

employee アプリをつくる Ver1.2

Seiichi Nukayama

2021-12-27

目次

1	社員一覧を表示する	1
1.1	プロジェクト chap13 を作成する	1
1.2	GetEmployeeListLogic をつくる	1
1.3	EmployeeServlet サブレットをつくる	1
1.4	employee.jsp をつくる	1
1.5	できあがり図	2
1.6	手順をまとめてみる	2
2	新規社員登録機能を追加する	3
2.1	EmployeeDAO に create() メソッドをつくる	3
2.2	CreateEmployeeLogic をつくる	3
2.3	新規社員登録ボタン	3
2.4	EncFilter フィルターをつくる	3
2.5	CreateEmployeeServlet サブレットをつくる	3
2.6	createEmp.jsp をつくる	4
2.7	CreateConfirmServlet サブレットをつくる	4
2.8	createConfirm.jsp をつくる	4
2.9	CreateEmployeeServlet サブレットの doPost() メソッドをつくる	4
2.10	createResult.jsp をつくる	5

1 社員一覧を表示する

h2 の example データベースの employee テーブルを使った Web アプリをつくる。

1.1 プロジェクト chap13 を作成する

動的 Web プロジェクト chap13 を作成する。

example プロジェクトの以下のファイルを chap13 プロジェクトにコピーする。

パッケージ構成はそのままにする。

< コピーするファイル >

- dao.EmployeeDAO.java
- model.Employee.java

1.2 GetEmployeeListLogic をつくる

model パッケージに GetEmployeeListLogic.java を作成し、以下のメソッドを実装する。

1. execute() メソッド

引数: なし。

戻り値: List<Employee>

このメソッドを実行すると、EmployeeDAO の findAll() メソッドを実行し、Employee クラスの List を取得できる。

docoTsubu プロジェクトの GetMutterListLogic.java (p403) を参考にする。

1.3 EmployeeServlet サーブレットをつくる

servlet パッケージに EmployeeServlet.java を作成する。

1. @WebServlet アノテーション: "/employee"

2. doGet() メソッド:

GetEmployeeListLogic の execute() メソッドを実行し、employee のリストを取得する。

そのリストを "empList" という名前で リクエストスコープに保存する。

"/WEB-INF/jsp/employee.jsp" ヘフォワードする。

3. doPost() メソッド:

(今のところ、なにもしない)

1.4 employee.jsp をつくる

WebContent/WEB-INF/jsp に employee.jsp を作成する。

1. empList の内容を表示する。

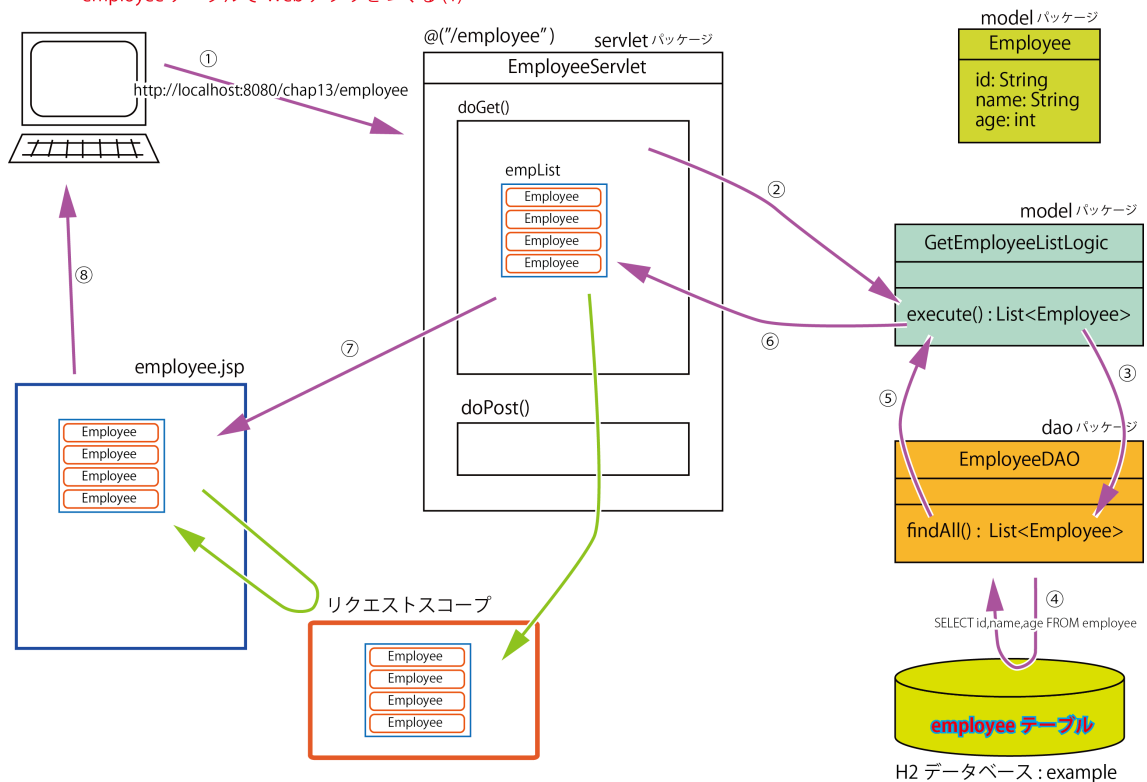
2. empList の各レコードの表示には <table>タグを使用する。

1.5 できあがり図



1.6 手順をまとめてみる

employee テーブルで Web アプリをつくる (1)



2 新規社員登録機能を追加する

新規社員登録機能をつけてみる。

2.1 EmployeeDAO に create() メソッドをつくる

p401, p402 のコードを参考にして、EmployeeDAO に create() メソッドをつくる。

1. 引数は Employee クラスのインスタンス
2. 戻値は boolean。登録成功で true。失敗で false。

2.2 CreateEmployeeLogic をつくる

p402 を参考に、CreateEmployeeLogic をつくる。

1. execute() メソッドをつくる。
2. その引数は、Employee インスタンス。
3. 戻値は boolean。成功は true。失敗は false。

2.3 新規社員登録ボタン

「新規社員登録」というボタンをつける。

「社員一覧」画面の上部につければよいだろう。

そのボタンには `"/chap13/create"` というリンクをつけておく。これは、`"CreateEmployeeServlet"` サーブレットの `WebServlet` アノテーションになる。そして、このサーブレットの `doGet()` メソッドが動作する。

2.4 EncFilter フィルターをつくる

フォームからは日本語が送られてくるので、文字エンコードを指定するフィルターを作成する。

1. filter パッケージに `EncFilter.java` という名前で作成する。
2. `request.setCharacterEncoding("UTF-8")` だけでよい。

2.5 CreateEmployeeServlet サーブレットをつくる

`servlet` パッケージに `CreateEmployeeServlet.java` を作成する。

1. `@WebServlet` アノテーション: `"/create"`
2. `doGet()` メソッド
`"/WEB-INF/jsp/createEmp.jsp"` へフォワードする。
3. `doPost()` メソッド

フォームからデータを受取り、データベースへの登録をおこなうようにする予定だが、今は何も書かない。

2.6 createEmp.jsp をつくる

WebContent/WEB-INF/jsp に createEmp.jsp を作成する。

1. フォームにて id, name, age を入力できるようにする。
2. 「確認」ボタンをクリックすることで、/chap13/createConfirm にデータを送信する。

2.7 CreateConfirmServlet サークレットをつくる

これは、入力された内容の確認画面を表示するためのサークルットである。

1. @WebServlet("/createConfirm")
2. フォームから id, user, age を取得する。
3. Employee クラスのインスタンスを作成する。(変数名を仮に emp とする)
4. emp をリクエストスコープに保存する。
5. WEB-INF/jsp/createConfirm.jsp にフォワードする。

2.8 createConfirm.jsp をつくる

リクエストスコープの emp を取得して、確認画面を作り、もし良ければ、/chap13/create にポスト送信する。

1. emp を使って、id, name, age を確認する。
2. もし良ければ「登録」ボタンをクリックする。送信先は "/chap13/create"。
3. もしだめならば、「取消」ボタンをクリックする。リンク先は "/chap13/create"。

登録ボタンをクリックすると、id, name, age の各データが、"/chap13/create" に post 送信される。これは "hidden" で送信すればよい。

2.9 CreateEmployeeServlet サークレットの doPost() メソッドをつくる

先ほどはこの doPost() メソッドは空白のままにしておいたが、今ここで実装する。

1. フォームから id, name, age を取得して、Employee クラスのインスタンスをつくる。
2. CreateEmployeeLogic のインスタンスを作成し、その execute() メソッドに Employee インスタンスを渡して、登録処理を行う。
3. もし登録がうまくいけば、文字列変数に "登録されました" を代入する。
4. もしうまくいかなければ、文字列変数に "失敗しました" を代入する。
5. msg をリクエストスコープに保存する。

6. /WEB-INF/jsp/createResult.jsp にフォワードする。

2.10 createResult.jsp をつくる

結果を確認する。

1. msg を表示する。
2. /chap13/employee に遷移する。