

Dokumentation First.FM

Inhaltsverzeichnis

1	Einleitung	2
1.1	Allgemeines	2
1.2	Formatierung	2
2	Klassen	2
2.1	User	2
2.2	Band	3
2.3	Bandmember	3
2.4	Song	3
2.5	Album	3
2.6	Album2Genre	3
2.7	Label	4
2.8	Genre	4
2.9	Comment	4
2.10	Comment2User	4
2.11	Comment2Song	4
2.12	Comment2Band	4
2.13	Instrument	4
2.14	Upload	4
2.15	Recommendation	5
2.16	Songrating	5
2.17	Friendslist	5
2.18	User2Played	5
2.19	Artist2Song	5
2.20	User2Band	5
2.21	Band2Song	5
2.22	Song2Genre	6
2.23	Song2Album	6
3	Klassendiagramm	7

1 Einleitung

1.1 Allgemeines

In dieser Dokumentation werden die Besonderheiten des Klassendiagramms, für das communitybasierte Musik-Streamingportal FIRST.FM, erläutert. Bei dem Musik-Streamingportal FIRST.FM, haben Benutzer die Möglichkeit selbstkomponierte Musik zu veröffentlichen, Musik anderer Benutzer oder Bands zu hören sowie selbst einer Band beizutreten (haben Sie wirklich die Möglichkeit einer Band beizutreten?).

Am Ende dieser Dokumentation ist, zur besseren Übersicht, das komplette Klassendiagramm noch einmal in seiner Vollständigkeit zu finden.

1.2 Formatierung

In der Dokumentation sowie im dazugehörigen Diagramm gilt die folgende Formatierung.

Klassennamen

Der Name einer Klasse ist im Regelfall **fett** formatiert.

Attribute

Attribute einer Klasse sind folgendermaßen gekennzeichnet

- Reguläres Attribut
- * Attribut ist Teil des Primärschlüssel
- / Abgeleitetes Attribut
- + Fremdschlüssel

Relationentypen

- Reguläre Relationen zwischen Klassen sind durch eine einfache schwarze Linie gekennzeichnet.
- Sogenannte XOR-Relationen besitzen an der ausgehenden Klasse ein Rautesymbol (♦). Hierbei darf die Klasse, für die gespeicherte Entität, nur mit einer einzigen der Klassen gleichzeitig in Relation stehen.

2 Klassen

2.1 User

Eine Entität der Klasse **User** wird aus Speicherplatzgründen durch einen Surrogatschlüssel (*id) identifiziert. Ein anderer Schlüsselkandidat ist hierbei der -loginname,

dieser ist ebenfalls einmalig. Als weitere Besonderheit ist hierbei das Attribut `-online` zu sehen, welches wahr ist wenn der Benutzer online ist und sonst falsch. Der Onlinestatus lässt sich leider nicht ohne Weiteres aus dem Attribut `-last_seen` ableiten. Ein User ist nach dem Abmelden sofort Offline, allerdings auch sobald die Differenz zwischen `-last_seen` und der Jetztzeit größer als 15 Minuten ist.

2.2 Band

Eine **Band** besteht aus mindestens einem User. Dies mag auf Anhieb nicht sinnvoll erscheinen, allerdings kann ein Benutzer auch Mitglied einer Band sein, ohne dass der Rest der Band bei FIRST.FM einen Account haben muss.

2.3 Bandmember

Die Klasse **Bandmember** soll es den Benutzern einer Band ermöglichen, Mitglieder einer Band hinzuzufügen welche nicht über einen Benutzeraccount bei FIRST.FM verfügen. Um die Zugehörigkeit zu einer bestimmten Band zu definieren verfügen diese Entitäten über den Fremdschlüssel `+band_id`. Desweiteren verfügen diese Entitäten über den Fremdschlüssel `+instrument_id`, welche das von Ihnen in der Band gespielte Instrument definiert.

2.4 Song

Ein einzelner Song verfügt natürlich immer über einen Titel (`-title`) und wurde meist zu einem bestimmten Datum veröffentlicht (`-release_date`). Ausserdem verfügt die Klasse **Song** noch über ein abgeleitetes Attribut `-/status` welches sich aus der Quantität und Qualität von Bewertungen aus der Tabelle **Songrating** ableitet. (Bei einer Bewertung von 4,5 bei über 10000 Bewertungen erhält der Song einen Goldstatus, bei einer Bewertung von 4,6 bei über 100000 Bewertungen den Platinstatus.)

2.5 Album

Hier wird dargestellt, welchen Namen (`-name`) das Album trägt, an welchem Datum (`-date`) es veröffentlicht wurde und welchem Label (`+label_id`) es zugeordnet wird.

2.6 Album2Genre

Relationentabelle zwischen einem Album und einem oder mehreren Genres. Ist notwendig um anzugeben, welchem Genre oder welchen Genres das Album angehört.

2.7 Label

Die Klasse **Label** ist eine Liste von Labels, welche Alben veröffentlicht haben.

2.8 Genre

Die Klasse **Genre** ist eine Liste von Genre denen einzelne Songs oder ganze Alben angehören können. Der -name ist hierbei verpflichtend.

2.9 Comment

Die Klasse **Comment** ist nötig für das Erstellen von Kommentaren zu Songs, Alben, Bands oder Solokünstlern. Hierbei wird sowohl eine Zeit (-timestamp) an der das Kommentar erstellt wurde als auch der Inhalt (-content) des Kommentars und der Name des Benutzers (+user_id) der den Kommentar (*id) erstellt hat, gespeichert.

2.10 Comment2User

Relationentabelle zwischen einem Kommentar und einem Benutzer. Damit ein Benutzer einen anderen Benutzer oder Solokünstler kommentieren kann.

2.11 Comment2Song

Relationentabelle zwischen einem Kommentar und einem Song. Notwendig damit Benutzer Songs kommentieren können.

2.12 Comment2Band

Relationentabelle zwischen einem Kommentar und einer Band. Ist dazu da, dass Benutzer ebenso Kommentare zu Bands abgeben können.

2.13 Instrument

Die Klasse **Instrument** ist Liste von Instrumenten, welche von Benutzern oder Bandmitgliedern gespielt werden.

2.14 Upload

Relationentabelle zwischen einem Benutzer und einem Song. Diese Tabelle ist nötig um zu kennzeichnen welcher Benutzer einen bestimmten Song zu einem bestimmten Datum hochgeladen hat.

2.15 Recommendation

Diese Tabelle soll es einem Benutzer (+*recommender_id) ermöglicht, einem anderen Benutzer (+*recipient_id) einen bestimmten Song (+*song_id) zu empfehlen.

2.16 Songrating

Relationentabelle zwischen einem Benutzer und einem Song. Ist notwendig, damit ein bestimmter Benutzer (+*user_id) zu einem oder zu mehreren bestimmten Songs (+*song_id) auf einer Skala von eins bis fünf Bewertungen (-rating) abgeben kann. Aus der Bewertung eines Songs leitet sich dann das Attribut -/status der Klasse **User** ab.

2.17 Friendslist

Diese Tabelle stellt die Freundschaftsbeziehungen zwischen Benutzern dar. Eine Freundschaft ist immer zweiseitig und somit wird gezeigt, dass ein Benutzer (+*user_id) mit einem anderen Benutzer (+*friend_id) in einem freundschaftlichem Verhältnis steht. Da es sein kann, dass ein Benutzer nicht mit einem anderen befreundet sein möchte, gibt es noch das Attribut -active welches nur dann den Wert wahr besitzt wenn die Freundschaft von dem anderen Benutzer bestätigt wurde.

2.18 User2Played

Relationentabelle zwischen einem Benutzer und einem Song. Es soll dem Benutzer möglich sein herauszufinden wann und damit auch wie oft er einen bestimmten Song schon abgespielt hat.

2.19 Artist2Song

Relationentabelle zwischen einem Benutzer und einem Song, falls der Song eine Solo-Komposition des Benutzers ist.

2.20 User2Band

Die Klasse **User2Band** definiert die Zugehörigkeit eines Benutzers zu einer Band, sowie über den Fremdschlüssel +instrument_id, der das von dem Benutzer in der Band gespielte Instrument darstellt.

2.21 Band2Song

Relationentabelle zwischen einer Band und einem Song, falls der Song eine Komposition einer Band ist.

2.22 Song2Genre

Relationentabelle zwischen mehreren Genres und einem Song. Ermöglicht es einem Song mehrere Genres zuzuordnen.

2.23 Song2Album

Relationentabelle zwischen einem Album und einem Song, falls der Song Teil eines Albums oder mehrerer Alben ist.

