

データベースの処理 - 更新と削除

レコードの更新

「--resource」で作成されたコントローラー内の「**edit**」メソッドと「**update**」メソッドを使って、登録内容の編集画面とレコードの更新処理を実装します。

ルーティングのメソッド

```
<form action="kadai06_1/{ 投稿記事のID }" method="POST">
  @method( "PUT" )
  <button type="submit">更新</button>
</form>
```

コントローラーのupdateメソッド

引数：articlesテーブルのID

storeメソッドとの違いは、「レコードの挿入」と「レコードの更新」になります。

更新する対象のIDが引数で送られてきますので、そのIDを使って、更新対象のレコード情報を抽出してから更新となります。

```
/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $this->validate( $request, $articleDao::$rules, $articleDao::$messages );

    $articleDao = new Article();
    $article = $articleDao->find( $id );
    $article->title = "更新したサンプルのテキスト";

    DB::transaction(function() use( $article ) {
        $article->save();
    });

    return redirect()->route( "sample06_1.show", $article->id );
}
```

レコードの削除

「--resource」で作成されたコントローラー内の「**destroy**」メソッドを使って、レコードの削除処理を実装します。

ルーティングのメソッド

レコードの削除には、ルーティングに「**DELETE**」メソッドを指定することになっていますので、formタグを使って削除対象のレコードIDなどを送信するようにします。

```
<form action="sample06_1/{ 投稿記事のID }" method="POST">
  @method( "DELETE" )
  <button type="submit">削除</button>
</form>
```

コントローラーのdestroyメソッド

引数：articlesテーブルのID

```
/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    $articleDao = new Article();

    DB::transaction(function() use( $articleDao, $id ){
        $articleDao->destroy( $id );
    });

    return redirect()->route( "kadai06_1.index" );
}
```

課題

- [kadai09_1](#)
- [kadai09_2](#)