

model API

MaachangComet には、model 処理系(DAO)の API が存在します。

MaachangComet の Dao には、少し癖のあるルールが存在します。それは、下記のように、Dao を使った場合などです。

```
TestTable.find( “ where $id=? ” ,id ) ;
```

どのような癖かということ、参照条件のカラム名の前に \$ か : を設定する必要があることです。これは、Javascript で渡されたパラメータを JDBC との型変換を時に、カラム名が必要であるためです。

これらの内容を次ページより、説明していきます。

データベースコミット

```
function commit()
```

説明 : 現在のデータベース更新内容をコミットします.ただし、conf/record.conf の autoCommit=false である必要があります.

データベースロールバック

```
function rollback()
```

説明 : 現在のデータベース更新内容をロールバックします.ただし、conf/record.conf の autoCommit=false である必要があります.

SQL 実行

```
function sql(sql)
```

sql : 実行対象の SQL 文を設定します.

戻り値 : 実行 SQL が[select]系の場合は、json 形式、それ以外なら、文字列が返されます.

説明 : 平文の SQL を実行したい場合に利用します。Model オブジェクトにあるメソッドは、そちらを利用してください。

空のテーブルカラムを取得.

```
function Model::bean()
```

戻り値 : 空のテーブルカラムが返されます.

説明 : 基本的にこれを使って、情報を追加し、save メソッドで保存してください.

テーブルカラムを保存

```
function Model::save(table)
```

table : 保存対象のテーブルカラムを設定します.

戻り値 : [true]の場合、保存成功です.

説明 : ID が存在しない場合は、Insert され、存在する場合は、Update されます.

指定条件でテーブル更新

```
function Model::update(where,params)
```

where : 更新条件を設定します.

params : 更新パラメータを設定します.1 件以上の場合、array で設定します.

戻り値 : [true]の場合、更新成功です.

テーブルカラムを削除

```
function Model::remove(table,join)
```

table : 削除対象のテーブルカラムを設定します.

戻り値 : [true]の場合、削除成功です.

テーブルカラムを結合テーブルを含めて削除

```
function Model::removeJoin(table)
```

table : 削除対象のテーブルカラムを設定します.

戻り値 : [true]の場合、削除成功です.

説明 : このテーブルに[src_join]が定義されている場合、その条件の関連も含めてデータ削除します.

指定条件でテーブル削除

```
function Model::removeFind(where,params)
```

where : 削除条件を設定します.

params : 削除パラメータを設定します.1 件以上の場合、array で設定します.

戻り値 : [true]の場合、削除成功です.

指定条件でテーブルを結合テーブルを含めて削除

```
function Model::removeFindJoin(where,params)
```

where : 削除条件を設定します.

params : 削除パラメータを設定します.1 件以上の場合、array で設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

説明 : このテーブルに[src_join]が定義されている場合、その条件の関連も含めてデータ削除します.

指定条件でテーブル取得

`function Model::find(where,params,sort)`

`where` : 取得条件を設定します.

`params` : 取得パラメータを設定します.1 件以上の場合、array で設定します.

`sort` : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

指定条件でテーブルを結合テーブルを含めて1 対1 条件で取得

`function Model::findJoin(where,params,sort)`

`where` : 取得条件を設定します.

`params` : 取得パラメータを設定します.1 件以上の場合、array で設定します.

`sort` : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

説明 : このテーブルに[join]が定義されている場合、その条件の関連も含めてデータ取得します.

指定条件でテーブルを結合テーブルを含めて1 対他条件で取得

`function Model::findMany(where,params,sort)`

`where` : 取得条件を設定します.

`params` : 取得パラメータを設定します.1 件以上の場合、array で設定します.

`sort` : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

説明 : このテーブルに[src_join]が定義されている場合、その条件の関連も含めてデータ取得します.

指定条件でテーブル取得

`function Model::limit(offset,limit,where,params,sort)`

`offset` : オフセット値を設定します.

`limit` : リミット値を設定します.

`where` : 取得条件を設定します.

`params` : 取得パラメータを設定します.1 件以上の場合、array で設定します.

`sort` : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

指定条件でテーブルを結合テーブルを含めて1 対1 条件で取得

`function Model::findJoin(offset,limit,where,params,sort)`

`offset` : オフセット値を設定します.

`limit` : リミット値を設定します.

`where` : 取得条件を設定します.

`params` : 取得パラメータを設定します.1 件以上の場合、array で設定します.

`sort` : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

説明 : このテーブルに[join]が定義されている場合、その条件の関連も含めてデータ取得します.

指定条件でテーブルを結合テーブルを含めて1 対他条件で取得

`function Model::findMany(offset,limit,where,params,sort)`

`offset` : オフセット値を設定します.

`limit` : リミット値を設定します.

`where` : 取得条件を設定します.

`params` : 取得パラメータを設定します.1 件以上の場合、array で設定します.

`sort` : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

説明 : このテーブルに[src_join]が定義されている場合、その条件の関連も含めてデータ取得します.

指定条件でテーブルの先頭の 1 件を取得

```
function Model::first(where,params,sort)
```

where : 取得条件を設定します.

params : 取得パラメータを設定します.1 件以上の場合、array で設定します.

sort : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

指定条件でテーブルの先頭の 1 件を結合テーブルを含めて 1 対 1 条件で取得

```
function Model::firstJoin(where,params,sort)
```

where : 取得条件を設定します.

params : 取得パラメータを設定します.1 件以上の場合、array で設定します.

sort : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

説明 : このテーブルに[join]が定義されている場合、その条件の関連も含めてデータ取得します.

指定条件でテーブルの先頭の 1 件を結合テーブルを含めて 1 対他条件で取得

```
function Model::firstMany(where,params,sort)
```

where : 取得条件を設定します.

params : 取得パラメータを設定します.1 件以上の場合、array で設定します.

sort : 並べ替え条件を設定します.

戻り値 : 取得結果の内容が JSON 形式で返されます.

説明 : このテーブルに[src_join]が定義されている場合、その条件の関連も含めてデータ取得します.

指定条件でテーブルのカウンタを取得

```
function Model::count(where,params)
```

where : 取得条件を設定します.

params : 取得パラメータを設定します.1 件以上の場合、array で設定します.

戻り値 : 取得結果の件数が返されます.

<MissingMethod>

MaachangComet の Model には、MissingMethod が対応しています。MissingMethod とは、単純に言うと、存在しないメソッドを実行できるものですが、ここでは、SQL 文を極力書かず、メソッド名で代用して、Dao メソッドを呼び出すような感じのものです。

たとえば、以下のような感じで、

```
TestTable.find_by_id_and_name( [id,name] ) =>これを Dao に直すと=> TastTable.find( “ $id=? And $name=? ”,[id,name] )
```

のような感じで利用できます。

MaachangComet での Dao の内、MissingMethod に対応しているメソッドは、以下の通りです。

- removeFind,removeFindJoin
- find,findJoin,findMany
- limit,limitJoin,limitMany
- first,firstJoin,firstMany
- count