

我们想在测试环境下进行数据库的管理, Symfony的orm组件, 为我们提供了数据库操作的命令行。打开底部控制台, 输入`symfony console list doctrine`来查看可用的命令, `doctrine:database:create`命令是用来创建数据库的。

我们需要在测试环境下创建一个数据库, 如果你使用`php bin/console`创建命令, 我们需要指定`--env`选项为`test`环境, 这样这个命令就运行在了测试环境下。我们回车, 测试环境下的数据库和我们开发环境下的数据库是隔离的, 所以它自动在数据库名后面添加了下划线`test`。

我们删除这个数据库, 如果你使用`symfony console`, 来创建数据库, 我们需要在命令行前添加环境变量`APP_ENV=test`。再次执行, 测试环境下的数据库就创建成功了。

在上节课我们已经生成了一个数据迁移, 我们需要在测试环境下执行这个数据库迁移, 设置环境变量, 输入`APP_ENV=test symfony console doctrine:migrations:migrate`来执行这个数据库迁移。回车。啊, 我们单词拼写错了, 修改一下, 回车, 直接yes。数据库在测试环境下就迁移到了最新的版本。

我们查看数据库, 刷新, 我们的测试数据库中也出现了三张表, 测试环境下的数据库操作, 我们需要使用到Symfony的orm服务类, 这就需要将Symfony的服务类集成到我们的测试代码中, 这就是集成测试。

在下一节我们将讲解集成测试。