

Mobile Apps Workshop

Workshop 6



Überblick

- Referenten
- Inhalt des Kurses
- Termine
- Sourcen
- Abschlussarbeit & Kriterien zum Bestehen
- Selbsteinschätzung (JavaScript, Swift)
- Teams
- The Movie DB
- Hausaufgaben

Referenten

Denise Scherzinger - Software Engineer

denise.scherzinger@fhnw.ch



Oliver Gepp - Software Architect

oliver.gepp@fhnw.ch



- **Innovationsdienstleister**
 - Gegründet 1968
 - Hauptsitz in Schlieren (ZH)
- **4 Standbeine**
 - Softwareentwicklung
 - Produktentwicklung
 - Management-Consulting
 - Ventures
- **rund 1070 Mitarbeiter**
- **Umsatz 2018: CHF 171 Millionen**

→ <https://zuehlke-careers.com/>



Inhalt des Workshops

1. Native iOS Entwicklung mit Swift
2. Cross Plattform Entwicklung mit Flutter
3. API & DB
4. Mobile App Design
5. App-Stores, App-Icon, Splashscreen
6. Wrap-Up, FAQ, Empfehlungen
7. Abschlusspräsentation

Termine

| | | | | | | | | | | | | | |
|------------|------------|-----------|-----------|------------|------------|------------|-----------|------------|------------|------------|------------|------------|-----------|
| 17. Feb | 24. Feb | 2. Mrz | 9. Mrz | 16. Mrz | 23. Mrz | 30. Mrz | 6. Apr | 20. Apr | 27. Apr | 11. Mai | 18. Mai | 25. Mai | 8. Jun |
|------------|------------|-----------|-----------|------------|------------|------------|-----------|------------|------------|------------|------------|------------|-----------|

Swift & iOS

Flutter

Abschlusspräsentation

Sourcen

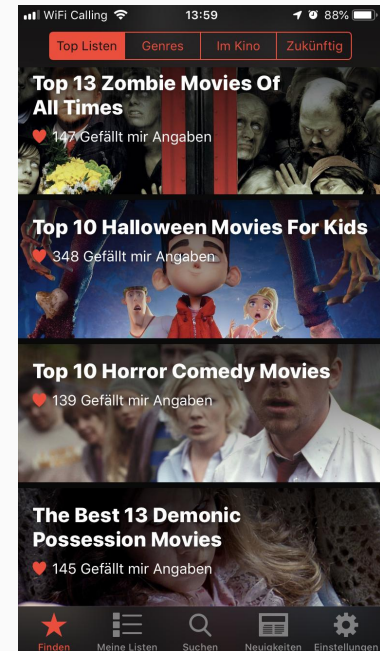
Sämtliche Unterlagen und Code bei github:

<https://github.com/Zuehlke/fhnw-mobile-workshop>



Abschlussarbeit - Thema Movie App

- Anzeige von Informationen über Filme und Serien
- Nutzung der API von <https://www.themoviedb.org>
- Lokales Speichern von Favoriten und Ratings
- Umsetzung eigener Ideen in Gruppenarbeit



Abschlussarbeit - Rahmenbedingungen

- App muss funktionieren und einen Zweck erfüllen
 - Einbindung von <https://www.themoviedb.org> und einer lokalen Datenbank
 - Inklusive App-Icon und Splash-Screen
 - Umsetzung in Flutter oder Swift
- Übergabe ausschliesslich per git url
 - git history vollständig (nicht nur ein commit)
 - E-Mail mit repo-url an Denise Scherzinger und Oliver Gep senden
- Abgabetermin: 04.06.2019 23:59 Uhr (spätestens)
- Abschlusspräsentation im letzten Workshop
- Kein Copy & Paste
- 2er/3er-Gruppe

Umfang der Abschlusspräsentation:

- Zeit: max 5 Minuten pro Gruppe
- Vorstellung der App
- Highlights der App - warum soll man die App installieren?
- Was waren die technischen Herausforderungen?
- Was würdet ihr nächstes mal besser machen?

Testat erreicht wenn:

- App wurde rechtzeitig eingereicht und ist funktionsfähig
- App enthält eigenen Splashscreen und App-Icon
- Abschlusspräsentation der App erfolgreich
- Mindestens eine Top-Liste anzeigen
 - Liste (Bild, Titel,...)
 - Detailansicht (Schauspieler, Handlung, Rating, Poster,..)
- Favorisieren, bewerten (lokal), etc
 - Favoritenliste
 - Offline fähig
- Suche
 - Titel suchen (mindestens Filmtitel)
 - Detailansicht (Schauspieler, Handlung, Rating, Poster,..)
- Eine eigene Idee umgesetzt
 - Kreativität: Etwas das Eure App einzigartig macht

- Swift-App
 - Verwendung von Moya und Realm
 - Keine XCode-Warnings (im eigenen Code)
 - Verwendung von SwiftUI

oder

- Flutter-App
 - Verwendung von http und sqflite Packages
 - State Management mit Provider Pattern

Wie gut sind Eure Kenntnisse in

- Swift / iOS
- Dart / Flutter
- Mobile App Entwicklung

**Jetzt:
Selbsteinschätzung abgeben**

Teams & MacBooks

Wer benötigt ein MacBook?

Bitte bei Janusz Szymanski melden:

janusz.szymanski@fhnw.ch

Wichtig: Es wird macOS Catalina für den iOS-Teil benötigt

Jetzt: Teams bilden

je 2 oder 3 Personen

→ Pair Programming

→ Abschlussarbeit

Team Up!



Order MacBook



Coollest App ever!

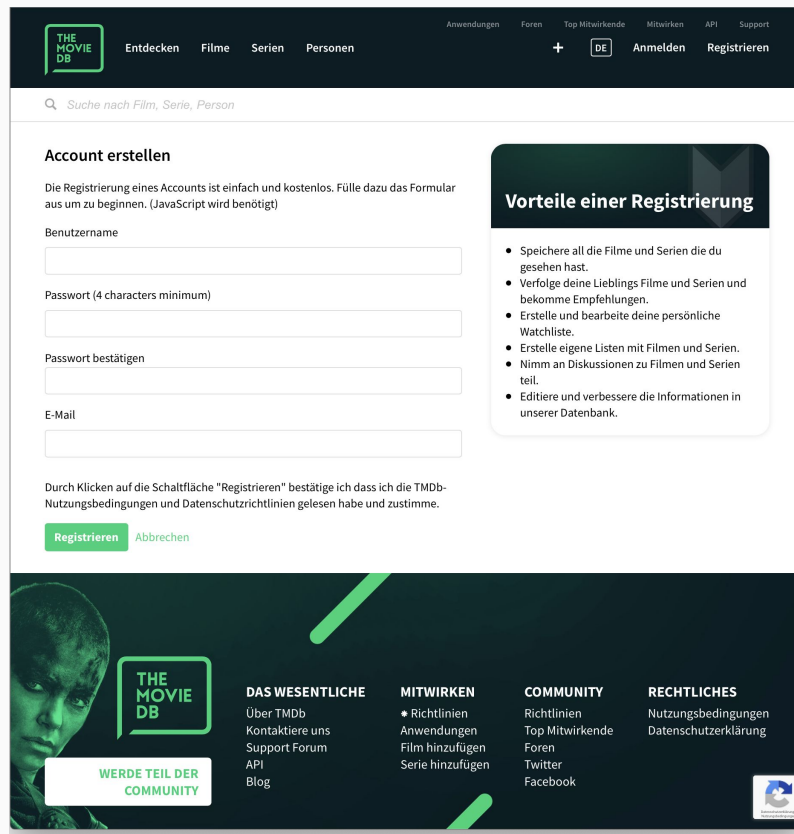


The Movie DB - Jetzt registrieren

→ Eigenen Account erstellen

<https://www.themoviedb.org/account/signup>

<https://developers.themoviedb.org/3/getting-started/introduction>



The screenshot shows the TMDB website's registration page. The header is dark with the TMDB logo and navigation links: Entdecken, Filme, Serien, Personen. On the right, there are links for Anwendungen, Foren, Top Mitwirkende, Mitwirken, API, and Support. A search bar is located below the header.

The main content area is titled "Account erstellen". It includes a sub-header "Die Registrierung eines Accounts ist einfach und kostenlos. Fülle dazu das Formular aus um zu beginnen. (JavaScript wird benötigt)". Below this are four input fields: "Benutzername", "Passwort (4 characters minimum)", "Passwort bestätigen", and "E-Mail". At the bottom of the form are two buttons: "Registrieren" (green) and "Abbrechen" (grey).

To the right of the form is a box titled "Vorteile einer Registrierung" (Benefits of registration). It lists several benefits:

- Speichere all die Filme und Serien die du gesehen hast.
- Verfolge deine Lieblings Filme und Serien und bekomme Empfehlungen.
- Erstelle und bearbeite deine persönliche Watchliste.
- Erstelle eigene Listen mit Filmen und Serien.
- Nimm an Diskussionen zu Filmen und Serien teil.
- Editiere und verbessere die Informationen in unserer Datenbank.

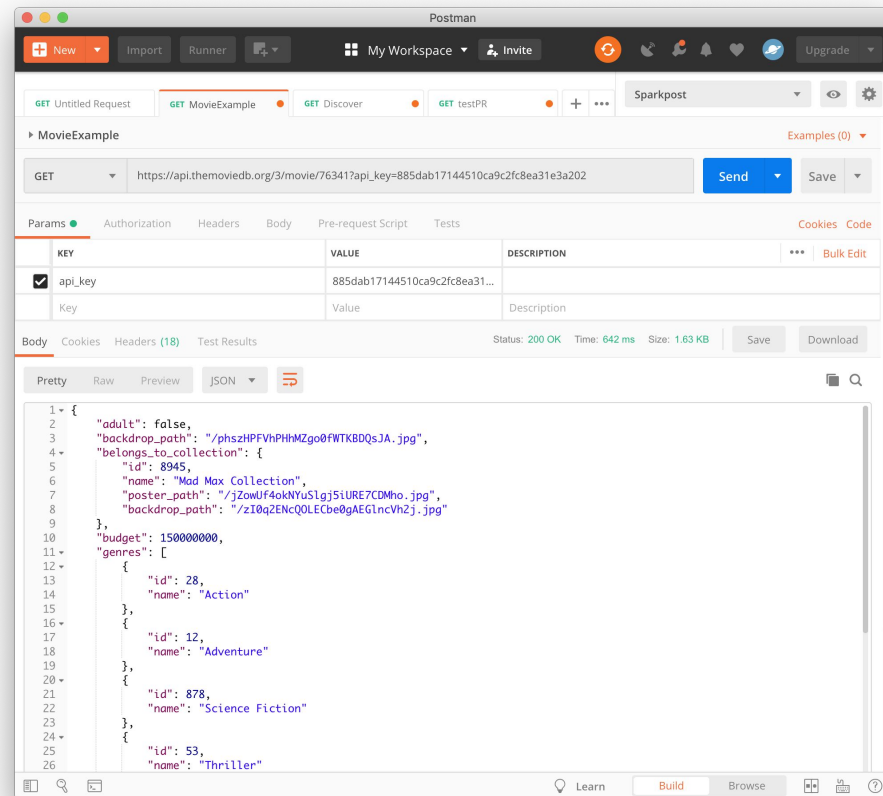
The footer features a large image of a movie character on the left. To its right is the TMDB logo and a button "WERDE TEIL DER COMMUNITY". Further right are four columns of links: "DAS WESENTLICHE" (Über TMDB, Kontaktiere uns, Support Forum, API, Blog), "MITWIRKEN" (Richtlinien, Anwendungen, Film hinzufügen, Serie hinzufügen), "COMMUNITY" (Richtlinien, Top Mitwirkende, Foren, Twitter, Facebook), and "RECHTLICHES" (Nutzungsbedingungen, Datenschutzerklärung). A small Creative Commons license logo is in the bottom right corner.

The Movie DB - API kennenlernen

→ Zum Beispiel mit Postman

- Was sind die beliebtesten Filme?
- Wie erhält man Bewertungen eines Films?
- Wie man Filmposter laden?
- Welche Schauspieler haben in einem Film mitgespielt?
- ...

→ <https://www.themoviedb.org/documentation/api/discover>



Hausaufgaben

Swift - Grundlagen erarbeiten

- Xcode aus dem AppStore installieren
- Die Dateien aus dem [Workshop-Repository](#) laden
- Alle 4 Playgrounds durcharbeiten
- Eigene kleine Algorithmen schreiben um Verständnis zu prüfen

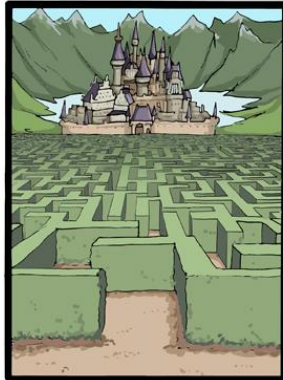
Flutter

- Flutter SDK installieren
(<https://flutter.dev/docs/get-started/install>)
- bevorzugte Entwicklungsumgebung installieren (z.B. Android Studio/IntelliJ, Visual Studio Code), inkl. Flutter und Dart Plugins
(<https://flutter.dev/docs/get-started/editor>)
- Flutter Demo App erstellen und starten, Hot Reload ausprobieren
(<https://flutter.dev/docs/get-started/test-drive>)

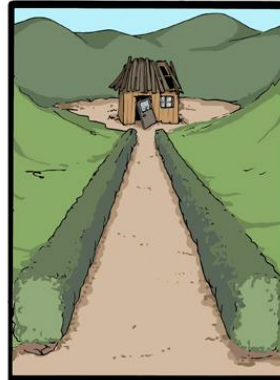
Anhang - Überblick App-Entwicklung

The dilemma of mobile apps development

Develop a native app for each device and maintain several projects



Use a unique framework (Phonegap, Adobe Air, Appcelerator) and maintain only one project

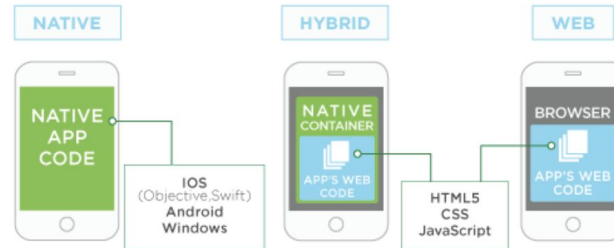


MOBILE APPS AT A GLANCE NATIVE VS. HYBRID VS. WEB APPLICATIONS



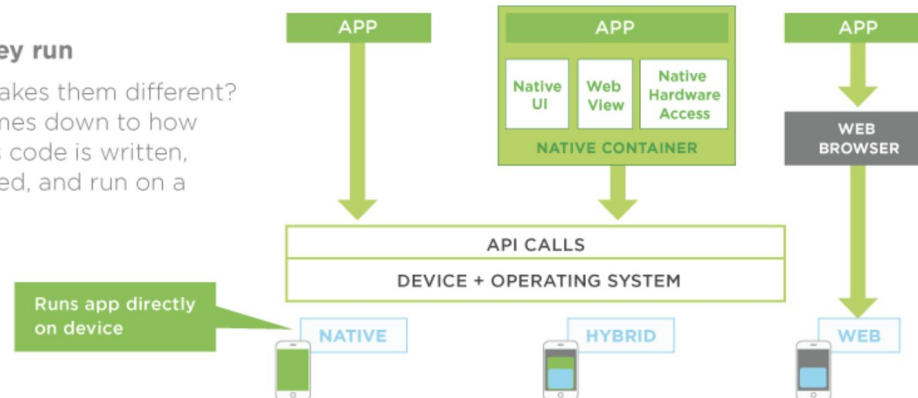
How they're structured

When it comes to building your mobile app, you have three options: native, web, or hybrid.



How they run

What makes them different? It all comes down to how an app's code is written, structured, and run on a device.



Web Apps

- Mobile Version einer Webseite
- Keine Installation - kein Speicherplatzverbrauch
- HTML5, CSS, JavaScript
- Vertrieb ohne AppStore, ohne Beschränkungen
- Zugriff auf Hardware ist limitiert
- Keine Offlinefunktionalität
- Keine Push-Notifications



Progressive Web Apps (PWA)

- Relativ Neues Browser Feature
- Service Workers & Cache -> limitierte offline-Nutzung
- Push APIs erlauben Push Notifications
- Vertrieb ohne AppStore, ohne Beschränkungen
- Nicht alle Web App Probleme werden gelöst
 - z.B. plattformspezifische Navigation



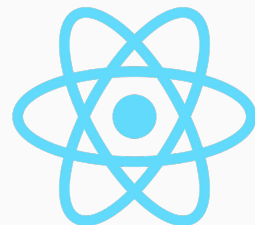
Hybrid Apps

- Kombination aus nativer App & Web
- Nativer App Container mit Webbrowser
- Webtechnologien (HTML5, CSS, JS/TS)
- Geringer Aufwand viele Plattformen abzudecken
 - Testaufwand nicht unterschätzen
 - Alte Android-Geräte stossen schnell an Grenzen
- Zugriff auf Hardware erfolgt über Plugins
 - Kamera, Mikrofon
 - Kalender, Kontakte, Fotos
 - Push Notifications
 - Aber: Anpassung von Plugins ist nicht trivial und setzt native Kenntnisse voraus



Cross Plattform Apps

- Entwicklung ähnlich der Entwicklung von Hybrid Apps
 - Eine gemeinsame Codebasis mit Optimierung für die jeweilige Plattform
 - Jeweils mit [Vor- und Nachteilen](#) verbunden.
- Prominente Vertreter:
 - Xamarin (Microsoft) → C#
 - Flutter (Google) → Dart
 - React Native (Facebook) → JavaScript



Native Apps

- Voller Hardwarezugriff und beste Performance
- Entwicklung für jede Plattform separat
 - iOS: Swift (oder Objective C)
 - Android: Kotlin (oder Java)
 - Ebenso spezifisch für legacy Plattformen
 - z.B. Blackberry, Windows, Symbian, ...
- Vertrieb nur über den jeweiligen App-Store
- Erste Ansätze für Code-Sharing sind im Entstehen
 - Zugriff auf Kotlin-Code von Swift
 - Z.B. zur Teilen der Business-Logik
 - Entwickeln von Android-Apps in Swift oder iOS-Apps in Kotlin