

# Dokumentation „Videoflix“

## Allgemeines

### Beschreibung der Anwendung

Videoflix ist eine JavaEE-basierte Webanwendung zur Verwaltung von Filmen und Serien. Nicht angemeldete Benutzer können dabei die Datenbank nach bestimmten Filmen oder Serien durchsuchen oder sich eine Liste aller existierenden Videos ansehen. Zu jedem Video kann auch eine Seite mit weiteren Details betrachtet werden. Über eine einfache Registrierungsseite können sich Benutzer registrieren und sich danach mit ihren Daten anmelden. Angemeldete Benutzer können Filme und Serien anlegen, bearbeiten oder löschen. Zu jedem Video kann optional ein Bild hochgeladen werden. Dieses kann in der Detailansicht des entsprechenden Videos wieder gelöscht werden.

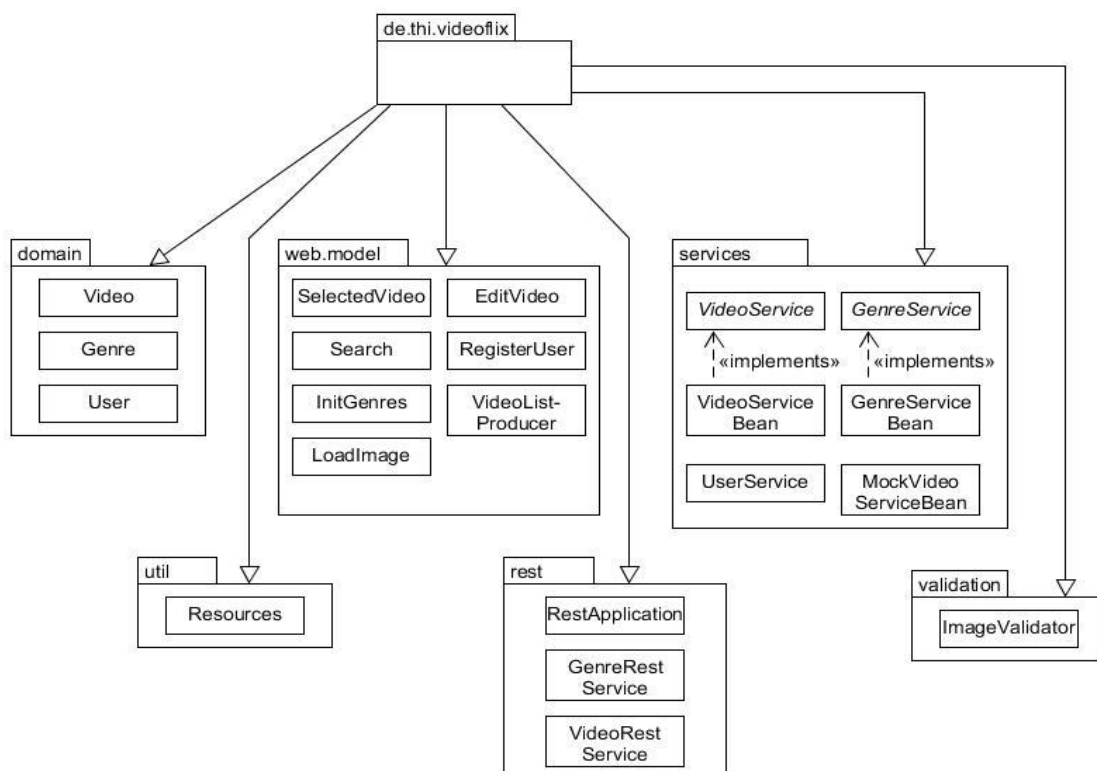
Die Anwendung ist unter [Link einfügen] erreichbar.

### Aufbau der Anwendung

Die Anwendung ist nach dem Model-View-Controller-Prinzip in drei Schichten unterteilt. Die Benutzeroberfläche wurde als webbasierte Views mit JSF (Java Server Faces) umgesetzt. Zur optischen Gestaltung wurde ein Template für alle Seiten eingesetzt. Gesteuert werden die Views mittels CDI-Beans. Durch Services werden Videoobjekte angelegt, bearbeitet und gesucht und Genres zu den Videos hinzugefügt oder aufgelistet. Mittels JPA wurden die Java-Objekte in die Datenbank abgebildet.

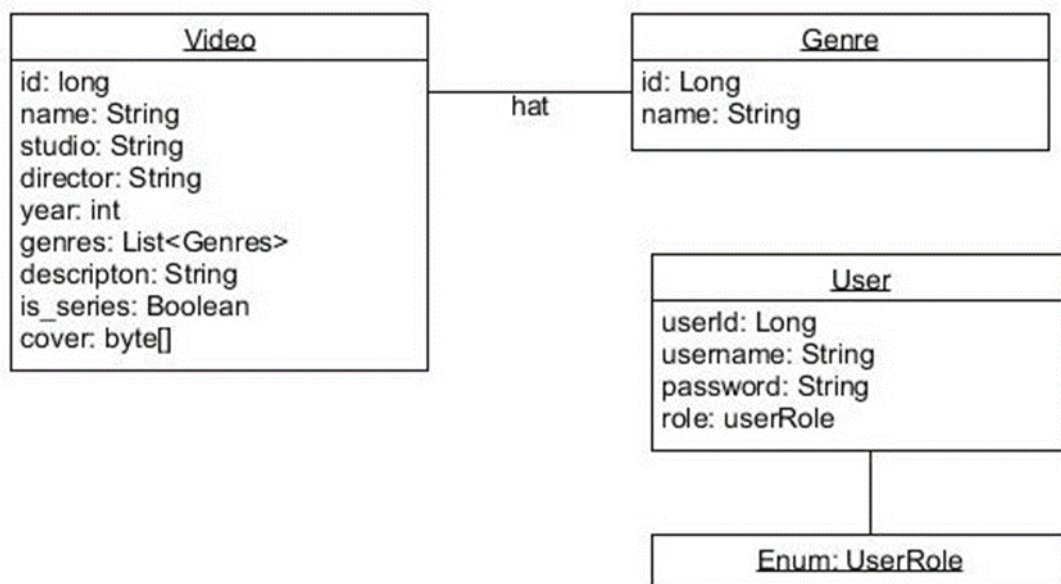
## Komponenten

### Package-Struktur



|            |  |
|------------|--|
| domain     | Entitäten, die im data model verwendet werden. Sie werden durch Services in der Datenbank persistiert. |
| rest       | REST-Webservice, um API der Applikation bereitzustellen.   |
| services   | Zuständige Services für die Datenhaltung, persistieren Domainobjekte.                                  |
| util       | Beinhaltet Hilfsklasse, um Instanzen des Entitymanagers in andere Klassen zu injizieren.               |
| validation | Beinhaltet Validator-Klasse, um hochgeladene Bilder zu überprüfen.                                     |
| web.model  | Enthält die CDI-Beans  |

## Domain



**Video:** Ein Videoobjekt, hat neben Id, Name und weiteren Details außerdem eine Liste von Genreobjekten. Jedem Video können damit ein oder mehrere Genres zugewiesen werden.

**Genre:** Ein Genreobjekt besitzt lediglich eine Id und einen Namen.

**User:** Ein Benutzerobjekt hat eine Id, einen Usernamen, ein Passwort und eine Benutzerrolle. Die Rolle wird als Enum aus zwei Möglichkeiten (USER und ADMIN) ausgewählt.

## UI-Struktur

|                    |  |
|--------------------|--|
| Index.xhtml        | Startseite der Anwendung. Hier steht ein Willkommenstext mit Beschreibung der Seite und Verweis auf die Registrierungsseite.   |
| addvideo.xhtml     | Beinhaltet eine Eingabemaske zum Hinzufügen neuer Videos.  |
| editvideo.xhtml    | In dieser Eingabemaske kann ein bestehendes Video bearbeitet werden.   |
| listVideos.xhtml   | Hier sind alle in der Datenbank bestehenden Filme und Serien mit einem Teil der Details aufgelistet. Neben jedem Video befindet sich der Link zur Detailansicht oder zum Bearbeiten des entsprechenden Videos. |
| videodetails.xhtml | Auf dieser Seite sind alle Details zu einem Video einzusehen, inklusive dem Titelbild des Videos. Dieses kann über einen Button auch gelöscht werden.  |
| search.xhtml       | Der Titel eines Films oder einer Serie kann in ein Suchfeld eingegeben werden und wird in der Datenbank gesucht.   |
| searchResult.xhtml | Die Ergebnisse einer Suche werden hier in einer Tabelle angezeigt. Ein Link neben dem jeweiligen Namen der gefundenen Videos leitet auf die Detailseite weiter.  |
| register.xhtml     | Um sich zu registrieren kann ein Benutzer seinen gewünschten Benutzernamen und ein Passwort eingeben. Bei erfolgreicher Registrierung wird er auf die Login-Seite weitergeleitet.                              |
| login.xhtml        | Bereits registrierte Benutzer können sich mit ihren Daten in der Anwendung anmelden und somit alle Funktionen nutzen.  |
| login-error.xhtml  | Zeigt eine Fehlermeldung an, falls der Login nicht funktioniert hat.   |
| template.xhtml     | Das Template ist auf allen Seiten eingebunden und gibt der Anwendung ein einheitliches Design. Auf der linken Seite aller Seiten wird ein Navigationsmenü angezeigt.   |

## Schnittstelle Rest-API

Die Schnittstelle ist über den Pfad „/api“ zu erreichen. Dies wird in der Klasse „RestApplication“ festgelegt.

Durch die Klasse „VideoRestService“ kann auf Services über die URL-Erweiterung „/videos“ mit POST und GET auf die Datenbank zugegriffen werden. Bei Anhängen der Id eines Videos über „videos/{videoid}“ werden alle Informationen über dieses Video angezeigt.

In „GenreRestService“ ist der Zugriff auf den Service über „/genres“ definiert. Über diese URL werden alle in der Datenbank vorhandenen Genres aufgelistet. Über GET können auch neue Genres hinzugefügt werden.