

# Listeners System

---

This directory contains event listeners that process events dispatched by the application.

## Listeners

### ObserverEventListener

- **Event:** ObserverEvent
- **Purpose:** Handles observer-related actions
- **Queue:** Background processing enabled
- **File:** ObserverEventListener.php

### AdventurerEventListener

- **Event:** AdventurerEvent
- **Purpose:** Handles adventurer-related actions
- **Queue:** Background processing enabled
- **File:** AdventurerEventListener.php

## Implementation Details

All listeners implement the **ShouldQueue** interface, meaning they run in the background via Laravel's queue system. This ensures that event processing doesn't block the main application flow.

## Adding New Listeners

1. Create a new listener class in this directory
2. Implement the **ShouldQueue** interface if background processing is needed
3. Add the event-listener mapping in **EventServiceProvider.php**
4. Update the Events README with the new listener details

---

PROF

## Queue Configuration

Make sure your queue system is properly configured in your **.env** file:

```
QUEUE_CONNECTION=database
```

And run the queue worker:

```
php artisan queue:work
```

## Error Handling

Listeners should include proper error handling and logging:

```
public function handle(Event $event): void
{
    try {
        // Process the event
    } catch (\Exception $e) {
        \Log::error('Event processing failed', [
            'event' => get_class($event),
            'error' => $e->getMessage()
        ]);
    }
}
```