

---

# AdoptAPal

Ein Android Projekt von Anna-Lisa Merkel,  
Erik Bünnig und Molham Al-khodari

---

# Gliederung

- Projektidee
- Projektstand der Services
  - Authentication
  - Animals
  - User
- Technische Probleme
- Weiteres Vorgehen
- Zusammenarbeit

# Projektidee

- Eine Applikation, um Tiere in der Umgebung zu finden und zu adoptieren.



# Projektstand der Services

- JSON in, JSON out
- Authentication funktioniert in Isolation
- Edge Cases gefunden und mit Issues dokumentiert
- Tests automatisiert für schnellen Dev Cycle



# Authentication Service

- Vorhanden
  - Erstellen von Nutzern
  - Erstellen eines Tokens für einen Nutzer
  - Validieren von Tokens
- Noch nicht vorhanden
  - Löschen von Nutzern
  - Bearbeiten von Nutzern
  - Token-Invalidierung bei Passwortänderung
  - Unique constraint für Email
  - Stateless unit tests

# Animal Service

- Vorhanden
  - Erstellen von Tieren
  - Erstellen von Farben (Fell- / Tierfarben)
  - Erstellen von Tier-Kategorien (Tierarten)
  - Endpoint zum Abrufen aller Tiere / Farben / Tier-Kategorien
  - Endpoint zum Abrufen eines Tieres / Farbe / Tier-Kategorie anhand der ID
- Noch nicht Vorhanden
  - Endpoint zum Erstellen eines Tieres
  - Endpoint zum Löschen eines Tieres
  - Abrufen von Tieren anhand der Nutzer-ID
  - Upload / Speichern von Bildern
  - Unittests

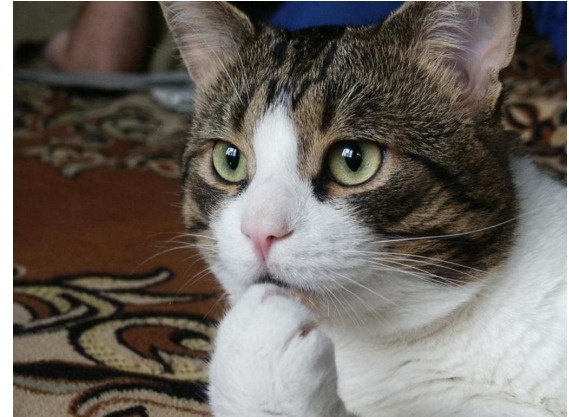
# User Service

- Vorhanden Endpoints
  - Abrufen alle Nutzer und Adressen
  - Abrufung einer Nutzer oder Adressen anhand der ID
  - Erstellung von User und Adressen
  - Updaten von User oder Address anhand der ID
  - Löschen von User oder Address anhand der ID
- UnitTest
  - Address Resource Test
  - User Resource Test
- Noch nicht Vorhanden
  - Middlewares
    - is Authenticated
    - is Owner

▼	UserService
▼	User
	GET getAll
	GET getByID
	GET getByEmail
	PUT UpdateByID
	DEL DeleteByID
	POST CreateUser
▼	Address
	GET getAll
	GET getByID
	POST createAddress
	PUT updateByID
	DEL deleteByID

# Technische Probleme

- JWT werden bei Passwortänderung nicht invalidiert
- JSON Request-Body nicht korrekt erkannt
- Unit tests sind nicht stateless
- Quarkus bei IntelliJ CE
- Unterschiedliche Versionen von Quarkus





# Weiteres Vorgehen

- Meilenstein 1
  - 11. Januar
  - Endpoints fertig
  - Unittests vorhanden
- Meilenstein 2
  - 08. Februar
  - Verknüpfung der Services
  - Unittests komplett
  - Integrationstests vorhanden
  - Infrastruktur
- Meilenstein 3
  - 03. März
  - Einbinden Frontend
  - Test coverage etwa +80%



# Zusammenarbeit

- Regelmäßige Treffen vor der Vorlesung
  - Triage zum Abgleichen der Stände
- Online Kommunikation bei Problemen
  - Wie immer über Discord
  - vorwiegend asynchron
- Gitlab
  - Tracking Issues und Tasks
  - Merge Requests mit Reviewprozess



**Danke für die Aufmerksamkeit**

