**1. AttributeError: 'SweetBase' object has no attribute 'name'** (Failing tests: test\_valid\_sweet\_model, test\_max\_length\_name)

* **Problem:** This error means that when pytest tries to access sweet.name (or any other attribute) on an instance of SweetBase, it cannot find that attribute.
* **Likely Cause:** This is the most critical issue. It suggests that SweetBase either:
  + **Is not correctly defined as a Pydantic BaseModel with the name field.** Perhaps SweetBase is an empty class, or you're importing a different SweetBase than the one with your Pydantic model definition.
  + **The Pydantic model (which you previously shared as Sweet) is not being imported as SweetBase in your test file.**
* **Impact:** If Pydantic isn't correctly processing SweetBase as a model, then none of its fields or validations will be applied, leading to cascading failures.

**2. Failed: DID NOT RAISE <class 'ValueError'>** (Failing tests: test\_missing\_required\_field\_name, test\_negative\_price, test\_invalid\_discount\_over\_100, test\_invalid\_category, test\_invalid\_image\_url)

* **Problem:** These tests expect a ValueError to be raised by the Pydantic model when invalid data is provided (e.g., empty name, negative price, invalid category). However, no ValueError was raised.
* **Likely Cause:** This directly ties back to the AttributeError. If SweetBase is not correctly configured as a Pydantic model (as mentioned above), then Pydantic's automatic validation (which would normally raise these ValueErrors) simply isn't happening.

**3. AssertionError: assert None is False** (Failing test: test\_is\_available\_false)

* **Problem:** This is the recurring issue where sweet.is\_available returns None instead of False (or True for the other test where it was expected).
* **Likely Cause:** Again, if SweetBase isn't properly initialized as your Sweet Pydantic model, then the @property methods like is\_available might not be behaving as expected, or they might be attempting to access attributes that don't exist.

**4. test\_valid\_sweet\_model (FAILED)**

* **Assertion Failure:** assert sweet.is\_available is True
* **Error:** AssertionError: assert None is True
* **Problem:** The is\_available property of your SweetBase model is returning None instead of True when the quantity is greater than 0. The test expects True but received None.

**5. test\_is\_available\_false (FAILED)**

* **Assertion Failure:** assert sweet.is\_available is False
* **Error:** AssertionError: assert None is False
* **Problem:** Similar to the first failure, the is\_available property is returning None instead of False when the quantity is 0. The test expects False but received None.

**6.** **test\_add\_sweet\_success (FAILED)**

* **What was expected:** The test expected the API to return a 201 Created status code, indicating that the sweet was successfully added.
* **What actually happened:** The API returned a 404 Not Found status code.
* **Meaning of the 404 Not Found:** This error means that the endpoint /sweets for the POST request (which is used to add a new sweet) either:
  1. **Does not exist** in your FastAPI application.
  2. **Is not correctly defined** to handle POST requests at that specific path.
  3. There's a **typo in the endpoint path** in your FastAPI code or in the test.