

Estrutura de Armazenamento dos campos no banco de dados

No sistema, os campos são armazenados em dois locais diferentes, em um JSON com todos os campos juntos e na tabela da entidade.

1. JSON do Formulário

- O sistema gera um JSON contendo todos os dados de cada formulário, incluindo os campos numéricos, inteiros e de data e hora, e armazena esse JSON na tabela `tbl_formulario`.
- Esse JSON serve como um registro completo das informações enviadas pelo usuário, sendo útil também para auditoria de dados.

2. Tabela da Entidade - Campo VARCHAR

- Este campo armazena o valor recebido em "string", sem conversões. O tipo VARCHAR foi mantido por questões de compatibilidade com o sistema legado.

3. Tabela da Entidade - Campos tipados

- Campos booleanos, são gravados como smallint (0 ou 1), com exceção do campo "deletado", esse está gravado como true or false
- Campos com tipos específicos, como numéricos, inteiros e de data e hora, têm uma coluna adicional que armazena o valor tipado e convertido:
 - **Numéricos e Inteiros:** Os campos numéricos possuem o sufixo `__number__` e os inteiros seguem a mesma estrutura, facilitando operações e consultas. Exemplo: `campo__number__`. O tipo do campo é `numeric` no postgresql
 - **Data e Hora (Datetime):** Os campos de data e hora possuem o sufixo `__datetime__`. Exemplo: `campo__datetime__`. o tipo do campo é `date` para campos que são Data e `timestamp` para campos com date e hora

Observações Importantes

- **Exibição de Dados:** Na listagem, é sempre utilizado o campo com o sufixo tipado (`__number__` ou `__datetime__`), pois ele contém o valor devidamente convertido, garantindo precisão nas operações.
- **Máscara de Formatação no Formulário:** No formulário de entrada de dados, uma máscara em JavaScript é aplicada ao `campo`, formatando o valor conforme o padrão do JSON. Essa máscara pode causar discrepâncias se houver uma incompatibilidade entre os

formatos de entrada e o armazenamento no PostgreSQL.

Possíveis Problemas e Soluções

- **Diferença no Valor Armazenado:** Para valores numéricos, caso o valor enviado pela API contenha vírgulas (por exemplo, "3,31"), pode ocorrer uma interpretação incorreta durante a conversão para o formato numérico, transformando o valor `3,31` em `331`.
- **Recomendações:** Nas Apis, e regras, sempre manter o padrao no formato ingles 99999.99 com ponto. Sugere-se validar e formatar os dados enviados para garantir que o sistema receba sempre o formato correto. A padronização dos dados de entrada evita diferenças entre os valores no campo VARCHAR e nos campos tipados com sufixo.

Revision #1

Created 11 November 2024 18:16:00 by agilityflow

Updated 5 March 2025 12:33:44 by agilityflow