

# **Integração de Atividades com o Banco de Dados**

Autor: Hypertech

Data: 31/08/2015

**Sumário**

1. Contexto.....3

2. Problemas Identificados.....4

3. Funcionamento Esperado.....5

4. Próximos Passos para Correção.....6

## 1. Contexto

O módulo de gerenciamento de atividades foi implementado utilizando React com Context API. A ideia central é manter uma lista de atividades no estado global da aplicação e permitir operações de criação, edição, remoção e consulta, além de funcionalidades adicionais para confirmação de presença.

O ActivityContext foi estruturado da seguinte forma:

- activities: lista de atividades (IActivity[]) no estado.
- addActivity(activity: IActivity): adiciona uma nova atividade.
- editActivity(updated: IActivity): atualiza uma atividade existente.
- removeActivity(id: string): remove uma atividade pelo ID.
- getActivityById(id: string): retorna uma atividade específica pelo ID.
- confirmPresence(id: string) / cancelPresence(id: string): funções para gerenciar confirmações de presença.
- isConfirmed(id: string) / confirmedActivities: funções e estado para verificar quais atividades estão confirmadas.

## 2. Problemas Identificados na Integração com o Backend

1. Valores nulos em data e horários:

Inicialmente, ao tentar enviar os dados de dataAtividade, atividadeStartTime e atividadeFinishTime para o backend, os campos estavam chegando como null. Isso resultava em falhas de validação no servidor, impedindo a criação correta das atividades.

2. Erro de tamanho 0:

Após ajustes iniciais, um novo erro foi identificado: o backend retornava um erro relacionado a tamanho 0. Isso indica que os dados enviados pelo frontend podem estar chegando vazios ou com arrays/strings sem conteúdo, possivelmente devido à forma como o estado estava sendo atualizado ou como os dados estavam sendo convertidos antes do envio.

## 3. Funcionamento Esperado

- As atividades são mantidas em memória no frontend utilizando useState.
- Cada operação (adicionar, remover, editar) altera o estado local, mas atualmente não há integração direta com a API do backend para persistência.
- As confirmações de presença também são mantidas localmente e não estão sincronizadas com o backend.
- As funções do contexto assumem que os IDs das atividades são do tipo string. Caso o backend utilize outro tipo (por exemplo, number), isso pode gerar inconsistências.

## 4. Próximos Passos para Correção

1. Verificar o tipo de dados esperado pelo backend:

- Garantir que id, dataAtividade, atividadeStartTime e atividadeFinishTime estejam no formato correto antes de enviar.
- Converter datas e horários para o padrão aceito pelo backend (por exemplo, ISO 8601).

2. Implementar a comunicação com a API:

- Substituir ou complementar o estado local com chamadas HTTP (fetch ou axios) para persistir alterações no

banco.

- Tratar respostas e erros do servidor, garantindo que dados inválidos não sejam mantidos no estado.

### 3. Sincronizar confirmações de presença:

- Criar endpoints no backend ou adaptar os existentes para que a confirmação/cancelamento de presença seja persistida.

### 4. Validação de dados no frontend:

- Adicionar validações antes de enviar para o backend, evitando que campos vazios ou nulos sejam enviados.

### 5. Testes:

- Testar cada operação individualmente (criar, editar, remover, confirmar presença) verificando o estado do frontend e o conteúdo persistido no banco.