Background Job Runner Documentation

1. Overview

The Background Job Runner is a system designed to execute PHP class methods as background jobs, with configurable retry logic, logging, and job priorities. It is built to handle background tasks in a Laravel-based application, with a focus on ensuring that jobs can be retried in case of failure, and that execution details are logged for monitoring purposes.

2. Usage Instructions

To use the background job runner, invoke the runBackgroundJob function with the class name, method name, and any parameters required by the method.

Usage command example:

php BackgroundJobRunner.php <ClassName> <MethodName> <param1> <param2> ...

3. BackgroundJob.php Function Explanation

3.1. logJob()

The logJob function logs job execution details to the log file with a timestamp.

Parameters:
- message: The message to log.

3.2. runBackgroundJob()

The runBackgroundJob function executes the specified method of a given class in the background, with retry logic and logging.

Parameters:
- className: The fully qualified class name (including namespace).
- method: The method name to call.
- params (optional): An array of parameters to pass to the method.

The function will attempt to execute the method up to the maximum retry attempts defined by the constant MAX_RETRY_ATTEMPTS. If execution fails, it will log the error and retry after a configurable delay.

4. BackgroundRunnerJob.php Function Explanation

4.1. runBackgroundJob()

The runBackgroundJob function in BackgroundRunnerJob.php extends the functionality of the original runBackgroundJob by allowing for dynamic configuration options such as retry count, delay between retries, and job priority.

Parameters:
- className: The fully qualified class name (including namespace).
- method: The method name to call.
- params (optional): An array of parameters to pass to the method.
- config (optional): An array of configuration options:
  - maxRetries: The maximum number of retry attempts.
  - retryDelay: The delay in seconds between retry attempts.
  - priority: The job priority, can be 'normal' or 'high'.

5. Job Configuration and Retry Settings

Job execution can be customized using several configuration options.

1. MAX_RETRY_ATTEMPTS: The maximum number of retry attempts for a job.

2. RETRY_DELAY: The delay in seconds between retry attempts.

3. Retry Settings in BackgroundRunnerJob.php:

You can configure the retry attempts and delay directly by passing a 'config' array to the runBackgroundJob function. The default settings are 3 retries and a 5-second delay.

6. Examples

6.1. Example 1: Run a Job Without Parameters

To run a job without parameters, use the following command:
php BackgroundJobRunner.php App\Jobs\JobClass executeTask

6.2. Example 2: Run a Job With Parameters

To run a job with parameters, include them after the method name:
php BackgroundJobRunner.php App\Jobs\JobClass executeTask param1 param2

6.3. Example 3: Run a Job with Custom Retry Settings

To run a job with custom retry settings, use the following syntax:
php BackgroundJobRunner.php App\Jobs\JobClass executeTask param1 param2 --retries=5 --delay=10

7. Error Handling and Logging

The system includes error handling that logs any failures during job execution. Each failed attempt is logged with the retry count and a message indicating the reason for failure.

If a job fails after the maximum retry attempts, the failure is logged with the timestamp and the total number of retry attempts.