リクエストとバリデーション

フォームリクエスト

CSRFとは

CSRF（クロスサイトリクエストフォージェリ）は、Webアプリケーションに存在する脆弱性やその脆弱性を利用した攻撃方法の１つで、お問い合わせのようなリクエストなどを処理するWebアプリケーションが、本来拒否すべき他のサイトからのリクエストを受信して処理してしまうことをいいます。

トレンドマイクロ - CSRF
https://www.trendmicro.com/ja_jp/security-intelligence/research-reports/threat-solution/csrf.html

「 **@csrf** 」はCSRF対策のために用意されているBladeディレクティブで、CSRF対策用の「 **Token（トークン）** 」と呼ばれるランダムな文字列を非表示フィールドとして挿入してくれます。そして、CSRFトークン値が正しいリクエストだけを受け付けるようになります。

```
<input type="hidden" name="_token" value="f6z4YiXLzNIy49QOT97pQljWcD5NRLTbW3hQzetd">
```

```
@section('content')
<form action="sample04" method="POST">
    @csrf
    <div>
        <label for="user_name">名前</label>
        <input type="text" id="user_name" name="user_name" value="" placeholder="ECC 太郎" required>
    </div>
    <div>
        <label for="email">メールアドレス</label>
        <input type="email" id="email" name="email" value="" placeholder="example@ecc.com" required>
    </div>
</form>
@endsection
```

二重送信の防止

リクエストのたびにCSRFトークンを作り直すことで、コンピュータからの連続リクエストや誤操作による二重送信を防止することができます。

```
// CSRFトークンの破棄
$request->session()->regenerateToken();
```

POSTメソッドのルーティング

```
Route::get('sample04', [Kadai04_1Controller::class, 'index']);
Route::post('sample04', [Kadai04_1Controller::class, 'post']);
```

リクエストデータの取得

リクエストデータを取り出すには、「 **Request** 」クラスの「 **input** 」メソッドを使います。

メソッドの引数にRequestクラスを指定します。

リクエストクラス

```
use use Illuminate\Http\Request;

public function post(Request $request) {

}
```

リクエストデータへのバリデーション

バリデーションルールの定義

```
use use Illuminate\Http\Request;

public function post(Request $request) {

    $request->session()->regenerateToken();

    $request->validate(
        [
            'name'  => ['required'],
            'email' => ['required', 'email'],
        ]
    );

    return view('sample04', );
}
```

リクエストクラスの作成

```
php artisan make:request Sample04Request
```

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class Sample04Request extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
```

```
     *
     * @return array
     */
    public function rules()
    {
        return [
            //
        ];
    }

}
```

バリデーションルールの作成

```
public function rules()
{

    return [
        'name'       => ['required', ],
        'email'      => ['email', ],
    ];

}
```

エラーメッセージの作成

```
public function messages() {

    return [
        'name.required'       => '名前が入力されていません',
        'email.email'         => 'メールアドレスの形式が間違っています',
    ];

}
```

作成したリクエストクラスの利用

```
use App\Http\Requests\Sample0404Request;

public function post(Sample04Request $request) {

}
```