

データベースの処理 - 挿入

レコードの挿入

「`--resource`」で作成されたコントローラー内の「**create**」メソッドと「**store**」メソッドを使って、新規登録の入力画面とレコードの挿入処理を実装します。

モデルクラスのsaveメソッド

```
public function store(Request $request)
{
    $articleDao = new Article();
    $articleDao->title = "サンプルのタイトル";

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

    return redirect()->route( "sample06.index" );
}
```

モデルにバリデーションルールとエラーメッセージの定義

`Rquest`クラスを用意しなくても、モデルクラスにバリデーションルールとエラーメッセージを定義することができます。

バリデーションルール

モデルクラス内に「**\$rules**」という名前のプロパティで、バリデーションルールを定義します。

```
// バリデーションルール
public static $rules = [
    "title" => [ "required" ],
];
```

エラーメッセージ

モデルクラスに「**\$messages**」という名前のプロパティで、バリデーションルールに対するエラーメッセージを定義します。

```
// エラーメッセージ
public static $messages = [
```

```
"title.required" => "タイトルが入力されていません",
];
```

Requestクラスで定義していたメソッド名と同じプロパティ名となりますので、覚えやすいです。

モデルクラスのリクエストルールとエラーメッセージの使用

```
/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->session()->regenerateToken();
    $this->validate( $request, $articleDao::$rules, $articleDao::$messages );

    $articleDao = new Article();
    $articleDao->title = $request->input( "title" );

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

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

課題

- [kadai08_1](#)