De cero a celery

Python Patagonia Meetup

¿Celery?

¿Qué es eso?

- Es una **cola de tareas** basada en pasaje de mensajes.
- Utiliza el concepto de **workers** para ejecutar las tareas, tanto de manera síncrona o asíncrona.
- Necesita un **Broker de Mensajería** para comunicar la aplicación cliente con los distintos workers.

¿Para qué sirve?

- Permite correr tareas en background, por ejemplo, responder una petición web lo antes posible y luego mediante consultas asíncronas actualizar el estado de la tarea para el usuario.
- Correr tareas luego de que finalizó una petición web, por ejemplo, envío de mails.
- Planificación periódica del trabajo.

http://docs.celeryproject.org

¿Brokers de Mensajería?

¿Qué es eso?

- Software que permite el pasaje de mensajes entre aplicaciones.
- Favorecen el desacoplamiento.
- Los brokers que soporta celery:
  - **RabbitMQ**
  - Redis
  - Amazon SQS
  - Zookeeper (Experimental)

RabbitMQ

- Es el broker por defecto en celery.
- Utiliza el protocolo AMQP (Advanced Message Queue Protocol)
- Permite envío de mensajes y monitoreo.

https://www.rabbitmq.com

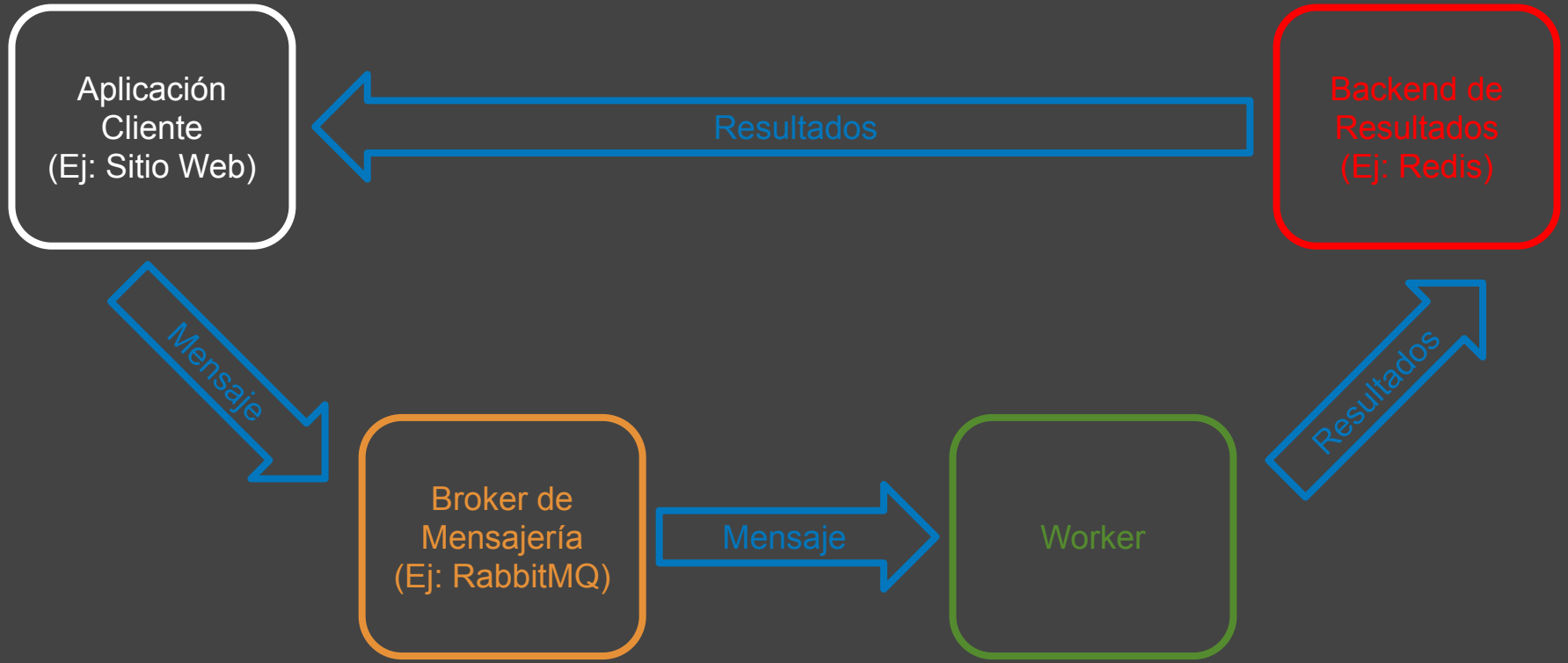
Entonces hasta acá...

Estados de las tareas

Si queremos seguir el estado de las tareas, celery debe guardar o enviar el estado de las mismas. Algunas de las opciones son:

- SQLAlchemy/Django ORM
- Memcached
- Redis
- RPC (RabbitMQ/AMQP)

Con el backend de resultados...



Funciones Importantes

Iniciando una tarea..

```
funcion.delay()
```

Función que se utiliza para poner en marcha una tarea. Devuelve una instancia de AsyncResult.

En caso de errores..

```
AsyncResult.get()
```

Devuelve la excepción lanzada desde el código de la tarea.

Funciones Importantes

Conociendo el estado de una tarea..

```
AsyncResult.state
```

Permite obtener el estado actual en que se encuentra una tarea.

Por defecto celery incluye estados predefinidos (PENDING, STARTED, SUCCESS, FAILURE, etc) pero se pueden definir estados propios.

Al Código!

Consideraciones Finales

- El software completo de Celery se encuentra distribuido bajo la licencia BSD.
- Actualmente se encuentra en la versión 4.3 (stable) y en la 3 dejó de tener soporte para Windows
- En producción, los workers trabajaran en background (demonio) utilizando systemd o supervisor.

¿Preguntas?
Muchisimas Gracias