

我们希望在测试代码中能够对数据库进行查询, 然后对查询到的数据进行验证, 最后实现自动化的测试。我们新建一个测试方法, 名称叫做`testEntityManagerQuery()`, 返回值是`void`, phpunit每个测试方法都是独立的, 所以我們也需要在新的测试方法中启动内核, 然后获取到对应的服务类对象。复制代码, 我们在使用命令行创建Entity类的时候, 对应的在Repository目录中也创建了对应的Repository类。

我们打开PostRepository类, Repository类上面添加了四个魔术方法, 分别是`find()`, `findOneBy()`, `findAll()`, `findBy()`。我们根据函数名称来进行一下理解:

`find()` 第一个参数是`$id`它是进行单项查询的, 很明显`find()`方法是根据传入的`$id`来获取对应的对象数据。

`findOneBy()`是根据查询条件获取一条对象数据。

`findAll()`是查询所有的数据。

`findBy()`是根据查询条件来获取所有符合条件的数据。

我们将在测试代码中获取到PostRepository这个服务类对象, 修改测试代码, 我们创建变量`$postRepo`, 等于`static::getContainer()`, 使用容器的方法来获取到PostRepository这个服务类对象。我们查询PostRepository在容器中的ID。

输入命令行`symfony console debug:container`, 参数输入PostRepository, 我们查询到了PostRepository服务类。它的ID就是类名。我们使用容器来获取到这个服务类对象, 我们下个断言。我们断言`$postRepo`变量就是PostRepository的一个实例。执行单元测试, 测试通过了, 我们上节课说在src目录下, 服务类都是私有的, 除非你设置为公开的, 我们没有设置PostRepository类为公开状态。我们依然使用容器获取到了服务对象, 这是什么原因呢?

我们继续查看容器中这个类, 我们看到容器中PostRepository这个服务类, 添加了一个`tags`, `tags`名称是`doctrine.repository.service`。我们再查看一下我们自己手写的PostFactory这个服务类, 我们自己手写的服务类没有`tags`。

我们可以查看一下doctrine源码, 在doctrine-bundle这个文件夹下, 依赖注入文件夹下, 有个Compiler文件夹, Compiler文件夹下有个`ServiceRepositoryCompilerPass`。我们打开这个类, CompilerPass是在容器进行编译的时候, 对容器中的类进行一些处理, 在`process()`方法中, 我们看到第27行, 容器会查找所有的`doctrine.repository.service`标签的服务类, 然后在第33行, `ServiceLocatorTagPass`的`register()`方法将对应的服务类ID注册到了容器中。

按着command键点击`register()`方法, 在`register()`方法中, 有个`ServiceLocator`这个服务类, `$locator`服务类对象会将我们的Repository服务ID绑定到`$locator`对象中, 最后容器会将ServiceLocator服务类注册到容器中。作为初学者, 你可能不明白`ServiceLocator`的这个类, 没关系。在后面我们会再次讲解到ServiceLocator, 这里你只需要对ServiceLocator有印象就可以了。

在CompilerPass这个方法中, 大概意思就是说当我们内核在进行启动的时候, 容器会编译所有的服务类, 容器会查找所有的`doctrine.repository.service`标签的服务类对象。将Repository服务类注册到ServiceLocator中, 当我们使用容器获取Repository类对象时, 它会从ServiceLocator中获取对应的Repository对象, 在后面课程中我们会再次见到ServiceLocator的这个服务类。

现在我们已经获取到了PostRepository这个对象。下面我们就可以对数据库进行查询了, 我们使用PostRepository的`findAll()`方法, 查询数据库中所有的Post对象, 现在我们断言`$posts`变量, 它包含了4个Post对象。

执行测试, 测试出错了, 数据库中有12条数据, 但是我们期待的是4条, 这是为什么呢? 当我们启动测试时, 所有的测试方法都会执行, 所以每次我们测试时, `testEntityManager()`这个方法也会执行, 每次执行都会向数据库中插入4条数据, 所以数据库表中Post的数据远远大于4条。

在下一节，我们将讲解在每次进行测试之前清空测试数据库。