

Consolidando o seu conhecimento

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) No construtor do `BaseController`, receba (e inicialize como atributo da classe) um novo parâmetro chamado `$cache`, do tipo `Psr\Cache\CacheItemPoolInterface`.

2) Nos `controllers` específicos, receba também este parâmetro e passe-o para o contrutor da classe base.

3) No método `novo`, logo após salvar a entidade com o Doctrine, salve a entidade no `cache`, com o seguinte código:

```
// ...
$cacheItem = $this->cache->getItem($this->cachePrefix() . $entity->getId());
$cacheItem->set($entity);
$this->cache->save($cacheItem);
// ...
```

4) Adicione também, na classe `BaseController`, o método abstrato `abstract public function cachePrefix(): string;`.

5) Implemente estes métodos nos `controllers` especializados;

- Em `EspecialidadeController`, faça este novo método retornar a `string especialidade_`
- Em `MedicoController`, faça este novo método retornar a `string medico_`

6) No método `buscarUm`, da classe `BaseController`, tente buscar a entidade do `cache`:

```
// ...
$entity = $this->cache->hasItem($this->cachePrefix() . $id)
? $this->cache->getItem($this->cachePrefix() . $id)->get()
: $this->repository->find($id);
// ...
```

7) No final do método `deletar`, adicione a seguinte linha: `$this->cache->deleteItem($this->cachePrefix() . $id);`.

8) No final do método `atualizar` (antes de retornar), adicione o seguinte conteúdo:

```
// ...
$cacheItem = $this->cache->getItem($this->cachePrefix() . $id);
$cacheItem->set($existingEntity);
$this->cache->save($cacheItem);
// ...
```