

订阅DeepL Pro以编辑此演示文稿。  
访问[www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document)，了解更多信息。

软件架构和设计模式 2021年秋季

实验3讲义。ORM魔法和服务层

**到期：2021年12月29日**

# 目标

理解*依赖性反转*。

使用SQLAlchemy的ORM（对象关系映射器）将一个类映射到一个数据库表。

实现一个服务层，让用户阅读文章。

练习测试驱动开发（TDD）。

# 任务描述

在这个实验室中，你将了解如何通过遵循*依赖性倒置*原则来保持领域模型的*纯洁性*--让基础设施依赖于领域模型，而不是反过来。

同时，你要在services.py中为EnglishPal实现一个服务层，它提供了一个名为read的核心服务。这个服务将为用户选择一个合适的文章来阅读。函数read接收以下四个参数作为输入，如果用户被成功分配到一篇文章阅读，则返回一个文章ID。

user：一个用户对象。在model.py中定义了User这个类。User 有一个重要的方法，叫做 read article。 user repo: 一个 UserRepository 对象。类 UserRepository 定义在 repository.py 中。 article repo: 一个 ArticleRepository 对象。ArticleRepository类定义在repository.py中。 session：一个SQLAlchemy会话对象。

如果用户没有正确的用户名或密码，函数read(user, user repo, article repo, session)会引发未知用户（UnknownUser）异常；如果文章库（即文章库）中没有文章的难度与用户的词汇量相符，则会引发NoArticleMatched异常。我们说一篇文章的难度等级*La*与用户的词汇等级*L*相匹配*u*，如果*La >Lu*。如果有多篇文章满足*L a >Lu*，那么就选择*L*最小的*a* 一篇。

一篇文章的难度记录在数据库表articles中的level字段中。一个用户的词汇量被定义为该用户的新词列表（记录在数据库表newwords中）中最难的前*n个*词的平均值，其中*n*是3或表newwords中属于该用户的新词数，以较小者为准。

为了简单起见，我们只考虑以下字典中的单词，其中的数值代表这些单词的难度。

d = {'starbucks':5, 'luckin':4, 'secondcup':4, 'costa':3, 'timhortons':3, 'frappuccino':6}。

下载以下启动代码即可开始使用。

orm.py model.py repository.py

1 of 2

软件架构和设计模式 **实验3讲义。ORM的魅力和服务层**

2021年秋季 **到期：2021年12月29日**

services.py conftest.py 测试 services.py

你必须完成orm.py和services.py，使运行以下命令可以使测试services.py中定义的所有五个测试用例通过： pytest -v -s test services.py。你不能修改 test services.py 中的任何内容。

提示：conftest.py可以帮助pytest在运行测试服务.py时进行设置。你可能想了解[什么是夹具以及如何使用它。](https://www.tutorialspoint.com/pytest/pytest_fixtures.htm)

# 要求

在一个小组内做实验。该小组必须与你的课程项目小组相同。

你的工作是完成下载的python源代码orm.py和services.py，使运行命令pytest -v -s test services.py不显示错误。

不要在services.py或model.py中写任何原始SQL语句。

在到期数据前提交一份用 "[阅读文件 "](https://readthedocs.org/)编写的实验报告[。](https://readthedocs.org/)你的实验报告必须遵循《[如何撰写计算机科学实验报告》](https://thehackpost.com/a-brief-guide-how-to-write-a-computer-science-lab-report.html)中描述的结构[。](https://thehackpost.com/a-brief-guide-how-to-write-a-computer-science-lab-report.html)

你的实验报告必须包含以下内容。

1. 修改后的orm.py。
2. 修改后的services.py。
3. 详细解释为什么你的修改能发挥作用。
4. 对这个问题的回答。你在services.py中读取的函数是否遵循单一责任原则（SRP）的原则？为什么或为什么不呢？

通过[LRR](http://118.25.96.118/nor/)提交你的实验报告（PDF格式）[。](http://118.25.96.118/nor/)不要忘记包括你的小组信息。不要漏掉任何小组成员的名字。

2 of 2