LU10.L01 - Codereview: Planung bis Vorbereitung

Musterlösung zu LU10.A01 - Codereview: Planung bis Vorbereitung. Die Zeilennummern beziehen sich auf den Stand `b18ef94` von ghwalin/exam_api. Geprüft werden `data/EventDAO.py`, `service/ExamService.py` und `util/authorization.py`.

Kick-off: Verteilung der Kapitel

Bei 4 Personen erhält jede Person zwei Kapitel. So deckt jede Person eine andere Kombination ab und jedes Kapitel wird zweimal geprüft:

Gutachter	01 Bezeichner	02 Architektur	03 Darstellung	04 Kommentare
A	x	x		
B		x	x	
C			x	x
D	x			x

Bei 5 Personen erhält die fünfte Person z.B. `01` und `03`.

Musterbefunde

Die Liste ist eine Auswahl. Wiederkehrende Befunde (z.B. leere Docstring-Angaben) sind einmal mit allen Fundstellen erfasst statt pro Zeile. Das ist auch in der eigenen Tabelle sinnvoll.

01 Bezeichner

Dateiname	Zeile	Bewertung	Befund
data/EventDAO.py, service/ExamService.py	–	N	Die Modulnamen sind in PascalCase geschrieben. Module müssen kurze Namen in Kleinbuchstaben haben, z.B. `event_dao.py`, `exam_service.py`.
service/ExamService.py	97–102	N	Die Variable `oldExam` ist in camelCase geschrieben. Richtig: `old_exam`.
data/EventDAO.py	38, 50, 89	B	Die Schleifenvariable `key` wird nicht verwendet. Besser `for event in self._eventdict.values()`.
util/authorization.py	19, 41	B	Die inneren Funktionen heissen `decorator` und `wrap`. Einheitliche Namen erleichtern das Lesen.

02 Architektur

Dateiname	Zeile	Bewertung	Befund
alle drei Dateien	118, 134, 95	B	Die Prüfung `if __name__ == '__main__'` enthält nur `pass` und einen Docstring. Sie hat keine Funktion.
service/ExamService.py	87–130	N	`save` liest das alte Objekt, lädt Personen, erzeugt eine UUID, baut das Objekt und speichert es. Die Methode hat mehrere Aufgaben (SRP).
data/EventDAO.py	92–94, 104–105	N	Dateien werden mit `open` geöffnet, aber nicht mit `with`. Bei einer Exception bleibt die Datei offen.
data/EventDAO.py	88–91	N	JSON wird aus Strings zusammengesetzt statt mit `json.dumps`. Ist die Liste leer, entsteht das ungültige JSON `]`.

03 Darstellung und Formatierung

Dateiname	Zeile	Bewertung	Befund
data/EventDAO.py	81–82	B	Zwei Leerzeilen innerhalb der Klasse vor einer Methode. Richtig ist eine.
data/EventDAO.py	60, 66, 97, 103	B	Leerzeilen zwischen `def` und Docstring bzw. nach dem Docstring.
util/authorization.py	1–8	B	Die Imports sind nicht gruppiert: `datetime` (Standardbibliothek) steht nach den externen Paketen.

04 Kommentare und Dokumentation

Dateiname	Zeile	Bewertung	Befund
data/EventDAO.py, service/ExamService.py	19–24, 20–25	B	Der Docstring des Konstruktors nennt «Parameters:», führt aber keine auf.
data/EventDAO.py	64	N	Der Docstring von `read_event` nennt als Rückgabe «Exam object». Zurückgegeben wird ein `Event` oder `None`.
data/EventDAO.py	46–48, 72–74	N	`open_events` und `update_event` beschreiben Parameter und Rückgabewert nicht.
service/ExamService.py	88–92	N	Docstring von `save`: `args` ohne Beschreibung, `:return:` ohne Angabe (`True`/`False`).
service/ExamService.py, util/authorization.py	79, 15, 75	B	`:return:` ist leer (`put`, `token_required`, `read_keys`).
util/authorization.py	39	N	Die Funktion `teacher_required` hat keinen Docstring.

Ausserhalb der Codingstandards

Beim Lesen fallen auch Fehler in der Logik auf. Sie gehören ins Protokoll, auch wenn kein Standard sie verlangt.

Dateiname	Zeile	Bewertung	Befund
data/EventDAO.py	39	H	`event.timestamp` ist ein String (siehe `model/Event.py`). `event.timestamp.date()` wirft deshalb einen `AttributeError`, sobald ein Filter übergeben wird.
data/EventDAO.py	52–55	H	`open_events` fängt `TypeError` ab und übergeht ihn. Fehlerhafte Daten verschwinden ohne Meldung aus der Liste.
service/ExamService.py	68	H	`post` überschreibt den übergebenen Raum immer mit `H100`. Die Eingabe des Benutzers geht verloren.
service/ExamService.py	101–102	H	Die Einladung (`invited`) wird nur übernommen, wenn sich der Termin **ändert**. Richtig wäre umgekehrt: gleicher Termin, Einladung bleibt; neuer Termin, neue Einladung nötig.
service/ExamService.py	83	N	`put` aktualisiert eine Prüfung und meldet `201 Created`. Richtig ist `200 OK`.
service/ExamService.py	72, 85	N	Die Meldung «missing parameters» passt nicht: `save` liefert `False` nur, wenn die Prüfung nicht existiert. Richtig wäre `404`.
util/authorization.py	31	N	`except Exception` fängt jeden Fehler beim Prüfen des Tokens, auch Programmfehler. Gezielt `jwt.InvalidTokenError` und `KeyError` abfangen.
util/authorization.py	42–43	N	Fehlende Berechtigung liefert `401`. Der Benutzer ist aber angemeldet, richtig ist `403`.
util/authorization.py	76	N	`requests.get` ohne `timeout`. Antwortet der Server nicht, wartet die Applikation unbegrenzt.
util/authorization.py	61	B	`datetime.utcnow()` ist veraltet. Besser `datetime.now(timezone.utc)`.
data/EventDAO.py	2	B	`strftime` und `strptime` werden importiert, aber nicht verwendet.

Werkzeuge

Einige Befunde aus Kapitel 01, 03 und 04 findet pylint automatisch (siehe LU09.L01):

```
pylint --max-line-length=120 data/EventDAO.py service/ExamService.py
util/authorization.py
```

Die Befunde in Kapitel 02 und die Logikfehler findet kein Linter. Dafür braucht es das Review.

M450-LU10, M450-D1I

Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu10/loesungen/review_vorbereitung

Last update: **2026/11/23 09:22**