

Lab 3 - The Service Layer

Author: 占健豪, 王彦超, 陈致远, 汤佳伟

Date: 2021/6/7

Location: 22-206

Introduction

This lab is a follow-up to Lab 2. In this lab, we are going to implement a service layer in `services.py` for EnglishPal, which provides a core service called `read`. This service would choose a suitable article for a user to read. The function `read` takes as input the following four arguments and returns an article ID if the user has been successfully assigned with an article to read.

The function `read(user, user repo, article repo, session)` raises an `UnknownUser` exception if user does not have a correct user name or a correct password, or raises a `NoArticleMatched` exception if no article in the article repository, i.e., `article repo`, has a difficulty level matching the user's vocabulary level. We say that an article's difficulty level, L_a , matches a user's vocabulary level, L_u , if $L_a > L_u$. If more than one article satisfies $L_a > L_u$, then the one with the smallest L_a is chosen.

- `user`: a `User` object. The class `User` is defined in `model.py`.
- `user repo`: a `UserRepository` object. The class `UserRepository` is defined in `repository.py`.
- `article repo`: an `ArticleRepository` object. The class `ArticleRepository` is defined in `repository.py`.
- `session`: an `SQLAlchemy` session object.

Materials and Methods

Work flow

1. Review and analyze the requirements in `lab3.pdf`.
2. Learn about the relative knowledges with service layer and data layer.
3. Start with code.
4. Search for the coding techniques required online.
5. Finish the coding process.
6. Summarize and Write the document.

Source Codes

Modified `service.py`: // Modified part with code highlight

```

1  # Software Architecture and Design Patterns -- Lab 3 starter code
2  # An implementation of the Service Layer
3  # Copyright (C) 2021 Hui Lan
4
5  import repository
6  import model
7
8  # word and its difficulty level
9  WORD_DIFFICULTY_LEVEL = {'starbucks':5, 'luckin':4, 'secondcup':4,
10 'costa':3, 'timhortons':3, 'frappuccino':6}
11
12
13 class UnknownUser(Exception):
14     pass
15
16
17 class NoArticleMatched(Exception):
18     pass
19
20
21 def read(user, user_repo, article_repo, session):
22
23     dbuser = user_repo.get(user.username)
24     if dbuser is None or dbuser.password != user.password:
25         raise UnknownUser(user)
26
27     wordList = session.query(model.NewWord).filter(model.NewWord.username
28 == user.username).all()
29     difficulty_list = []
30     for w in wordList:
31         difficulty_list.append(WORD_DIFFICULTY_LEVEL.get(w.word))
32     difficulty_list.sort(reverse=True)
33     lu = 0
34     count = 0
35     for i in difficulty_list:
36         if count>=3:
37             break
38         lu += i
39         count+=1
40     lu /= count
41
42     qualified_articles =
43 session.query(model.Article).filter(model.Article.level >
44 lu).order_by(model.Article.level).all()
45     if not qualified_articles:
46         raise NoArticleMatched
47     else:
48         wanted_article = qualified_articles[0]
49         ar = article_repo.get(wanted_article.article_id)
50         session.add(ar)
51         dbuser.read_article(ar)
52         session.commit()
53         return wanted_article.article_id

```

Results

```
Windows PowerShell
PS C:\Users\CZY_C\Desktop\learn\大三\软件\EnglishPal\app> pytest -s -v test_services.py
===== test session starts =====
platform win32 -- Python 3.8.8, pytest-6.2.3, py-1.10.0, pluggy-0.13.1 -- C:\Users\CZY_C\anaconda3\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\CZY_C\Desktop\learn\大三\软件\EnglishPal\app
plugins: anyio-2.2.0
collected 5 items

test_services.py::test_read_article_level4 PASSED
test_services.py::test_read_article_level5 PASSED
test_services.py::test_read_article_level6 PASSED
test_services.py::test_user_with_wrong_password PASSED
test_services.py::test_user_with_wrong_username PASSED
```