

HTTP Responses

Generating HTTP Responses

Anonymous · 09/09/2026

Generating HTTP Responses

Derafu Spreadsheet makes it easy to generate downloadable spreadsheet files directly from your web application. This guide shows how to use the library to create HTTP responses for spreadsheet downloads.

PSR-7 Response Integration

Derafu Spreadsheet includes a PSR-7 compatible response generator for easy integration with frameworks that support PSR-7 standards.

Requirements

To use the HTTP response features, you need to install Nyholm's PSR-7 implementation:

```
composer require nyholm/psr7
```

If you don't want to use nyholm/psr7, you can use any PSR-7 compatible library implementing by your own SpreadsheetHttpResponseGeneratorInterface.

Basic Usage

```
<?php

use Derafu\Spreadsheet\Http\NyholmSpreadsheetHttpResponseGenerator;
use Derafu\Spreadsheet\Spreadsheet;

// Create or load a spreadsheet.
$spreadsheet = new Spreadsheet();
$spreadsheet->createSheet('Sheet1', [
    ['Name', 'Email', 'Age'],
    ['John Doe', 'john@example.com', 30],
    ['Jane Smith', 'jane@example.com', 25]
]);

// Create response generator.
```

```
$responseGenerator = new NyholmSpreadsheetHttpResponseGenerator();

// Generate PSR-7 response with auto-detected format (xlsx).
$response = $responseGenerator->createResponse(
    $spreadsheet,
    'users-export.xlsx'
);

// The response can now be sent by any PSR-7 compatible framework.
```

Specifying Format

You can explicitly specify the format for the response:

```
$response = $responseGenerator->createResponse(
    $spreadsheet,
    'users-export.csv',
    'csv' // Explicitly specify format.
);
```

PSR-7 Compatible Frameworks (Slim, Mezzio, etc.)

With PSR-7 compatible frameworks, you can directly use the `NyholmSpreadsheetHttpResponseGenerator`:

```
use Derafu\Spreadsheet\Http\NyholmSpreadsheetHttpResponseGenerator;
use Derafu\Spreadsheet\SpreadsheetLoader;
use Psr\Http\Message\ResponseInterface;
use Psr\Http\Message\ServerRequestInterface;

class ExportAction
{
    public function __invoke(ServerRequestInterface $request): ResponseInterface
    {
        // Create or load your spreadsheet
        $loader = new Loader();
        $spreadsheet = $loader->loadFromFile('data.xlsx');

        // Modify it if needed
        // ...

        // Generate response
        $responseGenerator = new NyholmSpreadsheetHttpResponseGenerator();
        return $responseGenerator->createResponse(
            $spreadsheet,
            'exported-data.xlsx'
        );
    }
}
```

```
}
```

Custom Response Generation

If you need to customize the response generation or use a different PSR-7 implementation, you can implement your own `SpreadsheetHttpResponseGeneratorInterface`:

```
<?php

namespace App\Spreadsheet;

use Derafu\Spreadsheet\Contract\Http\SpreadsheetHttpResponseGeneratorInterface;
use Derafu\Spreadsheet\Contract\SpreadsheetInterface;
use Psr\Http\Message\ResponseInterface;

class CustomResponseGenerator implements SpreadsheetHttpResponseGeneratorInterface
{
    public function createResponse(
        SpreadsheetInterface $spreadsheet,
        ?string $filename = null,
        ?string $format = null
    ): ResponseInterface {
        // Your custom implementation
    }
}
```

MIME Types for Different Formats

When generating HTTP responses, it's important to use the correct MIME type. Derafu Spreadsheet handles this for you, but here's a reference of the MIME types used for each format:

Format	MIME Type
XLSX	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
XLS	application/vnd.ms-excel
CSV	text/csv
ODS	application/vnd.oasis.opendocument.spreadsheet
JSON	application/json
XML	application/xml
YAML	application/yaml
HTML	text/html

Format	MIME Type
PDF	application/pdf

Security Considerations

When generating downloadable files from user data, be sure to:

1. **Sanitize data:** Ensure user-provided data doesn't contain malicious content.
2. **Validate filenames:** Clean and validate user-provided filenames.
3. **Set appropriate headers:** Ensure you're using the correct content type and disposition.
4. **Clean up temporary files:** Remove any temporary files after sending the response.

Performance Tips

For large spreadsheets, generating the response can be resource-intensive. Consider:

1. **Queuing exports:** For large exports, process them in a background job and notify the user when ready.
2. **Streaming responses:** Some frameworks support streaming responses to reduce memory usage.
3. **Pagination:** Consider exporting data in smaller batches if possible.

Last updated on 09/09/2026