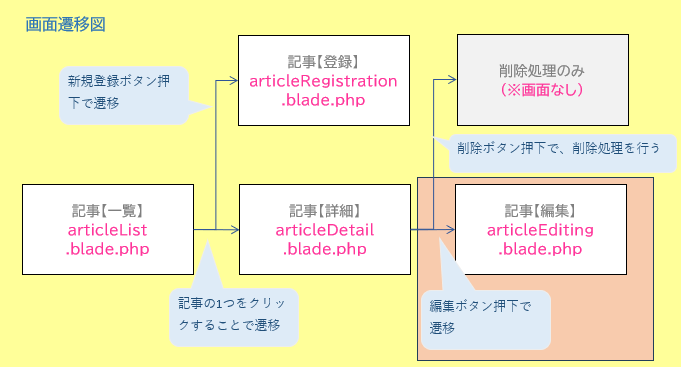
第８回　DB処理（UPDATE）

**【課題は２問】＊画面は1画面**

**記事編集画面を作成する**こと。（一覧画面→記事詳細→記事編集の順番で画面遷移すること）

＊赤背景部分が課題８で作成する画面処理です。



**【使用ファイル・設定一覧】**

■コントローラ

**ArticlesController.php**（第6回で作成済み）

　＊editメソッド、updateメソッドの処理を追加

■ビュー（ファイルは第3回で作成済み）

**articleDetail.blade.php（記事詳細画面から記事編集画面への遷移が必要）**

**articleEditing.blade.php（記事編集画面）**

■モデル

**Article.php**（第６回で作成済み）

■ルーティング

web.php（ルーティングは第6回で設定済み）

■ＤＢ（第5回で作成済み）

articlesテーブル

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 型 | 制約 | 備考 |
| id | AUTO\_INCREMENT | UNSIGNED |  |
| title | VARCHAR(255) | ----- | タイトル |
| body | TEXT | NULL許可 | 本文 |
| created\_at | TIMESTAMP | CURRENT\_TIMESTAMPを  デフォルト値として使用 | 作成日時 |
| updated\_at | TIMESTAMP | ●CURRENT\_TIMESTAMPを  デフォルト値として使用  ●レコードが更新されたときに  CURRENT\_TIMESTAMPを使用 | 更新日時 |
| deleted\_at | ソフトデリート用  TIMESTAMPカラム | ------ | 論理削除用  （＊） |

課題８－１　記事編集画面

講義資料を参考に作成すること。（一連の動作はサンプルサイトにて確認可能）

記事詳細画面にて、「編集」ボタンを選択すると、記事編集画面へ遷移すること。

＊記事詳細画面と同じ記事idのものが表示されていること。

＊「更新」ボタン押下時、「タイトル」「記事」の入力値チェックを行う。

　→エラーがあった場合、入力値は保持し、エラーメッセージを表示する。

→入力値に問題がない場合、articlesテーブルのデータを正しく更新する。

＊DBのデータ更新が完了したら、記事詳細画面へリダイレクトすること。

＊「戻る」ボタン押下時、記事詳細画面へ遷移すること。かつ、記事の内容が正しく表示されていること。

（課題８では「画像」更新はしないものとする。）

（「削除」ボタンの処理は課題８では未実装でOK。課題９で行います）

課題８―２　Extra

記事詳細画面に「本文」の改行が反映されること。

＊この時、HTMLタグ・記号・特殊文字が文字列として扱われる処理が無効にならないこと。

（例：「&yen;」は文字としてそのまま表示される。「\」表示されないこと。）

OK・NGパターンは次のページに画像あり。

【ヒント】

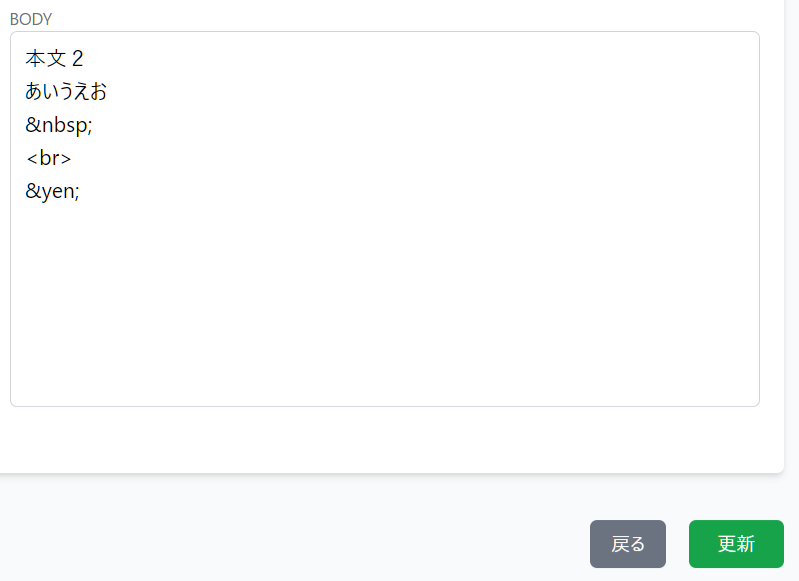
＊変更するのは表示時のみ。

＊DB処理などは一切変更しない。

＊下記を使用

1. nl2br
2. htmlspecialchars

■編集画面　＊「本文」入力値のみ抜粋　＊記事登録画面から遷移した場合も動作は同じ。

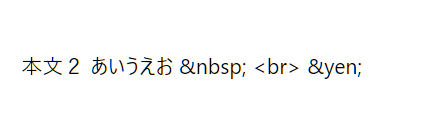
　（更新ボタン押下）

　　　　　　　　　　　　　　　　　　　　　　　　（編集画面または新規登録画面で更新ボタン押下）

■デフォルト（課題８－１完了時点）

＊詳細画面「本文」部分のみ抜粋。

＊すべて1行で表示される。



**こちらのOKパターンになるように修正。**

|  |  |
| --- | --- |
| ■NGパターン | ■OKパターン |
| ＊詳細画面「本文」部分のみ抜粋。  ＊タグや特殊文字が文字列として扱われていない。 | ＊詳細画面「本文」部分のみ抜粋。  ＊改行され、入力値はすべて文字列として扱われる。 |