

Laravel Job Batching

Laravel's **job batching** feature allows you to group multiple jobs into a batch and perform actions when the batch completes, fails, or progresses. This is particularly useful for tasks like processing large datasets or executing dependent jobs.

Defining Batchable Jobs

To make a job batchable, include the `Illuminate\Bus\Batchable` trait in your job class. This provides access to the `batch()` method, which allows the job to interact with its batch. For example:

```
namespace App\Jobs;

use Illuminate\Bus\Batchable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Foundation\Queue\Queueable;

class ImportCsv implements ShouldQueue
{
    use Batchable, Queueable;

    public function handle(): void
    {
        if ($this->batch()->cancelled()) {
            // Stop processing if the batch is cancelled
            return;
        }

        // Process the CSV data
    }
}
```

Dispatching Job Batches

You can dispatch a batch of jobs using the `Bus::batch()` method. This method also supports completion callbacks like *then*, *catch*, and *finally*:

```
use App\Jobs\ImportCsv;
use Illuminate\Bus\Batch;
use Illuminate\Support\Facades\Bus;

$batch = Bus::batch([
    new ImportCsv(1, 100),
    new ImportCsv(101, 200),
    new ImportCsv(201, 300),
```

```
]->then(function (Batch $batch) {
    // All jobs completed successfully
})->catch(function (Batch $batch, Throwable $e) {
    // Handle the first job failure
})->finally(function (Batch $batch) {
    // The batch has finished executing
})->dispatch();
```

Monitoring and Managing Batches

Laravel provides tools to inspect and manage batches:

- `$batch->progress()` returns the completion percentage.
- `$batch->cancel()` cancels the batch.
- `$batch->failedJobs` retrieves the number of failed jobs.

You can also retrieve a batch by its ID using `Bus::findBatch($batchId)`.

Adding Jobs Dynamically

You can add jobs to an existing batch from within a batch job using the `add()` method:

```
$this->batch()->add([
    new ImportCsv(401, 500),
    new ImportCsv(501, 600),
]);
```

Batch Failures and Pruning

When a job in a batch fails, the `catch` callback is triggered. By default, a failed job cancels the entire batch, but you can override this behavior using the `allowFailures()` method. To clean up old batch records, schedule the `queue:prune-batches` Artisan command.

Important Considerations

- **Database Migration:** Ensure you have a `job_batches` table by running `php artisan make:queue-batches-table` and migrating it.
- **Callbacks:** Avoid using `$this` in batch callbacks as they are serialized.
- **Batch Naming:** Use the `name()` method to assign a name for better debugging in tools like Horizon.

Laravel's job batching simplifies handling complex workflows by grouping jobs and providing robust tools for monitoring and managing their execution.

Revision #1

Created 17 January 2026 09:45:43 by AI Channel

Updated 17 January 2026 09:46:13 by AI Channel