

# Dolteng Scaffoldに対する機能追加 とマスタ-ディテールScaffoldの紹介

せいいち (takao)

2009/03/07

# 目次

- Dolteng Scaffoldに対する機能追加
  - Scaffoldに関して
  - Ruby on RailsのScaffold
  - RoR ScaffoldとDolteng Scaffoldの比較
  - Scaffoldのデモ
  - Scaffoldが生成する検索条件
  - 検索ロジック(S2Dao)
  - 検索ロジック(Kuina-Dao)
  - 検索ロジック(S2JDBC)
- マスタ-ディテールScaffoldの紹介
  - マスタ-ディテールScaffoldに関して
  - マスタ-ディテール形式に関して
  - マスタ-ディテールScaffoldのデモ
- おわりに

# Dolteng Scaffoldに対する機能追加

# Scaffoldに関して

- 機能
  - 1つのテーブルに対してCRUDが行える簡易なアプリケーションを生成する
- 有名なもの
  - Ruby on RailsのScaffold
  - CakePHPのScaffold
- 目的
  - 開発生産性の向上
    - 出発点をゴールに近付ける
    - 初心者に対してフレームワークの敷居を下げる
  - 競合するフレームワークへのアピール
    - デモによるアピール度が高い
    - 開発者層以外の人でも、その効果を理解できる

# Ruby on RailsのScaffold

- 実行方法
  - `ruby script/generate scaffold [モデル名]`
  - `ruby script/generate scaffold [モデル名] [[カラム名]:[型名]...]`
- 生成されるアプリケーション

Listing MstMenus							
Id	MenuName	MenuUrl	UpperLevelId	UpdatedAt			
1	ユーザ管理系	/	0	2008/01/13 23:16:06	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
2	ユーザマスタメンテナンス	/mst_users/list	1	2008/04/28 23:51:28	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
3	グループマスタメンテナンス	/mst_groups/list	1	2008/05/01 01:53:23	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
4	メニューマスタメンテナンス	/mst_menus/list	1	2008/04/04 20:19:17	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
5	ユーザとグループの関連メンテナンス	/rlt_users_groups/	1	2008/01/13 02:07:15	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
6	グループとメニューの関連メンテナンス	/rlt_groups_menus/	1	2008/01/13 02:07:04	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
7	ログイン履歴	/	0	2008/03/16 18:30:32	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
8	ログイン履歴(全ユーザ)	/login_irekis/list	7	2008/03/22 01:44:52	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
9	ログイン履歴(お客様御自身)	/login_ireki_self/	7	2008/03/22 15:18:09	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>
10	ログイン履歴(グラフ)	/login_ireki_graph/	7	2008/03/22 15:17:44	<a href="#">Show</a>	<a href="#">Edit</a>	<a href="#">Destroy</a>

« Previous 1 [2](#) [3](#) [4](#) Next »

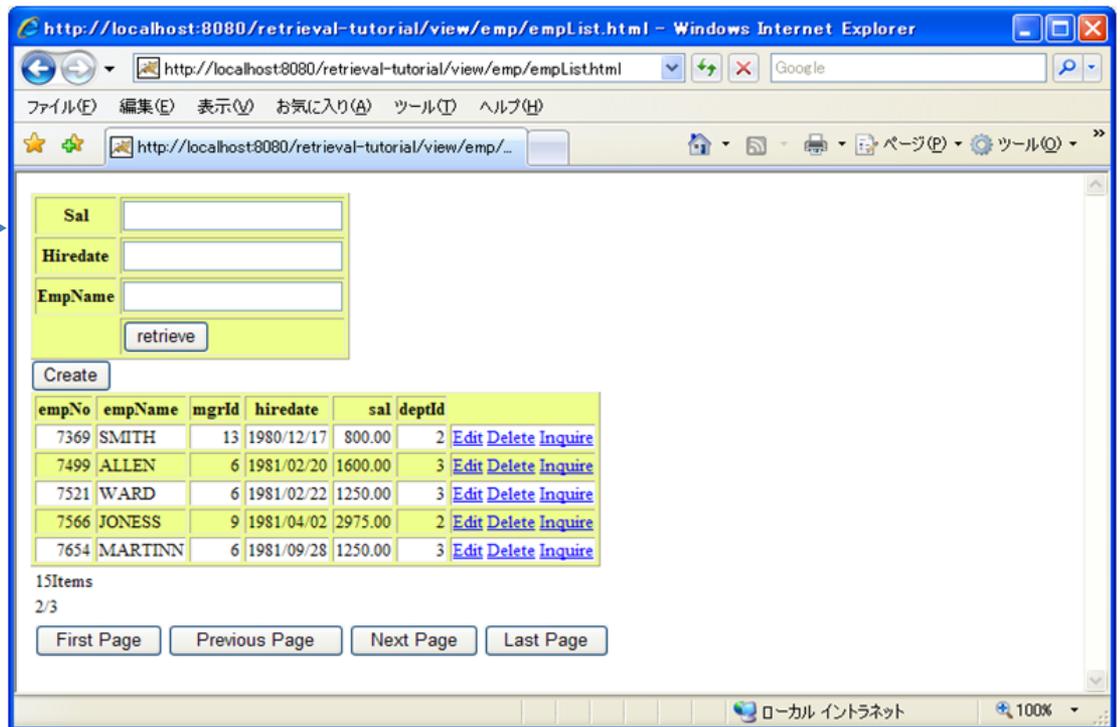
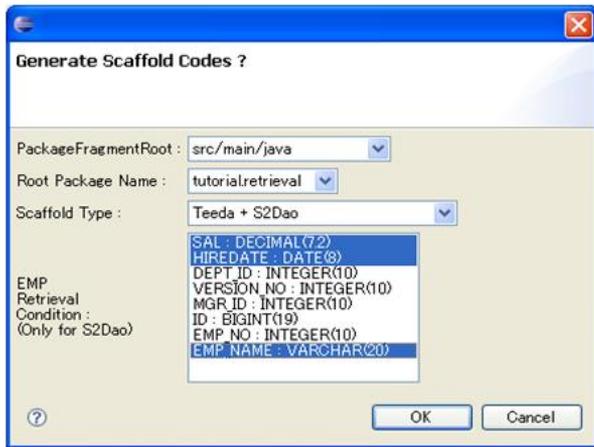
[New MstMenu](#)

# RoR ScaffoldとDolteng Scaffoldの比較

- Ruby on RailsのScaffold
  - ページ移動が可能
- Dolteng 0.34.0以前のScaffold
  - ページ移動ができない
- Dolteng 0.35.0以降のScaffold
  - ページ移動が可能
  - 検索条件の生成が可能

# Scaffoldのデモ

- S2Dao
- SAstruts
- Kuina-Dao
- SAstruts + Mayaa



# 目次

- Dolteng Scaffoldに対する機能追加
  - Scaffoldに関して
  - Ruby on RailsのScaffold
  - RoR ScaffoldとDolteng Scaffoldの比較
  - Scaffoldのデモ
  - Scaffoldが生成する検索条件
  - 検索ロジック(S2Dao)
  - 検索ロジック(Kuina-Dao)
  - 検索ロジック(S2JDBC)
- マスタ-ディテールScaffoldの紹介
  - マスタ-ディテールScaffoldに関して
  - マスタ-ディテール形式に関して
  - マスタ-ディテールScaffoldのデモ
- おわりに

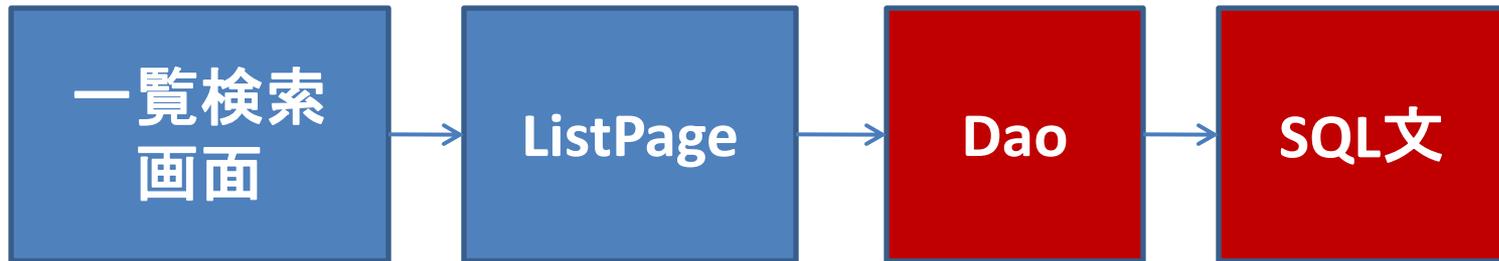
# Scaffoldが生成する検索条件

	テーブルのカラムタイプ	生成される条件
S2Dao版	String	LIKE(前方一致)
	Integer BigDecimal	$\geq$ デフォルト値:「0」
	Date Timestamp	$\geq$ デフォルト値:「1900/1/1」
Kuina-Dao版	String	LIKE(前方一致)
	Integer BigDecimal	$\geq$
	Date Timestamp	$\geq$

# Scaffoldが生成する検索条件

	テーブルのカラムタイプ	生成される条件
SAStruts版	String	LIKE (前方一致)
	Integer BigDecimal	$\geq$ デフォルト値:「0」
	Date Timestamp	$\geq$ デフォルト値:「1900/1/1」
SAStruts+ Mayaa版	String	LIKE (前方一致)
	Integer BigDecimal	$\geq$ デフォルト値:「0」
	Date Timestamp	$\geq$ デフォルト値:「1900/1/1」

# 検索ロジック (S2Dao)



- 生成指示画面にて選択した検索条件に従って、検索ロジックを生成
  - 検索パラメータをAndで繋げたメソッドを作成
  - 検索パラメータをAndで繋げたSELECT文の条件を作成

# 検索ロジックコード例 (S2Dao)

```
@Arguments({"argSal","argHiredate","argEmpNo","argEmpName"})  
public Emp[] findBySalAndHiredateAndEmpNoAndEmpNamePagerCondition(  
    java.math.BigDecimal argSal, java.util.Date argHiredate,  
    java.lang.Integer argEmpNo, java.lang.String argEmpName,  
    EmpPagerCondition dto);
```

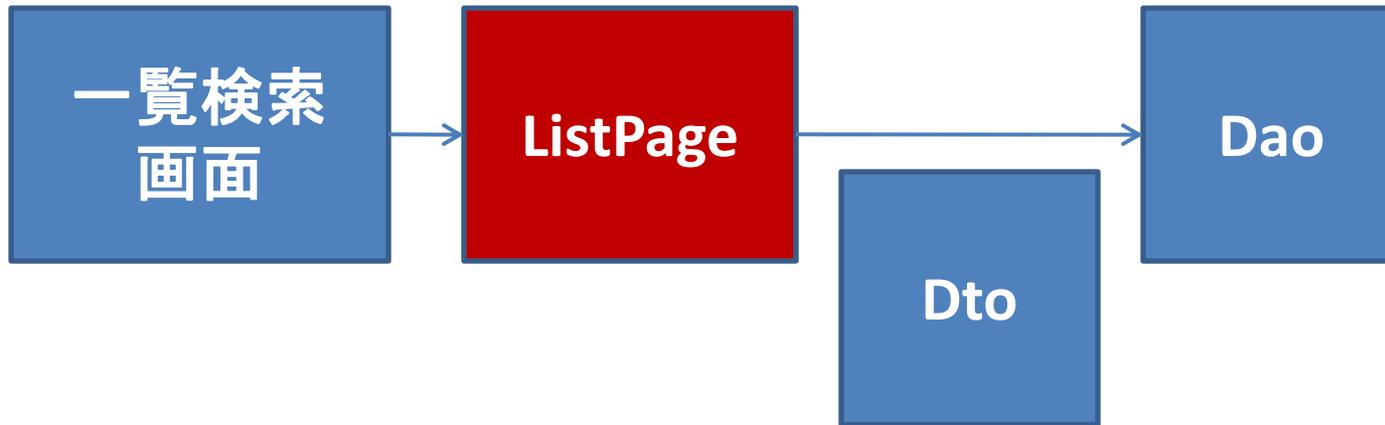
# 検索ロジックコード例 (S2Dao)

```
select * from EMP
/*BEGIN*/
where
  /*IF argSal != null*/
    SAL >= /*argSal*/'0'
  /*END*/
...
/*IF argEmpName != null*/
  /*IF argSal != null || argHiredate != null || argEmpNo != null*/
  and
  /*END*/
  EMP_NAME LIKE concat(/*argEmpName*/' ','%')
/*END*/
/*END*/
ORDER BY SAL,HIREDATE,EMP_NO,EMP_NAME
```

# 検索ロジックコード例 (S2Dao)

```
select * from EMP
/*BEGIN*/
where
  /*IF argSal != null*/
    SAL >= /*argSal*/'0'
  /*END*/
...
  /*IF argEmpName != null*/
    and EMP_NAME LIKE concat(/*argEmpName*/' ','%')
  /*END*/
/*END*/
ORDER BY SAL,HIREDATE,EMP_NO,EMP_NAME
```

# 検索ロジック (Kuina-Dao)



- 生成指示画面にて選択した検索条件に従って、検索ロジックを生成
  - Dtoクラスのプロパティにて、検索パラメータを作成

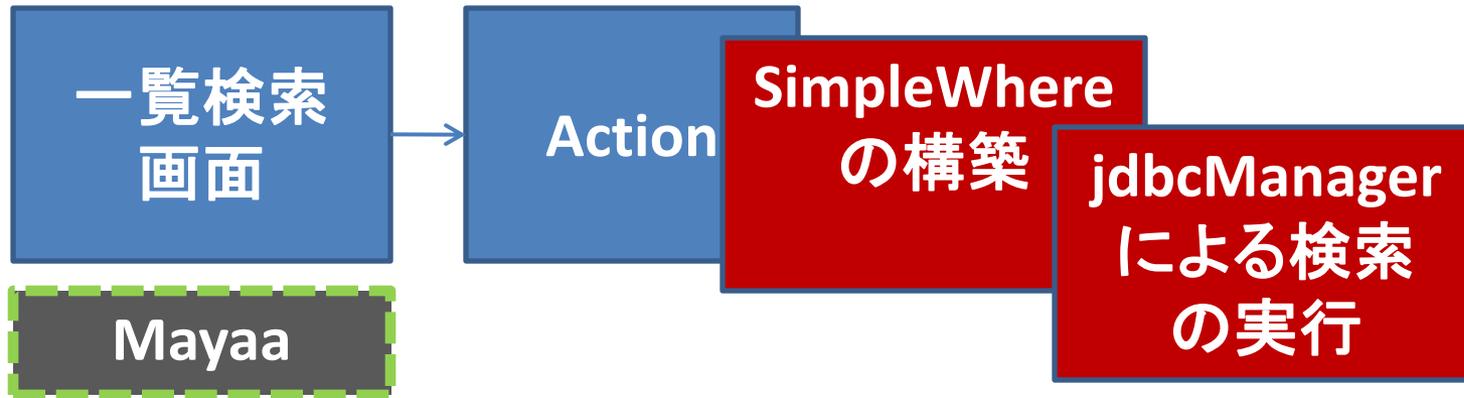
# 検索ロジックコード例 (Kuina-Dao)

```
public Class prerender() {  
    offset = empIndex;  
  
    EmpDto dto = new EmpDto();  
    dto.setMaxResults(limit);  
    dto.setFirstResult(empIndex);  
    setCondition(dto);  
    empltems = getEmpService().findByEmp(dto);  
  
    calculatePageIndex();  
  
    return null;  
}
```

# 検索ロジックコード例 (Kuina-Dao)

```
private void setCondition(EmpDto dto) {
    if (textSal != null) {
        dto.setSal_GE(textSal);
    }
    if (textHiredate != null) {
        dto.setHiredate_GE(textHiredate);
    }
    ...
    if (textEmpName == null || textEmpName.length() == 0) {
        dto.setEmpName_LIKE(textEmpName);
    } else {
        dto.setEmpName_LIKE(textEmpName + "%");
    }
}
```

# 検索ロジック (S2JDBC)



- 生成指示画面にて選択した検索条件に従って、検索ロジックを生成
  - 検索条件をSimpleWhereにて作成
  - jdbcManagerにより検索を実行
- デザインとロジックを分離したい場合は、Mayaaを利用

# 検索ロジックコード例 (S2JDBC)

```
@Execute(validator = false)
public String index() {
    SimpleWhere swh = new SimpleWhere()
        .ge("sal", empForm.sal != null && empForm.sal.length() > 0 ?
empForm.sal : 0)
        .ge("hiredate", empForm.hiredate != null &&
empForm.hiredate.length() > 0 ? empForm.hiredate : "1900/01/01")
        .ge("empNo", empForm.empNo != null && empForm.empNo.length() >
0 ? empForm.empNo : 0)
        .like("empName", empForm.empName+"%")
    ;
}
```

# 検索ロジックコード例 (S2JDBC)

```
emplItems = jdbcManager.from(Emp.class).where(swh)  
    .orderBy("SAL,HIREDATE,EMP_NO,EMP_NAME")  
    .limit(limit).offset(Integer.valueOf(empForm.offset))  
    .getResultList();
```

# 目次

- マスタ-ディテールScaffoldの紹介
  - マスタ-ディテールScaffoldに関して
  - マスタ-ディテール形式に関して
  - マスタ-ディテールScaffoldのデモ
  - 編集ロジック
  - 編集画面コード例
  - 編集ロジックコード例
- おわりに

# マスタ-ディテールScaffoldの紹介

# マスター-ディテールScaffoldに関して

- 機能

- マスタテーブルとディテールテーブルで対になっているデータをメンテナンスするアプリケーションを生成する

- 目的

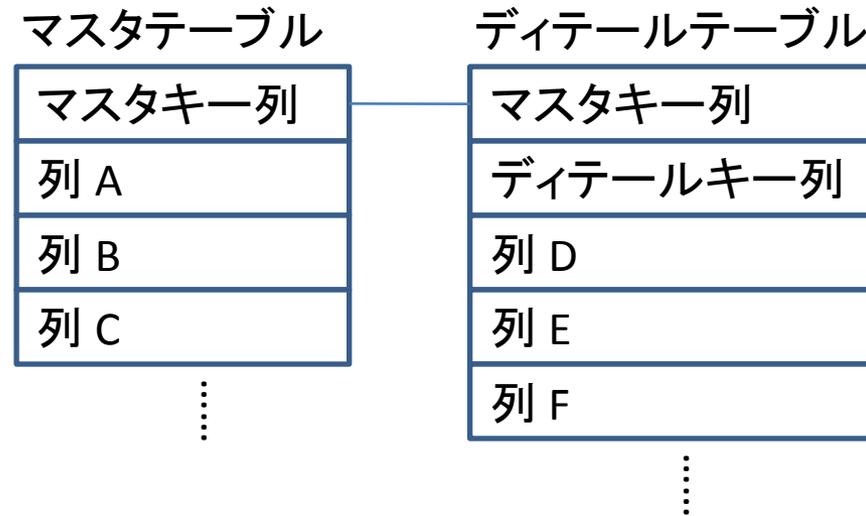
- 開発生産性の向上

- Scaffoldだけでは、生成後にディテールテーブルへのCRUDコードを追加する必要があり、時間がかかる
- 上記のコード追加作業を軽減するため

- 競合するフレームワークへのアピール

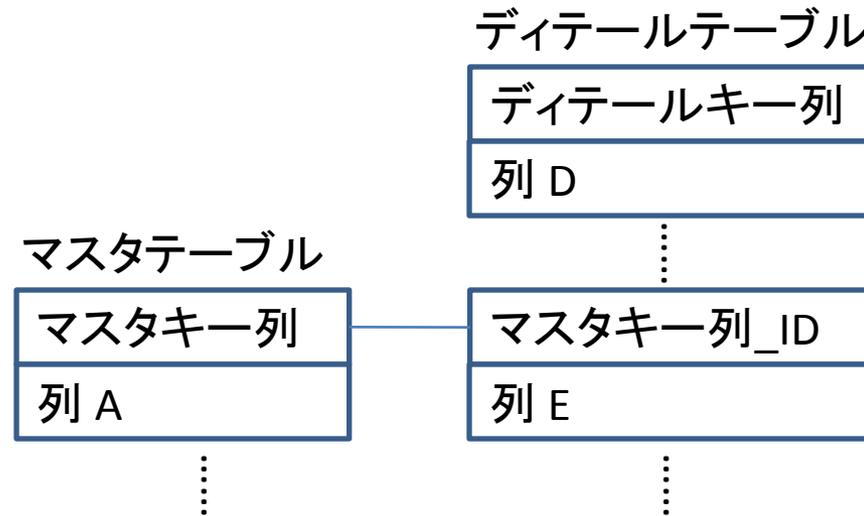
- Oracle ADFでもマスタディテール形式のAPを作成する仕組みが用意されている
- オープンソース側でもマスタディテール形式のAPを作成する仕組みを用意し、その適用範囲を広げるため

# マスタ-ディテール形式に関して(1)



- マスタテーブルのプライマリキー列が、ディテールテーブルのプライマリキー列の一部になっている形式

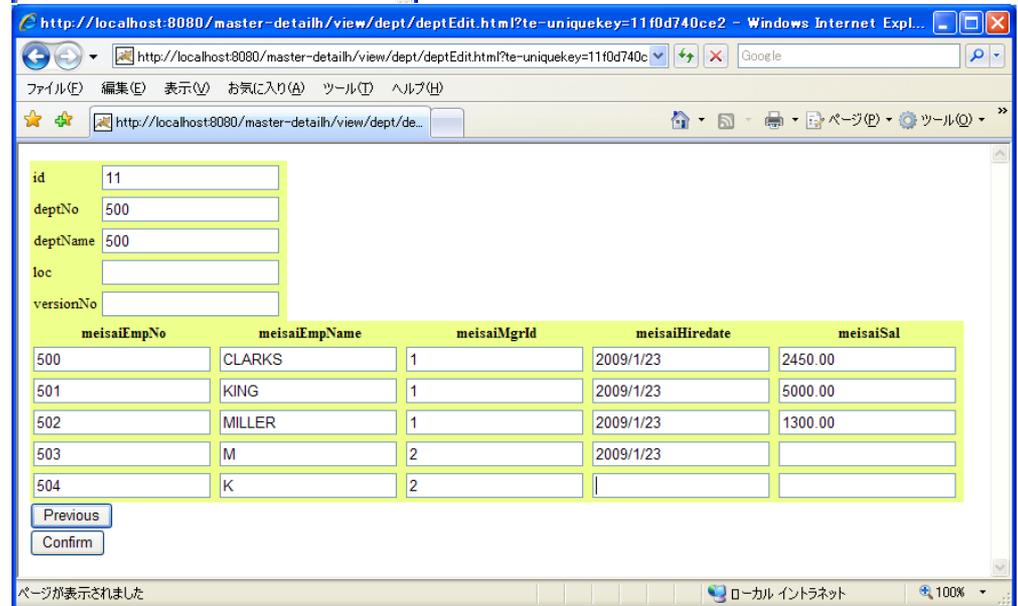
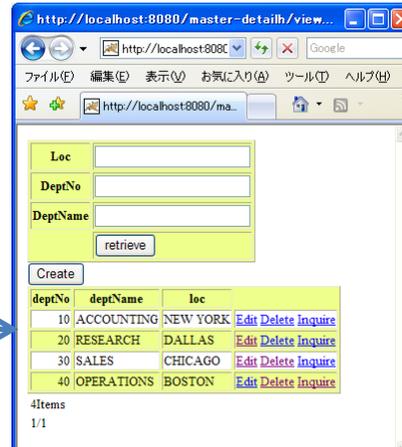
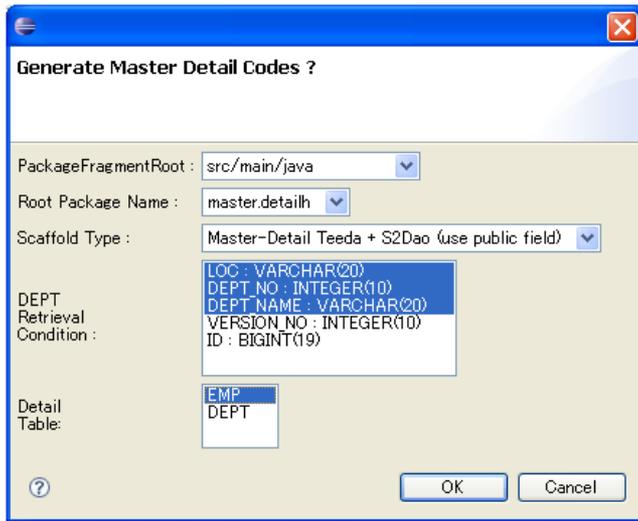
# マスタ-ディテール形式に関して(2)



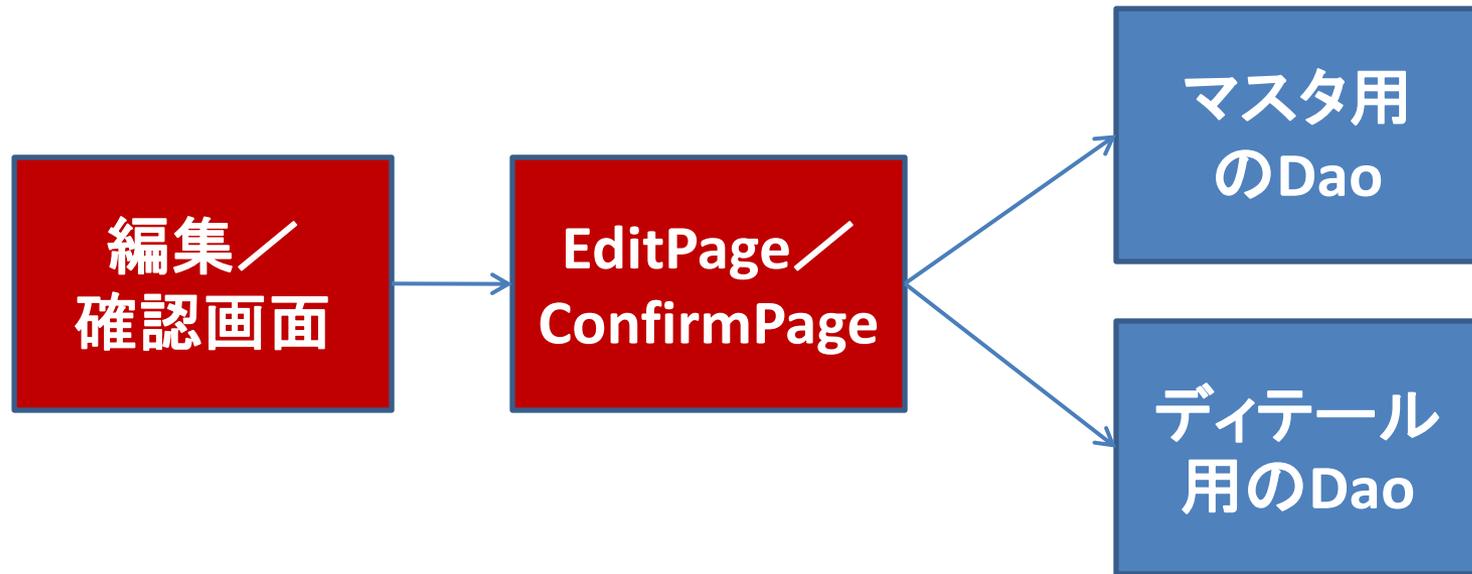
- マスタとディテールの結合を表す列名が、「マスタテーブル名+\_ID」となっている形式

# マスター-ディテールScaffoldのデモ

- S2Dao Public



# 編集ロジック



- 生成指示画面にて選択したマスタテーブルとディテールテーブルを編集するロジックを生成
  - 1件のマスタデータと複数件のディテールデータを扱える画面 (HTMLとPage.java) のコードを作成
  - マスタ用及びディテール用のDaoを使用して、更新を実行

# 編集画面コード例

id	<input type="text"/>
deptNo	<input type="text"/>
deptName	<input type="text"/>
loc	<input type="text"/>
versionNo	<input type="text"/>

マスタ部分

empNo	empName	mgrId	hiredate	sal
<input type="text"/>				

ディテール部分

```
<tbody id="empltms">
  <tr>
    <input type="hidden" id="meisaiId" />
    <td><input type="text" id="meisaiEmpNo" /></td>
    <td><span id="meisaiEmpNoMessage"></span></td>
    ...
    <td><input type="text" id="meisaiSal" /></td>
    <td><span id="meisaiSalMessage"></span></td>
    <input type="hidden" id="meisaiVersionNo" />
  </tr>
</tbody>
```

# 編集ロジックコード例

```
public Class doFinish() {  
    switch(super.crudType) {  
        case CrudType.CREATE:  
            ...  
        case CrudType.UPDATE:  
            deptDao.update(deptDxo.convert(this));  
            for (int i = 0; i < empltems.length; i++) {  
                empltems[i].meisaiDeptId = id;  
                EmpDao.update(empltems[i].convert());  
            }  
            break;  
        case CrudType.DELETE:  
            ...  
    }  
}
```

# おわりに

- 本セッションでの説明内容
  - 検索条件とページ移動が追加されたScaffold
    - S2Dao、Kuina-Dao、SAStruts、SAStruts+Mayaa
  - マスタ-ディテールScaffold
    - S2Dao Public
- 現在開発中
  - SAStruts+S2JDBC版の検索条件とページ移動が追加されたScaffold