

Projeto BeeHus

Bom Young Kim



Índice



0 1

Visão Geral

Infraestrutura

0 2



Por que?



0 3

Visão geral

A Beehus Tecnologia (Beehus Tech) é uma startup do setor de Wealth Tech, focada em desenvolver soluções tecnológicas para a consolidação e gestão de patrimônio de clientes no Brasil e no exterior. Possui como pilar fundamental, a tecnologia de dados. Utiliza múltiplas tecnologias para captura de dados, processamento diário, precificação de ativos e geração de relatórios personalizados para seus clientes. **O principal diferencial da Beehus é oferecer uma visão unificada e automática de todo o patrimônio.** A plataforma centraliza informações de diversos bancos e corretores, automatizando processos manuais para dar ao gestor financeiro uma visão clara e 360 graus dos ativos, facilitando a tomada de decisões estratégicas de forma eficiente e segura.



Infraestrutura

A combinação de Django, Celery, Celery Beat, e Redis forma uma arquitetura robusta para a construção de aplicações web modernas e escaláveis, que gerenciam tanto requisições síncronas quanto tarefas assíncronas complexas.

Django

Principal (backend). Gerencia a interface do usuário, a lógica de negócios síncrona, banco de dados (ORM), autenticação e a API que aciona as tarefas.

Redis

Serve como a "ponte de comunicação" de alta velocidade, armazenando as tarefas a serem executadas (fila) e os resultados das tarefas concluídas.

Celery

Executa tarefas demoradas (como extração de dados) em segundo plano, sem travar a resposta da aplicação web principal

Architecture Of Celery



Por que essa solução?

- 1 Escalabilidade: Permite que o site continue rápido, mesmo com muitas tarefas. Você pode adicionar mais "trabalhadores" (workers) para dar conta do recado sem sobrecarregar a página principal.
- 2 Rastreabilidade: Você consegue monitorar cada tarefa, sabendo se ela foi concluída ou se falhou. Ferramentas visuais mostram em tempo real o que está acontecendo nos bastidores.
- 3 Eficiência: Tira tarefas pesadas (extração de dados em vários pontos) da frente do usuário. O aplicativo mais leve e os recursos do servidor são usados de forma inteligente e otimizada.
- 4 Independência (com Docker): Cada parte do sistema (site, banco de dados, tarefas) funciona isolada em seu próprio "pacote". Isso facilita a instalação e evita problemas de compatibilidade.

The screenshot shows the Django administration interface for Periodic Tasks. On the left, there's a sidebar with links like Home, Periodic Tasks, Periodic tasks, and Add periodic task. The main area has a header "Django administration" and a sub-header "WELCOME, ADMIN". Below this, there are two tabs: "AUTHENTICATION AND AUTHORIZATION" and "PERIODIC TASKS".

AUTHENTICATION AND AUTHORIZATION:

- Groups: + Add
- Users: + Add

PERIODIC TASKS:

- Clocked: + Add
- Crontabs: + Add
- Intervals: + Add
- Periodic tasks:** + Add (This is the selected tab, highlighted in yellow)
 - Name: Print time every 30 seconds (with a note: "Short Description For This Task: []")
 - Task (registered): print_time
 - Description: Print the current time every 30 seconds
 - Schedule:
 - Interval Schedule: every 30 seconds (with a note: "Interval Schedule to run the task on. Set only one schedule type, leave the others null.")
 - Crontab Schedule: ----- (with a note: "Crontab Schedule to run the task on. Set only one schedule type, leave the others null.")
 - Solar Schedule: ----- (with a note: "Solar Schedule to run the task on. Set only one schedule type, leave the others null.")
- Solar events: + Add

Exemplo

Acompanhamento das tarefas e logs.

Agendamento de tarefas.

Diversos clientes e fontes, vai necessitar de uma solução para minimizar falhas de gestão de informação.

Obrigado!

