

# PHP Worker Manager

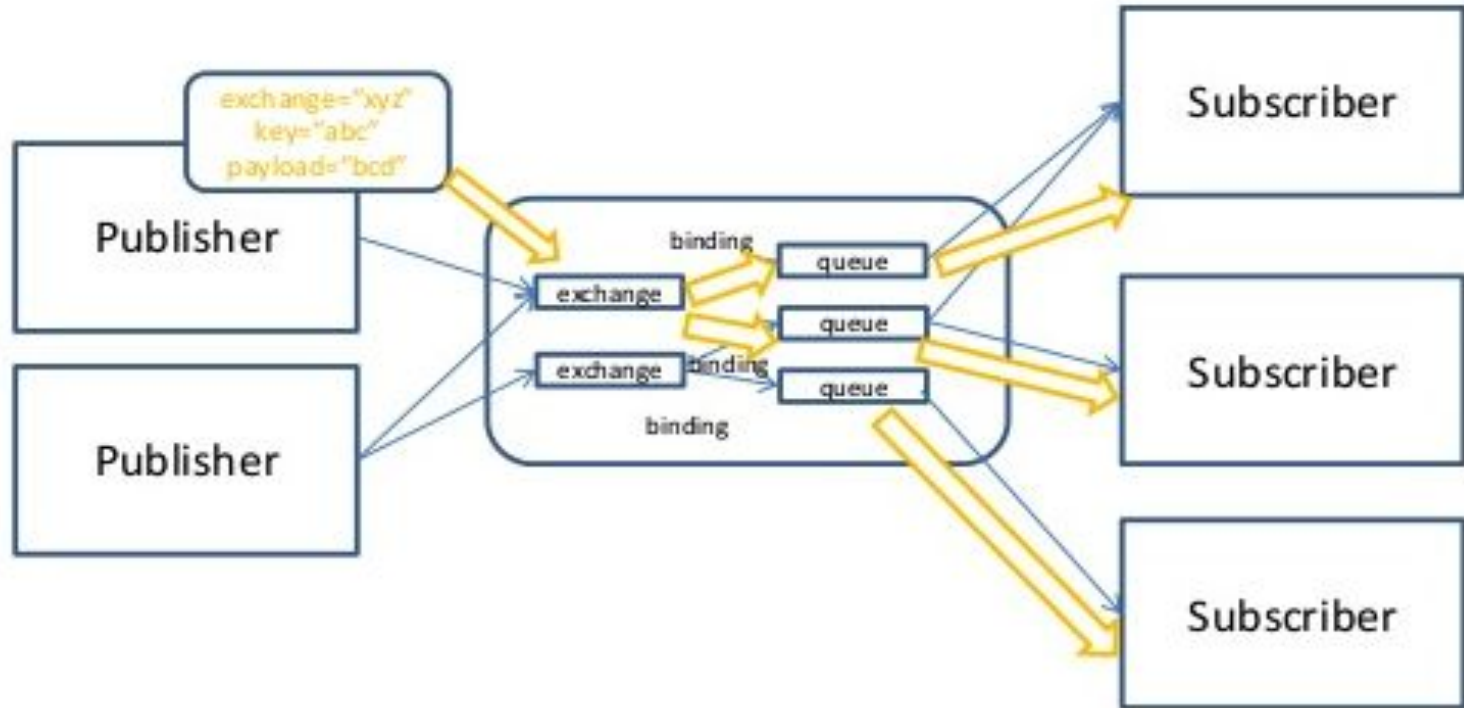
Employee Management

# Overview

1. Queueing the tasks (RabbitMq).
2. Create task.
3. Manage workers for background process.
4. Worker implementation.

# Queueing the tasks (RabbitMq)

- RabbitMq



# Create task

- Upload the file and make a record id for that file name.
- Create a queue in rabbitMq for the type of task.
- Queue the record id in respective queue.
- Record structure.
  - importId-fileName-status-completedRows-createdAt

# Manage workers for background process

- Workers are simple php process of a file running parallel.
- The count of workers are managed by below two requirements.
  - How cost the job is.
  - How much priority for the task.
- According to the no of tasks in queue need to increase and decrease the count of workers using worker manager.
- The worker details are stored in the redis for better management and tracking the process.

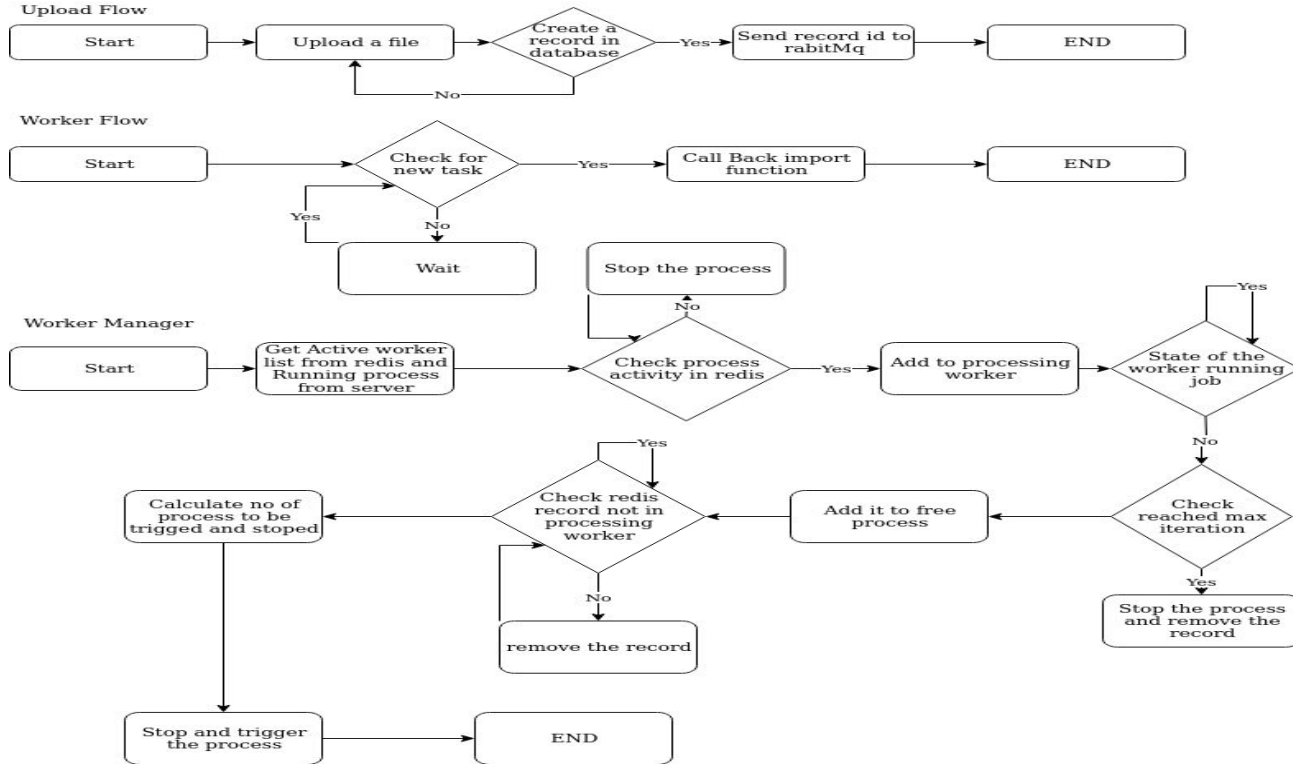
# Worker implementation.

- The workers are simple PHP file running without any dependency and parallelly.
- Here need to implement the worker to listen the respective queue for new tasks.
- If a new task comes to a queue means it get that task details and process the task.

# Import implementation

- Get import id from the rabbitMq.
- Get the file of the record.
- Chunk process the record to avoid failure scenarios.
- Update the result to the database.

# Workflow





End

Presented By.

Goplakrishnan.K