

Iteration 8 – Beleg für automatische Tests

Daten

Testende Person	Dennis Weinberger
Ausführungszeitpunkt	15. März
Getestetes Feature	Assessment
Ergebnis	Alle Tests bestanden
Coverage	48%

Testdurchlauf

```
(tsam_venv) C:\Users\denni\PycharmProjects\tsam\backend>pytest --cov --cov-config=.coveragerc
===== test session starts =====
platform win32 -- Python 3.7.2, pytest-4.1.2, py-1.9.0, pluggy-0.13.1
rootdir: C:\Users\denni\PycharmProjects\tsam\backend
plugins: cov-2.11.1
collected 116 items

testing\test_assignment.py .... [ 3%]
testing\test_auth.py ..... [ 7%]
testing\test_candidate_chat.py ..... [ 12%]
testing\test_candidate_skill_grade.py ..... [ 16%]
testing\test_essential_skill.py ..... [ 20%]
testing\test_file.py ..... [ 25%]
testing\test_grade_test_function.py ..... [ 30%]
testing\test_grade_scale_function.py ..... [ 35%]
testing\test_interview_question.py ..... [ 40%]
testing\test_job.py ..... [ 45%]
testing\test_marker.py ..... [ 49%]
testing\test_observation.py ..... [ 54%]
testing\test_role.py ..... [ 59%]
testing\test_source.py ..... [ 64%]
testing\test_transaction.py ..... [ 69%]
testing\test_user.py ..... [ 74%]
testing\test_weight_scale_function.py ..... [ 79%]

----- coverage: platform win32, python 3.7.2-final-0 -----
Name                               Stmts   Miss  Cover
-----
controllers\assessment.py           73      43    41%
models\assessment.py                10       0   100%
TOTAL                               83      43    48%

===== 116 passed in 41.36s =====
```

(die relevante Datei heißt fälschlicherweise „test_assignment.py“, dies wurde noch vor der Endabgabe zu „test_assessment.py“ geändert)

Konsequenzen

Alle Tests liefen erfolgreich durch. Das Feature kann in das Produkt eingebaut werden.

Bemerkungen

Die Code Coverage ist mit 48% etwas niedrig ausgefallen, da wir uns dem Ende des Projektes zuneigten und keine Zeit mehr hatten, ausführliche automatische Tests einzubauen. Dennoch haben wir uns an die eigene Definition of Done gehalten. Außerdem sind einige der nicht getesteten Codezeilen auf Exception-Handling zurückzuführen, das wir zeitlich nicht mehr ausführlich testen konnten und keine große Priorität hatte wie z.B. hier:

```
86
87 def delete_assessment(assessment_id: int):
88     session = SessionMaker()
89     try:
90         session.query(Assessment).filter(Assessment.assessment_id == assessment_id).delete()
91         session.commit()
92     except Exception as e:
93         session.rollback()
94         raise AssessmentException from e
95     finally:
96         session.close()
```