

フレームワーク応用実習

05. ページ分割機能の実装

株式会社ジーードライブ

今回学ぶこと

- シンプルなページ分割機能の実装方法

ページ分割機能

- ・ ページネーション (pagination) と呼ばれる
- ・ リストとして表示する項目の件数が多い場合に、複数のページに分割して表示する機能
- ・ ページ移動のためのナビゲーションリンクを備える



簡単なナビゲーションの例

- 構成
 - ページ番号のリンク
 - 次のページへ移動するリンク
 - 前のページへ移動するリンク

[前のページ](#) | [1](#) | [2](#) | [3](#) | [次のページ](#)

ページ番号のリンク

- 現在表示しているページ番号はリンクを無効にする
 - 2ページ目を表示している場合の例



次のページへのリンク

- 現在表示しているページが最後のページではない場合、リンクを有効にする

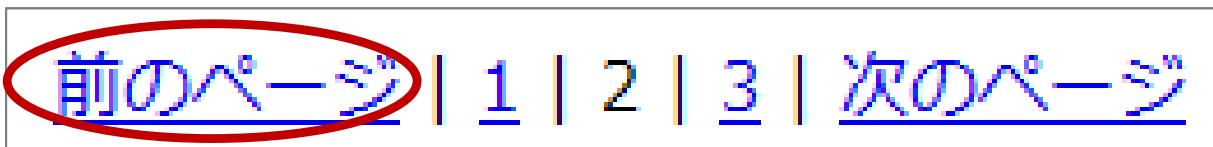
[前のページ](#) | [1](#) | [2](#) | [3](#) | [次のページ](#)

- 現在表示しているページが最後のページの場合、リンクを無効にする

[前のページ](#) | [1](#) | [2](#) | [3](#) | [次のページ](#)

前のページへのリンク

- 現在表示しているページが1ページ目ではない場合、リンクを有効にする



- 現在表示しているページが1ページ目の場合、リンクを無効にする



例題

membersテーブルに32件のデータがある。これらのデータを1ページあたり5件ずつ表示する場合、以下の問について考えてみましょう。

- 問 1. 全体のページ数は？
- 問 2. 4ページ目では、何件目～何件目までのデータを表示する？
- 問 3. 4ページ目に表示するデータをSQLで表すと？

例題

問1. 全体のページ数は？

$$\begin{array}{l} \text{トータル} \quad \text{ページあたり} \\ 32 \text{ 件} \div 5 \text{ 件} = 6 \cdots 2 \text{ 件} \rightarrow 7 \text{ ページ} \end{array}$$

問2. 4ページ目では、何件目～何件目までのデータを表示する？

問3. 4ページ目に表示するデータをSQLで表すと？

ページ	表示（何件目～何件目）	SQL文
1	1～5	SELECT * FROM members LIMIT 0, 5
2	6～10	SELECT * FROM members LIMIT 5, 5
3	11～15	SELECT * FROM members LIMIT 10, 5
4	16～20	SELECT * FROM members LIMIT 15, 5

公式化すると…

ページ当たりの表示件数 現在のページ
offset = numPerPage * (currentPage - 1);

MyBatisで使用するSQL文

SELECT * FROM members LIMIT #{offset}, #{numPerPage}

必要な処理

- ・ 全体のページ数の計算
- ・ 指定されたページ番号に該当するデータの取得
- ・ ページ番号ナビゲーションの作成

マッピングファイルの作成

- 以下のSQL文をマッピング用のファイルに記述する
 - データの全件数を取得するための COUNT() を含むSELECT文
 - ページごとのデータを取得するための LIMIT句を含むSELECT文

XMLの例

```
<select id="countMembers" resultType="long">
    SELECT COUNT(*) FROM members
</select>

<select id="getLimitedMembers" resultType="com.example.domain.Member">
    SELECT * FROM members
    LIMIT #{offset}, #{num}
</select>
```

インターフェースの例

```
Long countMembers();
List<Member> getLimitedMembers(@Param("offset") int offset,
                                @Param("num") int num);
```

補足:マッピングファイルの作成

- LIMIT句をもつSELECT文を記述する代わりにRowBoundsを利用して、取得するデータの範囲を指定することも可能

XMLの例: LIMIT句は記述しない

```
<select id="getMembers" resultType="com.example.domain.Member">
    SELECT * FROM members
</select>
```

インターフェースの例

```
List<Member> getMembers(); //全件取得
List<Member> getMembers(RowBounds bounds); //取得する範囲を指定
```

RowBoundsの利用例: 101件目から10件分のデータを取得する

```
List<Member> members = mapper.getMembers(new RowBounds(100, 10));
```

全体のページ数の計算

- 全体のページ数 = 全件数 ÷ 1ページあたりの表示件数
(小数点以下は切り上げ)
 - 小数点以下切り上げは `Math.ceil(N)` メソッドを使う

```
// 全件数の取得
Long count = mapper.countMembers();

// 全体のページ数の計算 : numPerPageは1ページあたりの表示件数とする
int totalPages =
    (int) Math.ceil((double) count / numPerPage);
```

マッピングファイルで定義しておいた
全件数取得のメソッド

ページ番号に該当するデータの取得

- LIMIT句を実装したメソッドを使い、ページに該当するデータを取得する

```
// page: ページ番号  
// numPerPage: 1ページ当たりの表示件数  
  
int offset = numPerPage * (page - 1);  
List<Member> members  
    = mapper.getLimitedMembers(offset, numPerPage);
```

マッピングファイルで定義しておいた
LIMIT句を実装したメソッド

表示するページ番号を取得する

- ページ番号はGETパラメータとして取得する
 - defaultValue属性：パラメータのデフォルト値の指定

```
@RequestMapping("/listMember")
public String list(
    @RequestParam(defaultValue="1") Integer page,
    Model model) throws Exception {
    ...
}
```

※ 対応するURLの例： /listMember?page=2

ページ番号ナビゲーションの作成

- 反復処理を使い、以下のようなHTMLを出力する
 - th:eachやth:switchを使用した煩雑な記述になる

3ページ目を表示している場合のHTML出力例：

```
<a href="?page=2">前のページ</a>
| <a href="?page=1">1</a>
| <a href="?page=2">2</a>
3
| <a href="?page=4">4</a>
| <a href="?page=5">5</a>
| <a href="?page=4">次のページ</a>
```

ブラウザ表示例

前のページ | 1 | 2 | 3 | 4 | 5 | 次のページ

練習

- 練習05