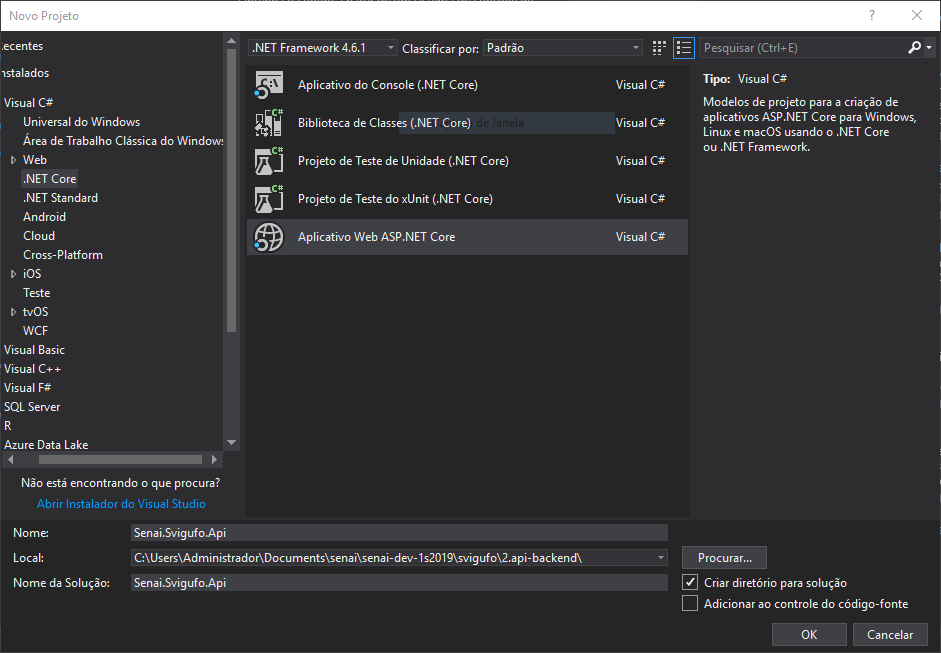
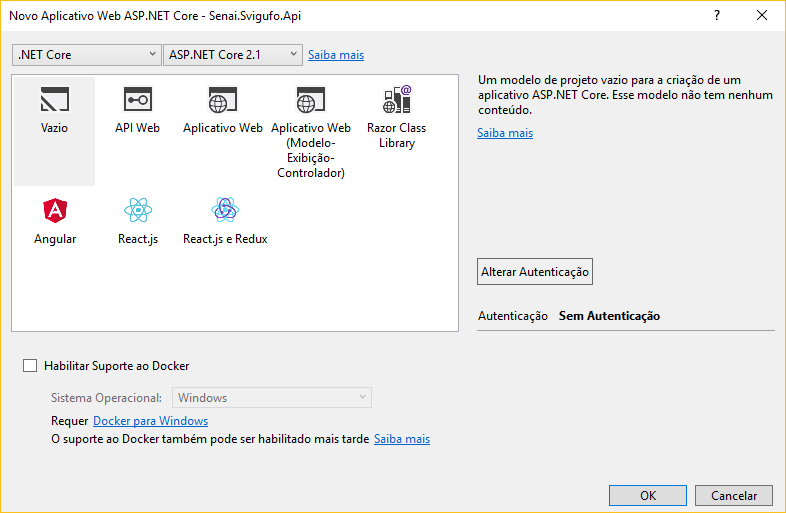
1. API
2. Startup – configurar o MVC e a versão que eu estou utilizando
3. TipoEventoController
4. TipoEventoDomain
5. ITipoEventoRepository
6. TipoEventoRepository
7. InstituicaoController
8. InstituicaoDomain
9. IInstituicaoRepository
10. InstituicaoRepository

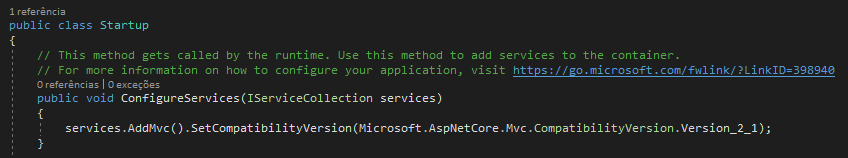
Explicar todo o contexto de API com imagens e apresentações.

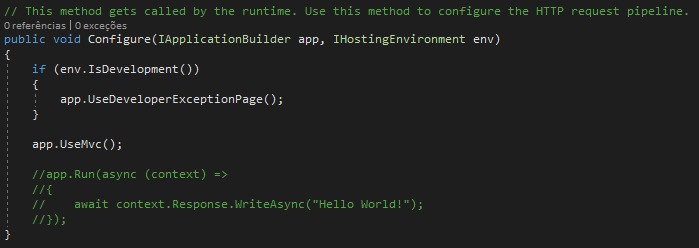
Explicar o que é JSON, API, vantagens, desvantagens

Somente após explicar todo o contexto, mostrar um projeto sendo criado

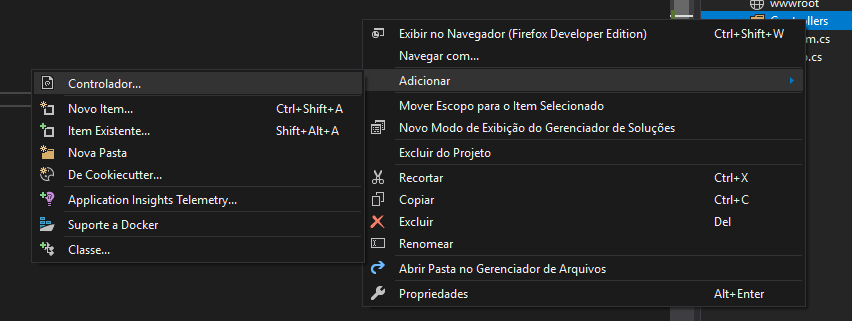






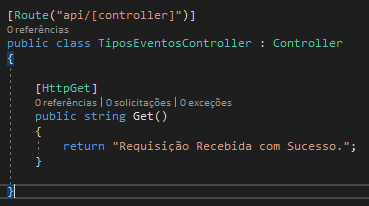


Nova pasta Controllers



QUANDO CRIAR O CONTROLLER, LEMBRAR DE FALAR SOBRE O CONTROLLERBASE – QUE ATUA SEM O SUPORTE A VIEW, POIS NÃO IREMOS PRECISAR DELA

TiposEventosController

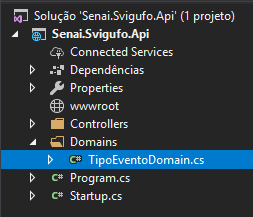


Realizar requisição

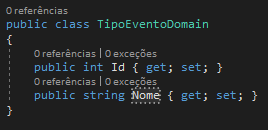
Mostrar na web – Navegador, a informação recebida – Não iniciar com o postman ainda

Criar nova pasta – Domains

Criar nova classe TipoEventoDomain



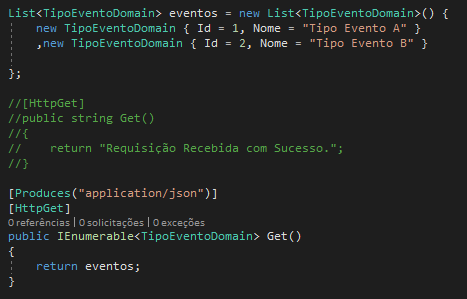
Incluir as propriedades



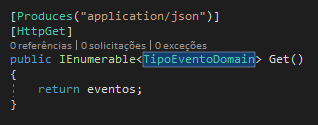
<https://stackoverflow.com/questions/23196931/returning-ihttpactionresult-vs-ienumerableitem-vs-iqueryableitem>

Diferença entre IEnumerable e IActionResult

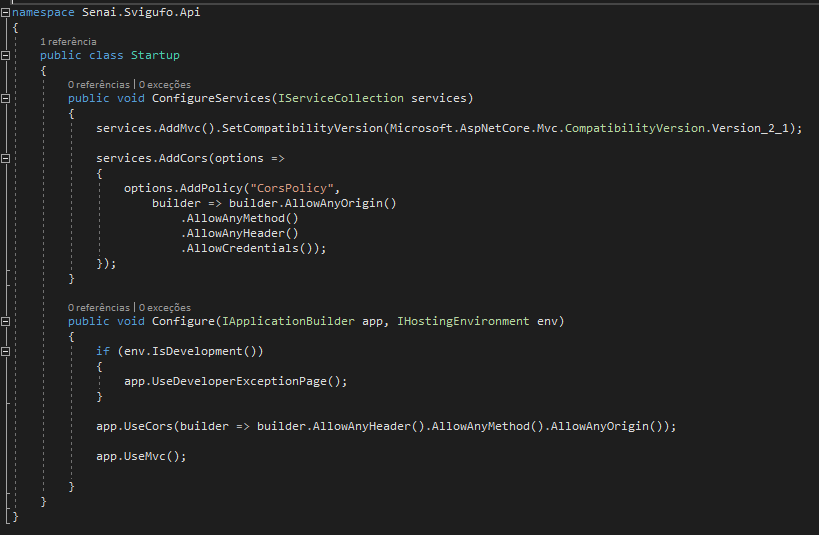
Realizar um retorno simples com uma lista local



Realizar a requisição



LEMBRAR DE INCLUIR O CORS NA NOSSA APLICAÇÃO PARA MOSTRAR O FRONT



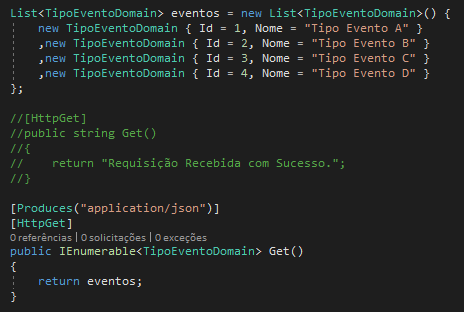
Mostrar o front-end fazendo a requisição para o back-end

E mostrar que sua implementação não é conhecida

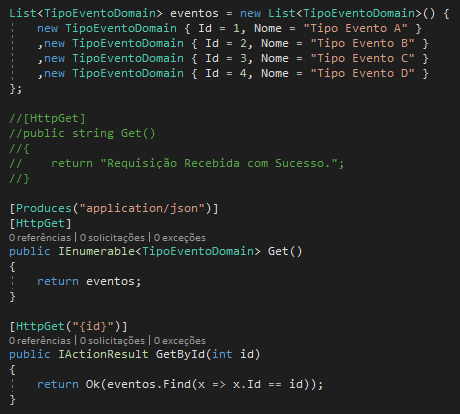
Eu nem mesmo sei/conheço aonde esses dados estão sendo armazenados

Adicionar mais dois eventos e mostrar o resultado

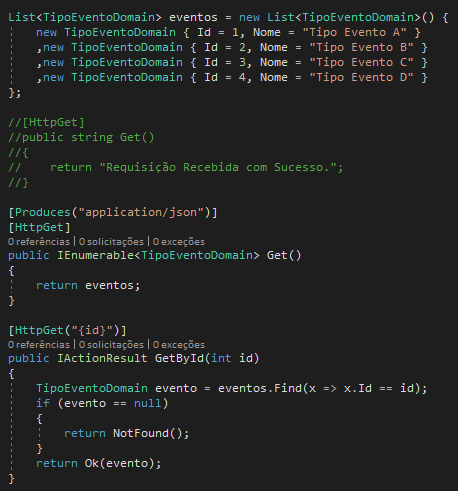
E também mostrar a interação no front-end

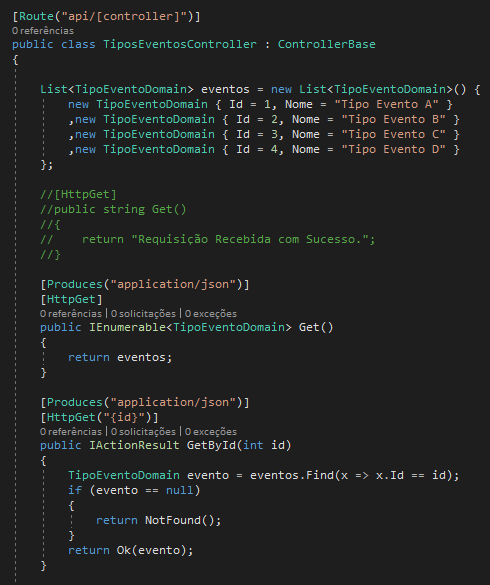


Buscando somente um



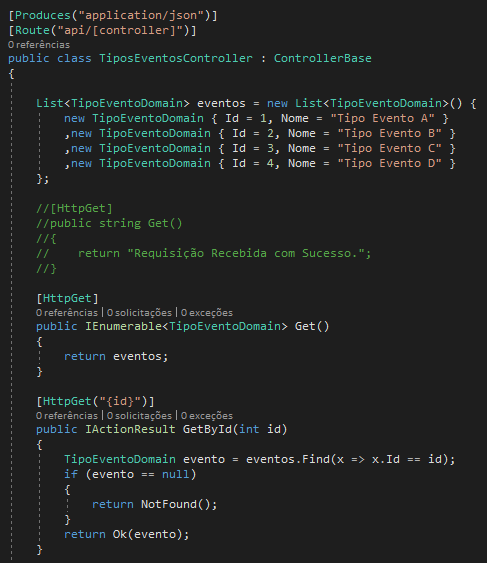
Melhorar realizando um if para ver se o evento foi encontrado





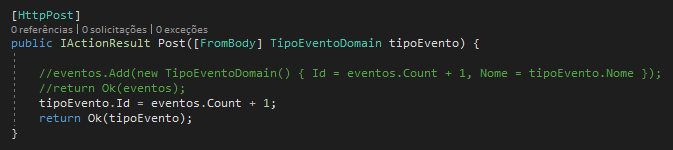
A saída também será produces

Podendo assim, colocar no controller diretamente

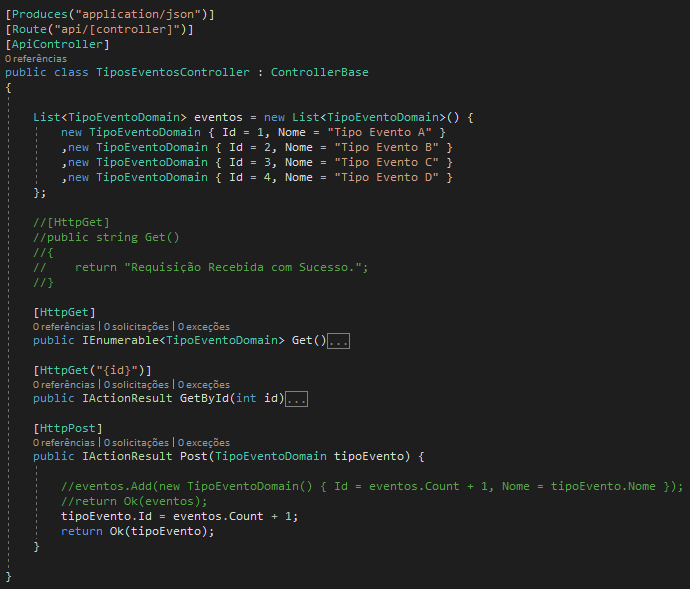


Trabalhando com POST

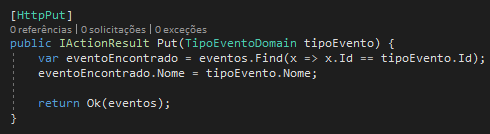
MOSTRAR SOMENTE AQUI O POSTMAN



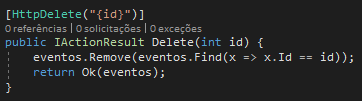
Explicar a anotação [ApiController]



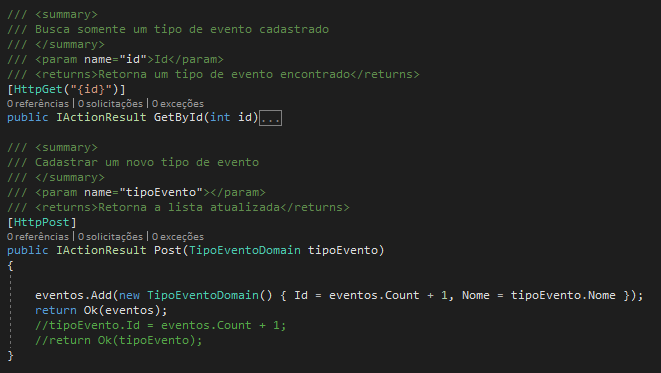
Mostrar o PUT



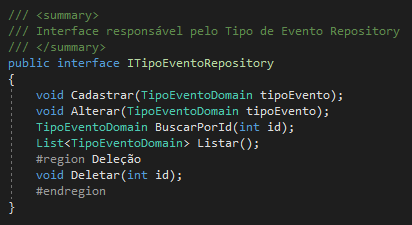
DELETE



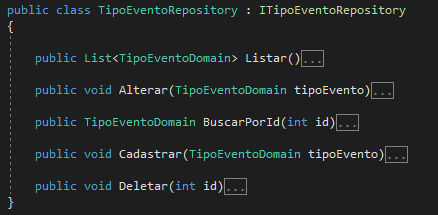
INCLUIR COMENTÁRIOS NO CÓDIGO



Criar uma nova interface

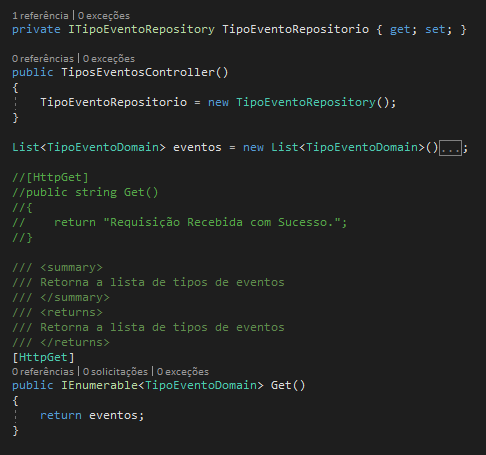


Criar um novo repositório

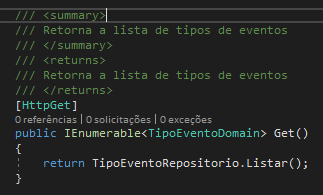


No controller, vamos alterar para ao invés de trabalhar com a lista local, chamarmos do nosso banco de dados.

Cria o construtor no controller

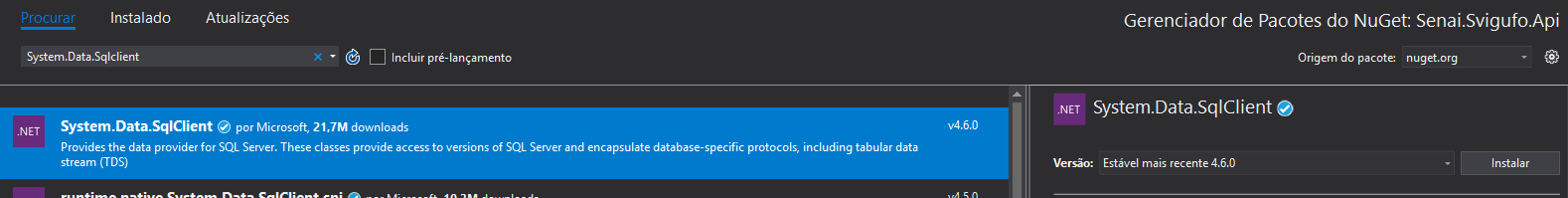


Todo a sua implementação continua a mesma



Porém, os dados virão agora do banco de dados

Botão direito -> gerenciar pacotes do nuget

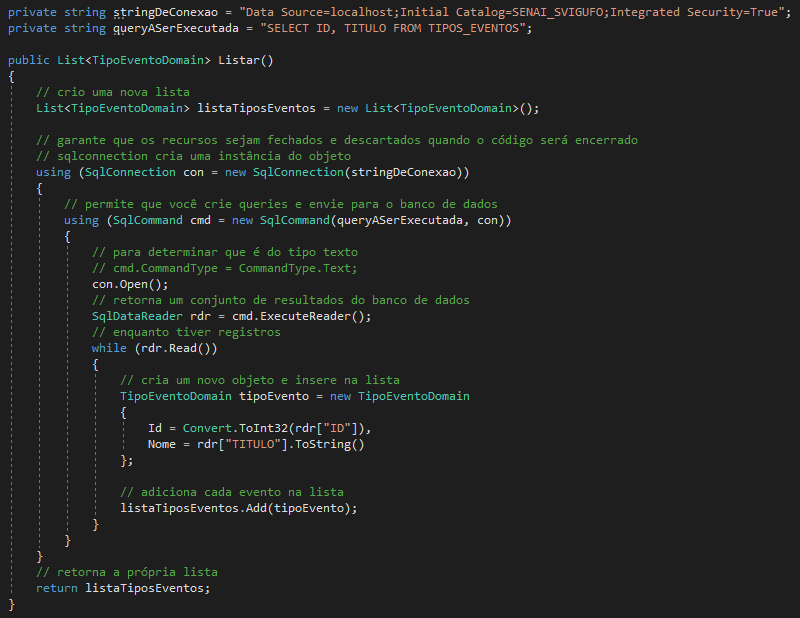


ADO.NET

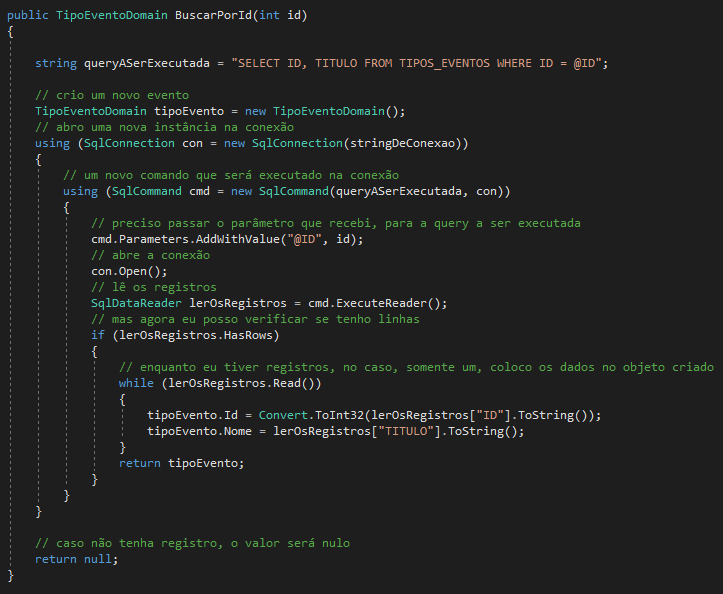
<https://docs.microsoft.com/pt-br/dotnet/framework/data/adonet/>

Implementação do Repositório

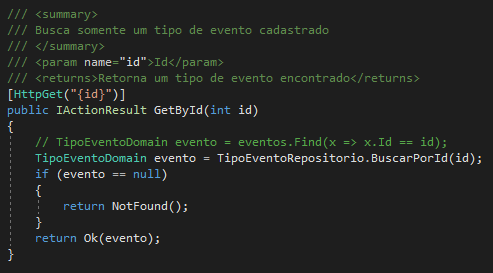
LISTAGEM



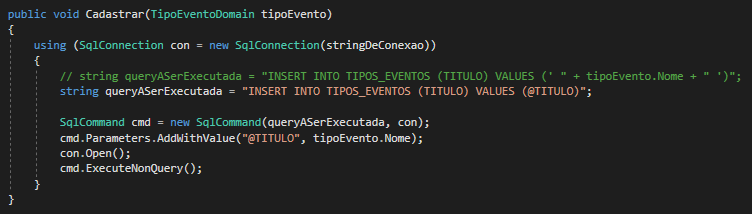
BUSCAR SOMENTE UM



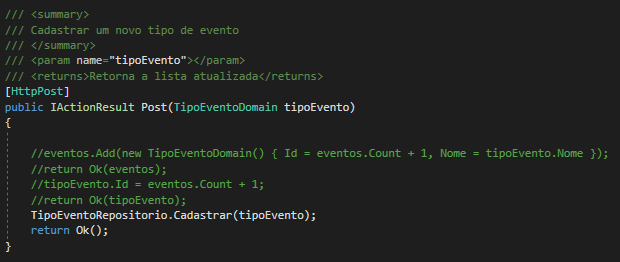
Mudo a implementação no controller, para realizar a busca agora no repositório



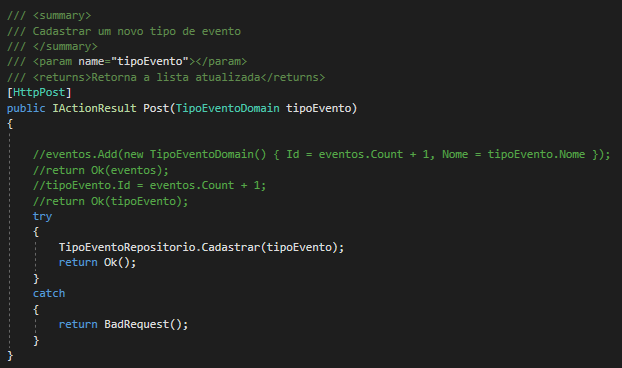
Alterar o repositório para cadastrar um novo tipo de evento



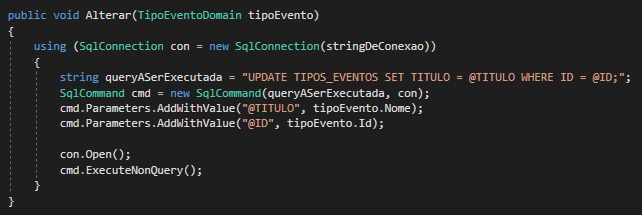
Alterar o controller



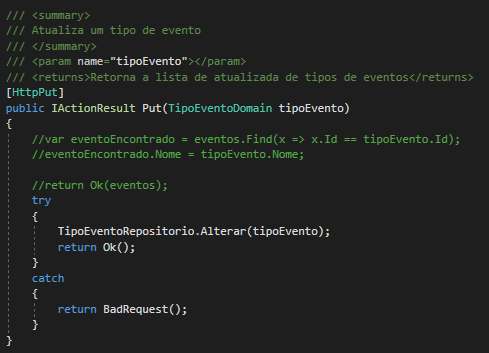
Realizar uma verificação na inserção para não travar a inserção



Alterar a implementação no repositório para atualizar



Ajustar a implementação no controller



Alterar o repositório para deletar um registro

