellungsfenster, Speichern der Einstellungen.

**Ressourcen** bieten Einstellungsmöglichkeiten für Layouts/Menüs etc:

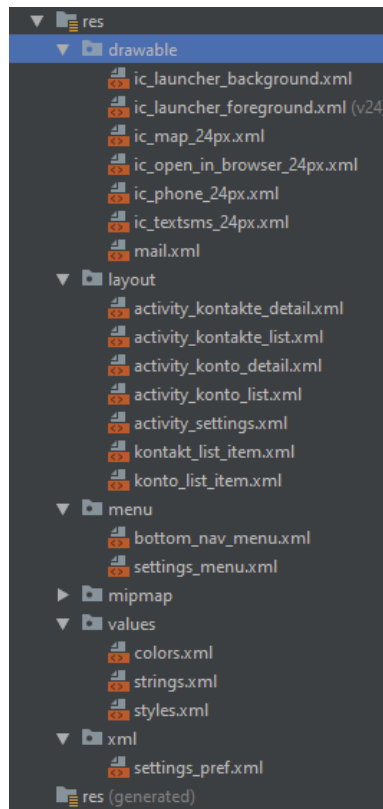


Abbildung 9: Aufbau der Ressourcen

- drawable: Icons welche die Anwendung verwendet.
- layout: Beschreibt die Layouts der verschiedenen Activities.
- menu: Beschreibt Layout der verschiedenen Menüs.
- mipmap: App-Icons
- values: Colors -> Farben, welche in der Anwendung verwendet werden sollen, Strings -> Statische Strings sollen dort definiert werden
- xml: Definierung der Einstellmöglichkeiten im Verbindungseigenschaften-Fenster.

# Libraries

Eine kurze Übersicht über die wichtigsten externen Libraries.

## **Daten:**

- Retrofit (Webservice-Zugriff)
- SimpleXMLConverter (Konvertieren zwischen XML und Kotlin-Objekten, auch für Retrofit benötigt!)
- Androidx.security-crypto (Verschlüsselung von Daten)
- Androidx.preference (Einstellungsmenü)

## **Layout:**

- RecyclerView (effiziente Listenansicht)
- Material-Library (beinhaltet unter anderem die Snackbar welche für die Darstellung von Fehlern verwendet wurde)

## **Testen:**

- Espresso (GUI)
- JUnit
- Mockito-Kotlin (Zum Mocken von Objekten/Klassen)