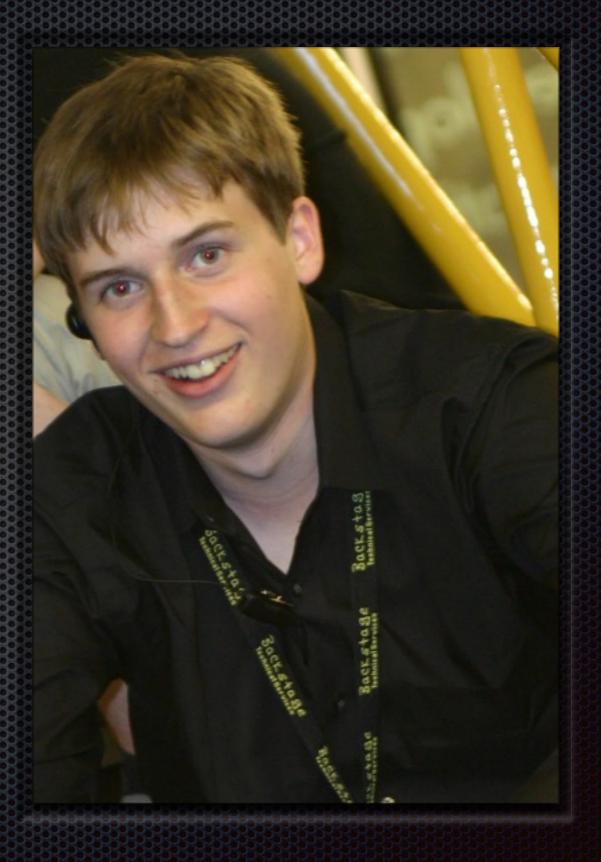# Introducing Amazon SQS

Meet Lee

@leesto

- PHPSW Organiser
- Web Team Lead for Gradwell
- Theatre Tech
- Theme Park Fan
- Occasional Photographer

What is SQS

"Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. A queue is a temporary repository for messages that are awaiting processing."

–Amazon Web Services

Why Queues?

- Ideal for tasks we don't need to complete in real time

    - Don't keep the user waiting

- Help handle incoming data / content

- Help scale large tasks

Why SQS?

- Amazon runs the service for you

- High availability

- Free for the first 1,000,000 requests per month

  - $0.50 per 1 million after

- AWS SDK

Queues

* Hold the messages to be processed

* Uniquely named per account

* Ability to build in a processing delay (up to 15 minutes)

* Can be created / configured through the UI or API

Messages

- Contains the actual job you want processed

- 1 blob - up to 256 KB

  - Found it useful to have this as a JSON object

- Option for storing additional attributes

Create a Message

```php
$this->awsClient->sendMessage(array(
    'QueueUrl' => 'http://sqs.us-east-1.amazonaws.com/123456789012/
myQueue',
    'MessageBody' => 'Hello World',
));
```

- Need to know the URL - separate call to get the URL from a name

Retrieve a Message

```
$this->awsClient->receiveMessage(array(
    'QueueUrl' => 'http://sqs.us-east-1.amazonaws.com/123456789012/
myQueue',
));
```

- Returns a Response model - check for Messages array

- Can implement long polling

- Queues define a 'message visibility' period
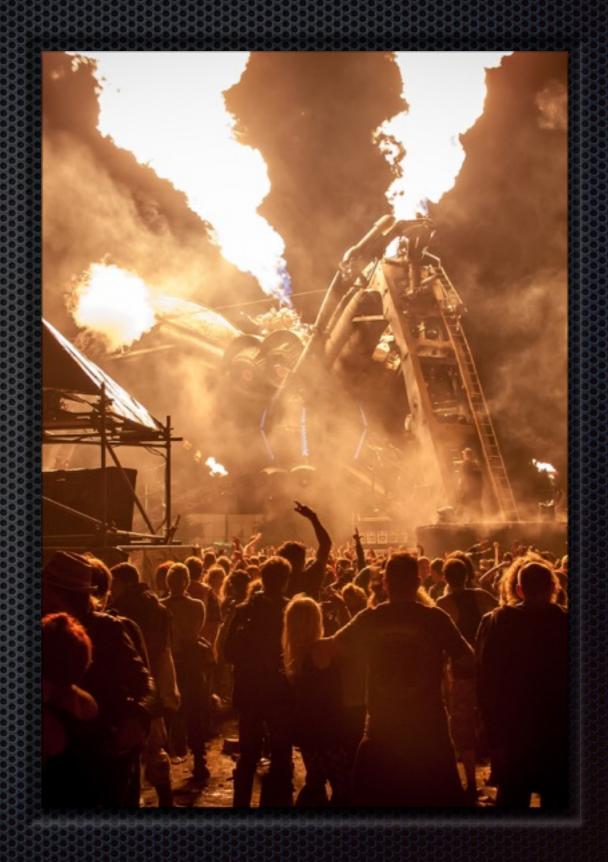
Gotchas

- First in, First out not guaranteed

- They guarantee 'at least once' delivery

- Retention Periods - default 4 days, max 14 days

Wait… There's More

- Security Policies

- Message Attributes

- Dead Letter Queues

- Batch Processing

Thank You

Questions?

joind.in/18340

@leesto

lee@leestone.co.uk