

Jiemamy API クイックスタートガイド



version 0.2

製作著作 © 2009 Jiemamy Project and the Others

概要

Jiemamy APIの基礎的な使用方法を、実例に沿いながら説明します。

目次

まえがき	iii
1. このドキュメントのライセンス	iii
2. その他ライセンス	iii
1. はじめに	4
1.1. 概要	4
1.2. 免責事項	4
1.3. 前提条件	4
2. Jiemamyの基本型	5
2.1. Jiemamyコンテキスト	5
2.2. Jiemamyファクトリ	5
2.3. ルートモデル	6
3. Dialect(SQL方言)	7
4. Jiemamyのモデル構築	8
4.1. 簡単なテーブルを作る	8
5. シリアライズ・デシリアライズ	10
6. インポート・エクスポート	11
7. APIについて	12
8. おわりに	13
8.1. サポートが必要な場合	13
8.2. プログラム・ドキュメントにミスを発見した場合	13

まえがき

このチュートリアルでは、簡単なJiemamyモデルの操作を通じて、Jiemamy APIの基本的な使い方を簡潔に紹介します。このチュートリアルを終了すれば、Jiemamy APIを利用したJiemamyモデル編集に関する基本的な知識が身に付きます。このチュートリアルを完了するのに要する時間は約(TODO)分です。

1. このドキュメントのライセンス

Copyright © 2006-2007 Jiemamy Project and the Others.

Licensed under the Apache License, Version 2.0 (the "License") you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

2. その他ライセンス

このドキュメントは、IPAフォント(バージョン 002.03)を使用しています。

IPAフォントは、一般利用者向けIPAフォント使用許諾契約に基づき、PDFに埋め込まれた形で再配布されています。 <http://ossipedia.ipa.go.jp/ipafont/>

第1章 はじめに

1.1. 概要

Jiemamyは、Webアプリケーションの開発に伴うデータベーススナップショット(スキーマと初期データ等)の管理スイートを提供します。GUIエディタによるER図の管理を通じてDBの状態・履歴を管理し、データベースのリファクタリングなどをサポートします。

1.2. 免責事項

理由の如何を問わず、Jiemamy(以下、「本ソフトウェア」と呼ぶ。)のインストールに起因して 生じた損害または損失に対して、Jiemamy Project及びその関係者(以下、「甲」と呼ぶ。)は一切の責任を負いません。

甲は、本ソフトウェアおよびそのドキュメント類に関して一切の保証を行いません。甲は、本ソフトウェアにバグがないこと、これが正常に動作すること、またはユーザ(以下、「乙」と呼ぶ。)の 期待通りに動作することを保証していません。

甲は、乙がご使用になるソフトウェア・プログラム(本ソフトウェアを含むがこれに限定されない)の 使用または使用不能に起因する、逸失利益、職務の中断、データの損失を含む(但しこれらに限定されない)、特別、派生的、間接的又は類似した損害やその他いかなる損害において、あらかじめ甲がそのような損害の 可能性について知らされていた場合であっても、一切の責任を負いません。



重要項目

ソフトウェアの性格上、JiemamyからDBに対して操作を行うことがあります、最悪の場合、その際にDBのデータを破壊してしまう可能性もあります。Jiemamyの利用に先立っては、必ず データのバックアップをお取りください。

1.3. 前提条件

Jiemamyを初めて使用する場合は、システムに次のソフトウェアをインストールしておく必要があります。下記ソフトウェアのインストールに関しては、Web上の各種リソースを参照して下さい。

- ・ Java 2 Platform Standard Edition (J2SE) 5.0 以降

第2章 Jiemamyの基本型

(TODO)

2.1. Jiemamyコンテキスト

Jiemamyの使用を開始する際には、Jiemamyクラスのインスタンスが必要となります。このインスタンスはJiemamyコンテキストと呼ばれ、Jiemamyのインスタンス空間を表します。Jiemamyモデルは、必ずいずれかのインスタンス空間に属し、異なったインスタンス空間に属しているモデル同士が 関連(親子関係や参照関係)を持つことができません。

Jiemamyコンテキストは、以下のように取得します。

```
Jiemamy jiemamy = Jiemamy.newInstance();  
...
```

例 2.1. Jiemamyコンテキストの取得

このメソッドを呼び出した結果、毎回異なったコンテキストのインスタンスが生成されます。

2.2. Jiemamyファクトリ

Jiemamyのモデルインスタンス等、新しいJiemamy関係のインスタンスは JiemamyFactory から取得します。

Jiemamyファクトリは、コンテキストから以下のように取得します。

```
Jiemamy jiemamy = Jiemamy.newInstance();  
JiemamyFactory factory = jiemamy.getFactory();  
...
```

例 2.2. Jiemamyファクトリの取得

ファクトリは、様々なJiemamyモデルのインスタンスを生成します。テーブルやカラム等はJiemamyFactory#newModel(Class)メソッドを使用して取得します。以下の例では、テーブルを新規に作成しています。

```
TableModel tableModel = factory.newModel(TableModel.class);
```

```
...
```

例 2.3. 新しいモデルインスタンスの取得

Jiemamyのインスタンスは、一部の例外を除いて new 演算子によって取得しません。ほとんどのインスタンスは、このファクトリから取得します。java.lang.Classを鋳型情報として、インスタンスを鋳造するイメージです。

また、データ型だけは特殊で、JiemamyFactory#newDataType(DataTypeMold)メソッドを使用します。DataType型のインスタンスは java.lang.Classだけでは生成にあたっての情報が足りないため、代りにDataTypeMold(データ型の鋳型)を使用します。データ型については、Dialect (SQL方言)に依存する為、DataTypeMoldはDialectから取得する必要があります。この件については後述しますが、簡単にDataTypeの生成例だけを以下に示します。

```
BuiltinDataTypeMold mold = dialect.findDataTypeMold(DataTypeCategory.VARCHAR);
BuiltinDataType dataType = factory.newDataType(mold);
dataType.getAdapter(SizedDataAdapter.class).setSize(36);
...
```

例 2.4. 新しいDataTypeインスタンスの取得

2.3. ルートモデル

JiemamyファクトリからはRootModelを取得する事ができます。RootModelは、Jiemamyモデルの根となるモデルであり、全ての有効なモデルは このモデルから辿ることができます。

```
Jiemamy jiemamy = Jiemamy.newInstance();
JiemamyFactory factory = jiemamy.getFactory();
RootModel rootModel = factory.getRootModel();
...
```

例 2.5. ルートモデルの取得

第3章 Dialect (SQL方言)

(TODO)

第4章 Jiemamyのモデル構築

(TODO)

4.1. 簡単なテーブルを作る

(TODO)

```
// initialize Jiemamy
Jiemamy jiemamy = Jiemamy.newInstance();
JiemamyFactory factory = jiemamy.getFactory();
RootModel rootModel = factory.getRootModel();

// set Dialect to ues and get instance
RootModelUtil.setDialect(rootModel, MySQLDialect.class);
Dialect dialect = rootModel.findDialect();

// create TABLE and set name
TableModel tableModel = factory.newModel(TableModel.class);
tableModel.setName("T_USER");

// create COLUMN and set name
ColumnModel columnId = factory.newModel(ColumnModel.class);
columnId.setName("ID");

// create DataType of INTEGER and set it to column
BuiltinDataTypeMold mold1 = dialect.findDataTypeMold(DataTypeCategory.INTEGER);
BuiltinDataType dataType1 = factory.newDataType(mold1);
columnId.setDataType(dataType1);

// add COLUMN to TABLE
tableModel.getAttributes().add(columnId);

// create COLUMN and set name
ColumnModel columnName = factory.newModel(ColumnModel.class);
columnName.setName("NAME");

// create DataType of VARCHAR(32) and set it to column
BuiltinDataTypeMold mold2 = dialect.findDataTypeMold(DataTypeCategory.VARCHAR);
BuiltinDataType dataType2 = factory.newDataType(mold2);
dataType2.getAdapter(SizedDataAdapter.class).setSize(36);
columnName.setDataType(dataType2);
```



```
// add COLUMN to TABLE
tableModel.getAttributes().add(columnName);

// add TABLE to RootModel
rootModel.getEntities().add(tableModel);
...
```

例 4.1. 簡単なテーブルを含むモデルの構築

このように、Jiemamyモデルは、主に以下のような手順を繰り返して構築していきます。

1. JiemamyFactoryから新しいモデルインスタンスを取得する。
2. モデルのプロパティにsetter等を使って値を設定する。この値はさらにJiemamyFactoryで生成するものかもしれません。
3. 1で生成したモデルを、親となるモデルにsetter等を使って設定する。

第5章 シリアライズ・デシリアライズ

作成したモデルは、シリアライザを使用して、XML形式で保存することができます。このXML形式はJiemamy Model Editorが利用するXML形式と 同一です。

シリアライズ・デシリアライズは、JiemamySerializerを使用して行うことができ、このインスタンスはJiemamy#getSerializer()メソッドで 取得する事ができます。

シリアライズの例を以下に示します。ここではRootModelを第二引数に与えた出力ストリームに出力しています。

```
JiemamySerializer serializer = jiemamy.getSerializer();
serializer.serialize(rootModel, new FileOutputStream("./target/output.jer"));
...
```

例 5.1. シリアライズ

また、シリアライズによって出力したXMLは、下記の例のようにデシリアライズし、RootModelを得ることができます。

```
JiemamySerializer serializer = jiemamy.getSerializer();
RootModel deserialized = serializer.deserialize(new FileInputStream("./target/output.jer"));
...
```

例 5.2. デシリアライズ

第6章 インポート・エクスポート

作成したモデルは、インポータ・エクスポータを使用して、(TODO)

第7章 APIについて

以上の通り、Jiemamyはモデルの操作APIを公開し、作ったモデルを自由に扱うことができます。

しかし、リリースして間もないこともあり、未成熟であるのも事実です。今後のバージョンアップにより、バグ修正やAPIの修正・拡張を行っていく ことになると思いますが、そのポリシーを説明します。

Jiemamyの仕様コンポーネント(spec)のバージョンは、"major.minor[-SNAPSHOT]"形式です。majorは大なアーキテクチャの変更があった際に繰り上げ、minorは、仕様に变更(つまり、specに対するあらゆる変更)があった際に繰り上げます。APIの追加は随時行っていきます。そして、APIの廃止は @Deprecated アノテーションで対処し、時期を見て実際に削除します。@Deprecatedから実際の削除までの期間は、v1.0以前は2~3ヶ月、v1.0以降は未定です。(もしかしたらメジャーバージョンアップまで消さない、とするかもしれません。) また、API変更について、メソッドのシグネチャが変わる場合は、旧メソッドを @Deprecated とし、新メソッドをオーバーロードする形にします。シグネチャの変更を伴わない変更(挙動の変更=Javadocの変更)は、v1.0以前は随時行う。v1.0以降はどうするか検討中です。しかし、バージョンアップに伴い、既存のクライアントに影響を出さない事を重要事項として扱っていきます。

実装コンポーネントのバージョンingは、"major.minor.release[-SNAPSHOT]"形式です。major, minorは、準拠する対象仕様バージョンに合わせ(spec v0.2準拠の場合は、v0.2.x)、仕様に変更がなく実装のみを修正(bug fix等)した場合に、releaseを繰り上げます。

次に、Jiemamyのバグの基準を示します。

バグには「仕様バグ」(Javadoc自体の記述漏れ・不整合など)と「実装バグ」(コードの不整合など)がありえます。前者はspecの修正が必要となり、後者は実装の修正を行います。(その時のバージョン番号ポリシーは上記の通り)

Jiemamyの開発は、以下のルールを適用しています。

- Javadocの @throws は RuntimeException についても記述されていなければならない。つまり、メソッド呼び出しの結果、Javadocに記述のない例外(RuntimeExceptionを含む)が飛んだら、バグである。(これは「飛ぶべきではない例外が飛ぶ」という実装バグの可能性と、「飛ぶべき例外がJavadocに記述されていない」という仕様バグの可能性があります。)
- APIの挙動については全てJavadocに記述されているべきで、Javadocを読めば何が起きるか全て把握できるべき。「こういうケースではどうなるのだろう?」という疑問が発生する時点で、Javadocの記述漏れとみなす。(ただし、reflectionで強制的にフィールドを書き換えたりした 場合はどうなるのだろう、等はナシで。) 呼び出しの結果、Javadocに記述されていない動作をするのはバグである。(これは「すべきではない動作をする」という実装バグの可能性と、「すべきである動作がJavadocに記述されていない」という仕様バグの可能性があります。)
- JiemamyのAPIは NullPointerException をスローしない。NullPointerExceptionが飛んだらバグである。

以上のようなケースに遭遇(まだまだ多いと思います)した際には、最終章に記述してあるメーリングリストにご一報下さい。Jiemamyの品質向上にご協力お願いいたします。

第8章 おわりに

8.1. サポートが必要な場合

配布ドキュメントやWebサイト上で解決できない問題に遭遇した場合は、以下のメーリングリストに質問を投稿することができます。

Jiemamy-usersメーリングリスト: jiemamy-users@lists.sourceforge.jp

質問を投稿する際は、以下のページを参考に投稿内容を検討してください。早期解決に繋がります。

技術系メーリングリストで質問するときのパターン・ランゲージ -- 「問題の解決」から「情報の共有」に至るために
<http://www.hyuki.com/writing/techask.html>

8.2. プログラム・ドキュメントにミスを発見した場合

プログラム・ドキュメント等にミスを発見した際も、上記メーリングリストに投稿してください。