**Patch Notes - Sistema de Coleta de Dados FoxESS Cloud com MySQL**

**Versão Atual: 1.0.0**  
**Data de Lançamento: 03/09/2024**  
**Desenvolvedor Responsável: NEOGIER Energia**

**Descrição Geral**

Este sistema foi projetado para coletar dados de geração de energia de dispositivos vinculados à API FoxESS Cloud e salvá-los diretamente em um banco de dados MySQL. A solução é escalável, permitindo a coleta de dados de múltiplos clientes e dispositivos. Os dados históricos são armazenados em tabelas de histórico, enquanto as medições atuais são salvas em tabelas específicas para cada cliente.

**Novos Recursos**

* **Coleta de dados de múltiplos clientes:** O sistema foi configurado para gerenciar dados de dois clientes (Jacob Liebe e Alfonso Sleutjes) e armazenar as informações em tabelas específicas no banco de dados MySQL.
* **Armazenamento de dados históricos e atuais:** O sistema coleta dados de geração diária, salvando os dados de dias anteriores ao atual nas tabelas de histórico (history\_c1, history\_c2) e as medições do dia atual nas tabelas de cliente (client1, client2).
* **Tratamento de dados faltantes (missing data):** Se a API FoxESS retornar valores ausentes para qualquer dia, o sistema automaticamente insere um valor de 0 para evitar erros de processamento.
* **Compatibilidade com dispositivos FoxESS:** A API se conecta diretamente com os dispositivos identificados por seus números de série (device\_sn) e coleta informações de geração de energia com base nas chaves de API configuradas para cada cliente.

**Estrutura e Funções**

**Clientes**

A configuração dos clientes é armazenada em uma lista de dicionários, contendo as seguintes informações:

* api\_key: Chave de autenticação da API FoxESS Cloud para o cliente.
* device\_sn: Número de série do dispositivo.
* cod\_ponto\_medicao: Código de identificação do ponto de medição do cliente.
* cod\_agente: Código do agente, derivado da chave API ou do nome do cliente.
* history\_table: Nome da tabela onde os dados históricos serão armazenados.
* current\_table: Nome da tabela onde as medições do dia atual serão salvas.

**Função get\_and\_save\_generation\_data**

Esta função é responsável por:

* Configurar a API com as credenciais e o dispositivo do cliente.
* Coletar os dados de geração de energia mensal e diário para o dispositivo.
* Salvar as medições de energia no banco de dados MySQL.
  + **Histórico**: Dados de dias anteriores ao atual são armazenados na tabela de histórico.
  + **Dados atuais**: A medição do dia atual é armazenada na tabela específica do cliente.
* Tratar dados faltantes, inserindo 0 para dias sem dados.

**Função process\_all\_clients**

Esta função itera sobre todos os clientes configurados e chama a função get\_and\_save\_generation\_data para cada cliente, processando e armazenando os dados no banco de dados.

**Tratamento de Erros**

O sistema implementa tratamento de erros robusto para garantir o bom funcionamento e identificar possíveis falhas:

1. **Erros de conexão com o banco de dados:**
   * O sistema captura exceções relacionadas à conexão com o MySQL e falhas nas consultas SQL, como entradas duplicadas ou valores fora de alcance.
2. **Erros na API FoxESS:**
   * Falhas na configuração da API (como chaves de API inválidas) são detectadas e reportadas imediatamente.
   * Se a API retornar dados ausentes ou inválidos, o sistema insere valores padrão (0) para garantir a consistência dos dados.
3. **Registro de Progresso:**
   * O sistema imprime logs detalhados a cada inserção de dados, exibindo a quantidade de energia salva e a tabela onde os dados foram armazenados. Isso auxilia no acompanhamento em tempo real das operações do sistema.

**Exemplo de Execução do Sistema**

**Execução de Coleta de Dados**

Para executar o sistema e processar todos os clientes configurados, basta rodar o script principal:

**Tela de celular com aplicativo aberto

Descrição gerada automaticamente**

**Resultados Esperados**

O sistema coleta os dados de geração de energia de cada cliente desde 01/01/2024 até a data atual. A medição do dia atual é inserida na tabela específica do cliente, enquanto as medições de dias anteriores são inseridas nas tabelas de histórico correspondentes. Em caso de dados ausentes, o valor 0 é registrado automaticamente.

**Melhorias Futuras**

1. **Suporte a mais clientes:** Expansão do sistema para suportar a adição dinâmica de novos clientes sem modificar o código principal.
2. **Melhor desempenho:** Otimização do tempo de execução da coleta de dados, especialmente para clientes com longos períodos de medição.
3. **Notificações em tempo real:** Implementar notificações de sucesso ou falha via e-mail ou mensagens push.
4. **Interface de Monitoramento:** Adicionar uma interface gráfica para acompanhar o status das operações e geração de relatórios.

**Alterações Técnicas**

**Adições**

* **Estrutura de Clientes**: Lista de clientes com chaves de API e tabelas de destino específicas.
* **Tratamento de Erros**: Tratamento de exceções, incluindo falhas de conexão com a API e o banco de dados.
* **Logs de Progresso**: Mensagens detalhadas para acompanhar as inserções no banco de dados.

**Correções**

* **Correção de Dados Ausentes**: Se os dados diários de geração estiverem faltando na API, o sistema grava 0 no banco de dados.

**Considerações Finais**

Esta versão inicial do sistema de coleta e armazenamento de dados FoxESS Cloud está preparada para rodar de maneira automatizada, garantindo a integridade dos dados e escalabilidade para novos clientes.