src.tests.test_create

Unit tests for the creation process of entities within the database via the REST endpoints. For creation, an ordering to the individual test cases must be specified to prevent foreign key violations during INSERTs.

---

test_create_student

Tests the POST endpoint at /student/ for student creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    student (Dict[str, Any]): Student to be created
```

Markers: - run (order=0)

test_create_teacher

Tests the POST endpoint at /teacher/ for teacher creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    teacher (Dict[str, Any]): Teacher to be created
```

Markers: - run (order=1)

test_create_assignment

Tests the POST endpoint at /asignment/ for asignment creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    asignment (Dict[str, Any]): Assignment to be created
```

Markers: - run (order=2)

test_create_mail

Tests the POST endpoint at /mail/ for mail creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    mail (Dict[str, Any]): Mail to be created
```

Markers: - run (order=3)

test_create_category

Tests the POST endpoint at /category/ for category creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    category (Dict[str, Any]): Category to be created
```

Markers: - run (order=4)

test_create_quest

Tests the POST endpoint at /quest/ for quest creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    quest (Dict[str, Any]): Quest to be created
```

Markers: - run (order=5)

test_create_subquest

Tests the POST endpoint at /subquest/ for subquest creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    subquest (Dict[str, Any]): Subquest to be created
```

Markers: - run (order=6)

test_create_attempt

Tests the POST endpoint at /attempt/ for attempt creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    attempt (Dict[str, Any]): Attempt to be created
```

Markers: - run (order=7)

test_create_question

Tests the POST endpoint at /question/ for question creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    question (Dict[str, Any]): Question to be created
```

Markers: - run (order=8)

test_create_assignment_question

Tests the POST endpoint at /assignmentQuestion/ for assignmentQuestion creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    assignment_question (Dict[str, Any]): AssignmentQuestion to be created
```

Markers: - run (order=9)

test_create_npc

Tests the POST endpoint at /npc/ for npc creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    npc (Dict[str, Any]): NPC to be created
```

Markers: - run (order=10)

test_create_challenge

Tests the POST endpoint at /challenge/ for challenge creation

```
Args:
    client (TestClient): Client for making HTTP requests to
    challenge (Dict[str, Any]): Challenge to be created
```

Markers: - run (order=11)

src.tests.test_read

Unit tests for the reading process of entities within the database via the REST endpoints.

test_read_student

Tests the GET endpoint at /student/ for student read

```
Args:
    client (TestClient): Client for making HTTP requests to
    student (Dict[str, Any]): Student to be read
```

test_read_teacher

Tests the GET endpoint at /teacher/ for teacher read

```
Args:
    client (TestClient): Client for making HTTP requests to
    teacher (Dict[str, Any]): Teacher to be created
```

test_read_all_assignments

Tests the GET endpoint at /asignment/ for asignment read

```
Args:
    client (TestClient): Client for making HTTP requests to
    asignment (Dict[str, Any]): Assignment to be created
```

test_read_all_mails

Tests the GET endpoint at /mail/ for mail read

```
Args:
    client (TestClient): Client for making HTTP requests to
    mail (Dict[str, Any]): Mail to be created
```

test_read_category

Tests the GET endpoint at /category/ for category read

```
Args:
    client (TestClient): Client for making HTTP requests to
    category (Dict[str, Any]): Category to be created
```

test_read_quest

Tests the GET endpoint at /quest/ for quest read

```
Args:
    client (TestClient): Client for making HTTP requests to
    quest (Dict[str, Any]): Quest to be created
```

test_read_subquest

Tests the GET endpoint at /subquest/ for subquest read

```
Args:
    client (TestClient): Client for making HTTP requests to
    subquest (Dict[str, Any]): Subquest to be created
```

test_read_attempt

Tests the GET endpoint at /attempt/ for attempt read

```
Args:
    client (TestClient): Client for making HTTP requests to
    attempt (Dict[str, Any]): Attempt to be created
```

test_read_question

Tests the GET endpoint at /question/ for question read

```
Args:
    client (TestClient): Client for making HTTP requests to
    question (Dict[str, Any]): Question to be created
```

test_read_assignment_question

Tests the GET endpoint at /assignmentQuestion/ for assignment_question read

```
Args:
    client (TestClient): Client for making HTTP requests to
    assignment_question (Dict[str, Any]): Question to be created
```

test_read_npc

Tests the GET endpoint at /npc/ for npc read

```
Args:
    client (TestClient): Client for making HTTP requests to
    npc (Dict[str, Any]): NPC to be created
```

test_read_challenge

Tests the GET endpoint at /challenge/ for challenge read

```
Args:
    client (TestClient): Client for making HTTP requests to
    challenge (Dict[str, Any]): Challenge to be created
```

src.tests.test_update

Unit tests for the update process of entities within the database via the REST endpoints.

---

test_update_student

Tests the PATCH endpoint at /student/ for student update

```
Args:
    client (TestClient): Client for making HTTP requests to
    student (Dict[str, Any]): Student to be updated
```

test_update_teacher

Tests the PATCH endpoint at /teacher/ for teacher update

```
Args:
    client (TestClient): Client for making HTTP requests to
    teacher (Dict[str, Any]): Teacher to be created
```

test_update_assignment

Tests the PATCH endpoint at /asignment/ for asignment update

```
Args:
    client (TestClient): Client for making HTTP requests to
    asignment (Dict[str, Any]): Assignment to be created
```

test_update_attempt

Tests the PATCH endpoint at /attempt/ for attempt update

```
Args:
    client (TestClient): Client for making HTTP requests to
    attempt (Dict[str, Any]): Attempt to be created
```