Laravel - Response

Topics : Laravel
Written on December 22, 2023

In Laravel, the `Response` class is used to build HTTP responses that can be sent back to the client. It is part of the Illuminate HTTP package and provides a convenient way to work with HTTP responses. Here are some common operations and features related to the `Response` class in Laravel:

Creating Responses:

You can create responses using various methods. For example, you can use the `response` helper function or the `Response` facade:

```
use Illuminate\Support\Facades\Response;

// Using the response helper function
$response = response('Hello, World!');

// Using the Response facade
$response = Response::make('Hello, World!');
```

Sending JSON Responses:

You can easily send JSON responses using the `json` method:

```
return response()->json(['key' => 'value']);
```

Setting Status Code:

You can set the HTTP status code for the response:

```
return response('Not Found', 404);
```

Attaching Headers:

You can attach headers to the response:

```
return response('Hello')->header('Content-Type', 'text/plain');
```

Redirecting:

You can create redirects using the `redirect` method:

```
return redirect('new-url');
```

Attaching Cookies:

As mentioned earlier, you can attach cookies to the response:

```
return response('Hello')->cookie('username', 'john_doe', 30);
```

File Downloads:

You can force the user's browser to download a file:

```
return response()->download($pathToFile);
```

Views and HTML Responses:

You can return HTML content or views as responses:

```
return response()->view('welcome');
```

Response Macros:

You can extend the `Response` class with your own macros for custom functionality:

```
use Illuminate\Support\Facades\Response;

Response::macro('uppercase', function ($value) {
    return response(strtoupper($value));
});

// Usage
return response()->uppercase('hello');
```

Response Instances:

You can manipulate response instances using various methods, such as `with`, `withErrors`, and `header`:

```
return response('Hello')->with('key', 'value')->withErrors(['error' =>
'Something went wrong']);
```