

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula:

- Crie a interface `EntidadeFactory`
 - Nela, defina o método público `criarEntidade`, que recebe uma string em JSON por parâmetro
- Crie a classe `EspecialidadeFactory`, que irá implementar a interface `EntidadeFactory`
 - Nesta classe, implemente o método `criarEntidade`, que irá instanciar, atribuir os dados e retornar uma `Especialidade`, a partir de uma string em JSON recebida por parâmetro
- Na classe `MedicoFactory`, implemente a interface `EntidadeFactory` e renomeie o método `criarMedico` para `criarEntidade`
- Crie a classe abstrata `BaseController`
 - Essa classe herdará de `AbstractController`
 - No seu construtor, injete um `ObjectRepository`, um `EntityManagerInterface` e uma `EntidadeFactory`, e atribua-os a atributos protegidos da classe
 - Implemente os métodos `novo`, `remove`, `buscarTodos`, `buscarUm` e `atualiza`, baseados nos métodos dos *controllers*
 - Implemente também o método abstrato `atualizarEntidadeExistente`, que recebe por parâmetro a entidade existente e a entidade enviada
- Na classe `EspecialidadesController`, herde da classe `BaseController`
 - Remova todos os seus atributos e suas atribuições dentro do construtor
 - Injete uma `EspecialidadeFactory` no seu construtor
 - Dentro do seu construtor, chame o construtor da classe mãe, passando os parâmetros necessários
 - Remova os métodos `nova`, `remove`, `buscarTodas`, `buscarUma` e `atualiza`
 - Implemente o método `atualizarEntidadeExistente`, e através da anotação `@param`, tipe os seus parâmetros
 - Dentro do método `atualizarEntidadeExistente`, atualize a `descricao` da `Especialidade` existente baseado na `descricao` da `Especialidade` enviada
- Na classe `MedicosController`, herde da classe `BaseController`
 - Remova todos os seus atributos e suas atribuições dentro do construtor
 - Aonde `medicosRepository` era utilizado, substitua por `repository`
 - Aonde `medicoFactory` era utilizado, substitua por `factory`
 - Dentro do seu construtor, chame o construtor da classe mãe, passando os parâmetros necessários
 - Remova os métodos `novo`, `remove`, `buscarTodos`, `buscarUm` e `atualiza`
 - Implemente o método `atualizarEntidadeExistente`, e através da anotação `@param`, tipe os seus parâmetros
 - Dentro do método `atualizarEntidadeExistente`, atualize os atributos do `Medico` existente baseado nos atributos do `Medico` enviado
- Na pasta do seu projeto, dentro de **config**, no arquivo **routes.yaml**, mapeie as rotas da aplicação, que antes eram feitas nas anotações `@Route` dos métodos

Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.

