继续修改测试代码，我们已经使用findAll()方法获取到数据库中所有的数据，下面我们来使用findOneBy()方法来获取数据库中符合查询条件的第一条数据。查询条件它是一个数组，其中数组的key，就对应了Post对象的属性，我们查询title等于Post title 01的数据，它的返回值是Post对象或者为空值。

定义一个变量，我们期待$findOneByPost这个变量和$post1是相同的，下个断言，执行测试，测试通过了。

我们再来测试find()方法，find()方法第一个参数是ID，我们通过刚刚获取到的$findOneByPost对象的ID查询一个Post对象，我们期待$findPost和$findOneByPost是相同的。执行测试，代码测试仍然通过了。

最后我们来测试findBy()方法，在执行测试代码时，插入的4条数据都是草稿状态，现在我们来修改一下插入的数据，我们让第2条数据修改为已发布状态，现在我们插入数据库中的数据，第2条是已发布状态，其他3条仍然是草稿状态现在我们使用findBy()方法来获取数据库中的所有草稿状态的文章，查询条件的key，仍然是Post对象的属性，查询它为draft，我们断言$findByPosts的对象一共有3条。

```
class EntityManagerTest extends KernelTestCase
{
    // ...

    public function testEntityManager(): void
    {
//        $kernel = self::bootKernel();

        $this->assertSame('test', self::$kernel->getEnvironment());
//        $entitymanager = static::getContainer()-
>get('doctrine.orm.entity_manager');
        $this->assertInstanceOf(EntityManagerInterface::class, $this-
>entitymanager);
        //$myCustomService = self::$container->get(CustomService::class);

        $factory = static::getContainer()->get(PostFactory::class);
        $this->assertInstanceOf(PostFactory::class, $factory);

        $post1 = $factory->create('Post title 01', 'Post Body 01');
        $post2 = $factory->create('Post title 02', 'Post Body 02', null,
'published');
        $post3 = $factory->create('Post title 03', 'Post Body 03');
        $post4 = $factory->create('Post title 04', 'Post Body 04');
        $this->entitymanager->persist($post1);
        $this->entitymanager->persist($post2);
        $this->entitymanager->persist($post3);
        $this->entitymanager->persist($post4);

        $this->entitymanager->flush();

        $postRepo = static::getContainer()->get(PostRepository::class);

        $this->assertInstanceOf(PostRepository::class, $postRepo);

        $posts = $postRepo->findAll();
        $this->assertCount(4, $posts);
```

```
        $findOneByPost = $postRepo->findOneBy(['title' => 'Post title
01']);
        $this->assertSame($post1, $findOneByPost);

        $findPost = $postRepo->find($findOneByPost->getId());
        $this->assertSame($findOneByPost, $findPost);

        $findByPosts = $postRepo->findBy(['status' => 'draft']);
        $this->assertCount(3, $findByPosts);

    }

    // ...
}
```

再次执行测试，测试仍然通过了。

PostRepository提供的4个魔术方法，可以很快速的让我们来获取到数据库中的数据，但是如果执行模糊查询时，这四个方法就不起作用了，我们来看PostRepository的代码，第26行有个`findByExampleField()`方法，在方法中它创建了一个QueryBuilder对象，通过QueryBuilder对象来获取数据。

在下一节课，我们将使用QueryBuilder对象来创建模糊查询和基本查询的一些示例。