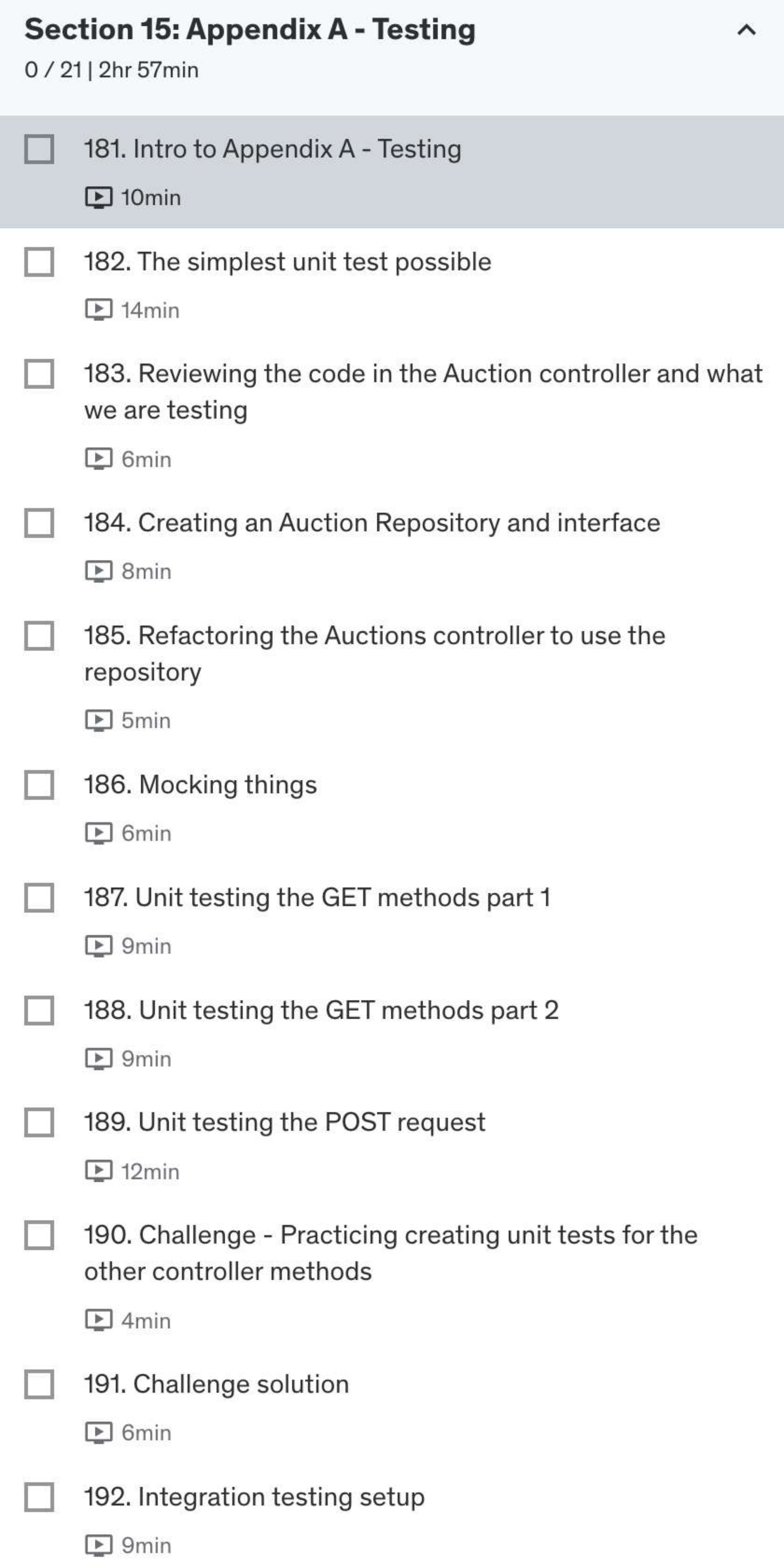


Section 15: Appendix A - Testing

0 / 21 | 2hr 57min

□ 181. Intro to Appendix A - Testing

▶ 10min

□ 182. The simplest unit test possible

▶ 14min

□ 183. Reviewing the code in the Auction controller and what we are testing

▶ 6min

□ 184. Creating an Auction Repository and interface

▶ 8min

□ 185. Refactoring the Auctions controller to use the repository

▶ 5min

□ 186. Mocking things

▶ 6min

□ 187. Unit testing the GET methods part 1

▶ 9min

□ 188. Unit testing the GET methods part 2

▶ 9min

□ 189. Unit testing the POST request

▶ 12min

□ 190. Challenge - Practicing creating unit tests for the other controller methods

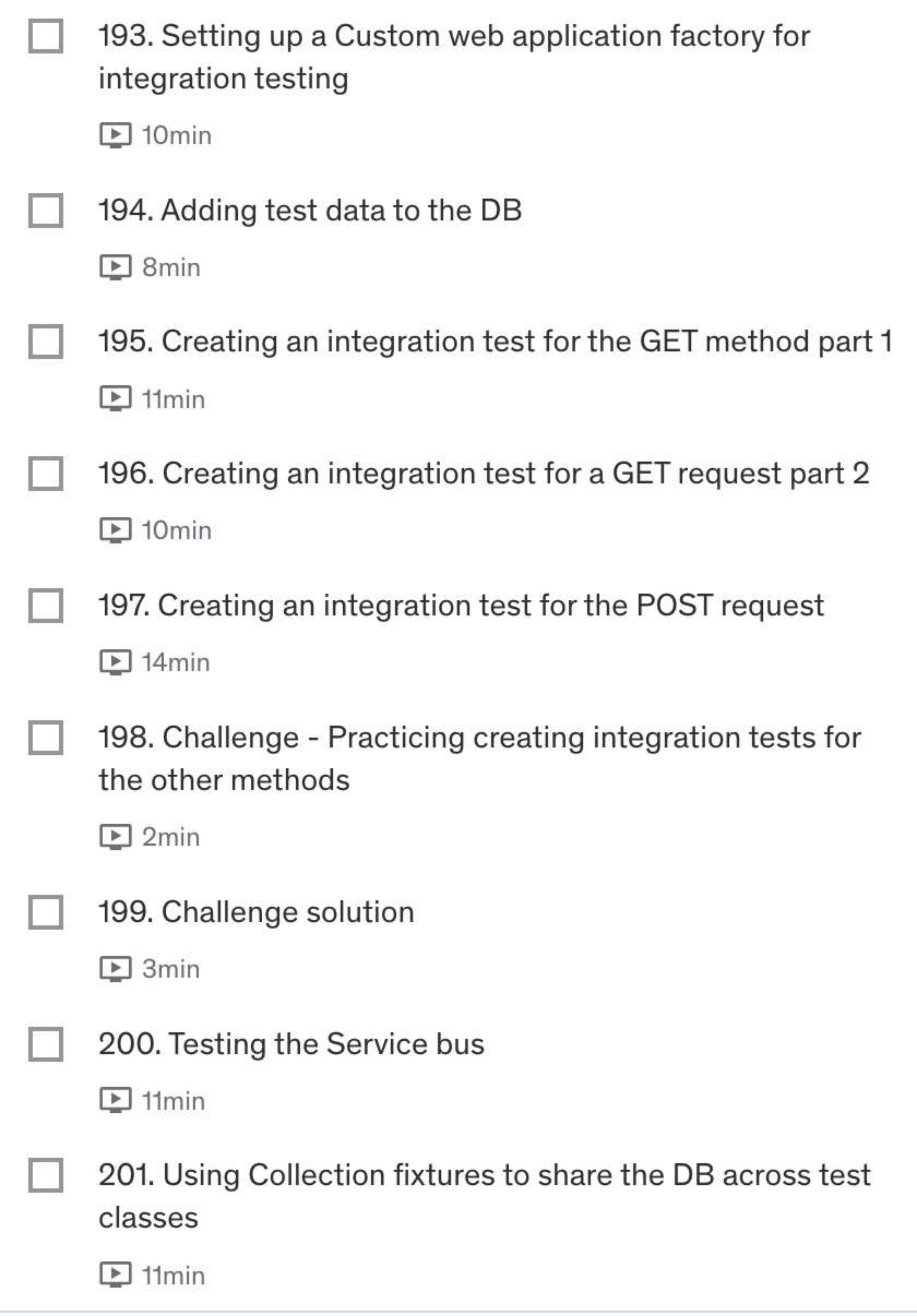
▶ 4min

□ 191. Challenge solution

▶ 6min

□ 192. Integration testing setup

▶ 9min



□ 193. Setting up a Custom web application factory for integration testing

▶ 10min

□ 194. Adding test data to the DB

▶ 8min

□ 195. Creating an integration test for the GET method part 1

▶ 11min

□ 196. Creating an integration test for a GET request part 2

▶ 10min

□ 197. Creating an integration test for the POST request

▶ 14min

□ 198. Challenge - Practicing creating integration tests for the other methods

▶ 2min

□ 199. Challenge solution

▶ 3min

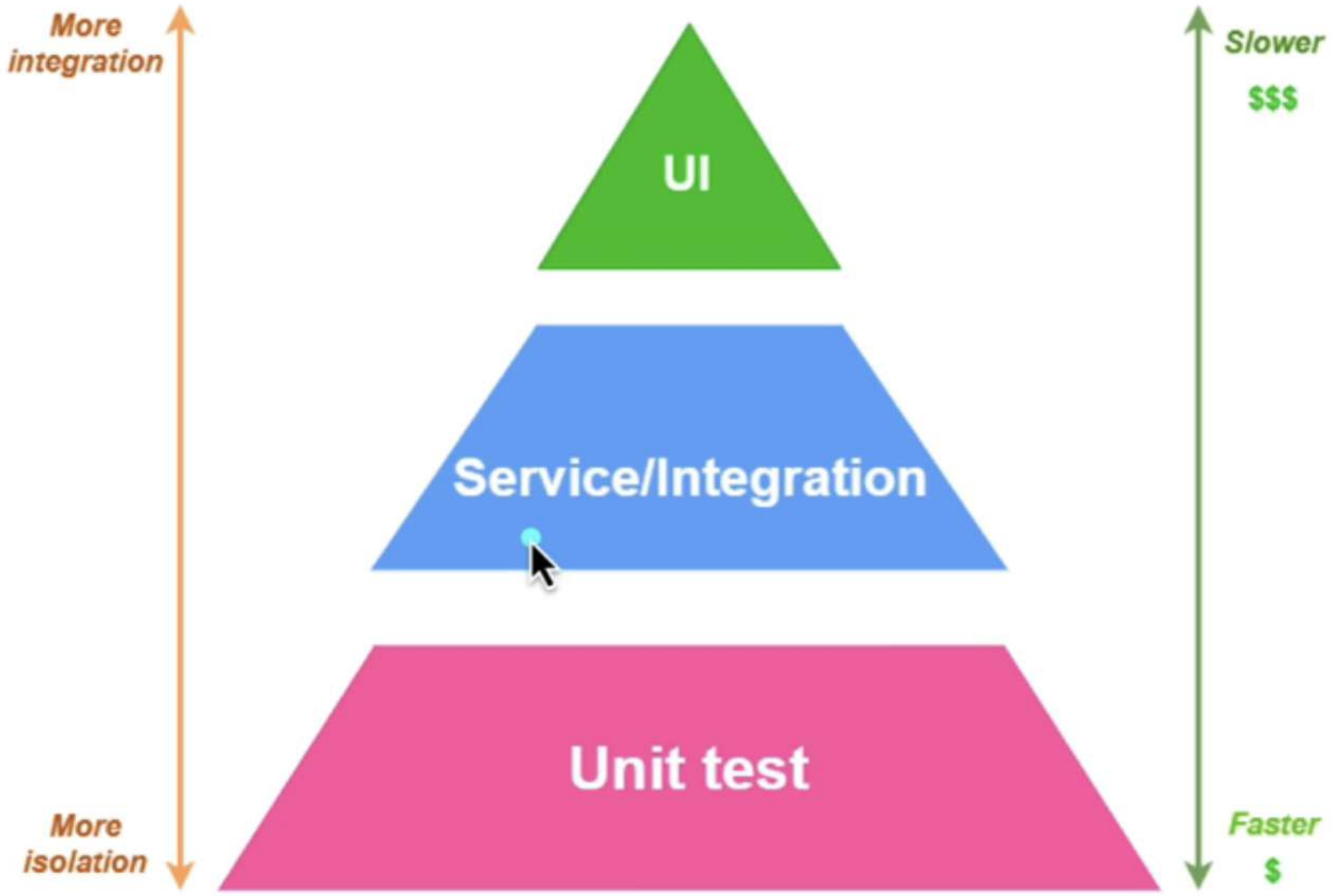
□ 200. Testing the Service bus

▶ 11min

□ 201. Using Collection fixtures to share the DB across test classes

▶ 11min

The testing pyramid



Documentation

Less coupled code

A good test is…

- Fast
- Isolated
- Repeatable
- Self-Checking
- Timely

What should I test?

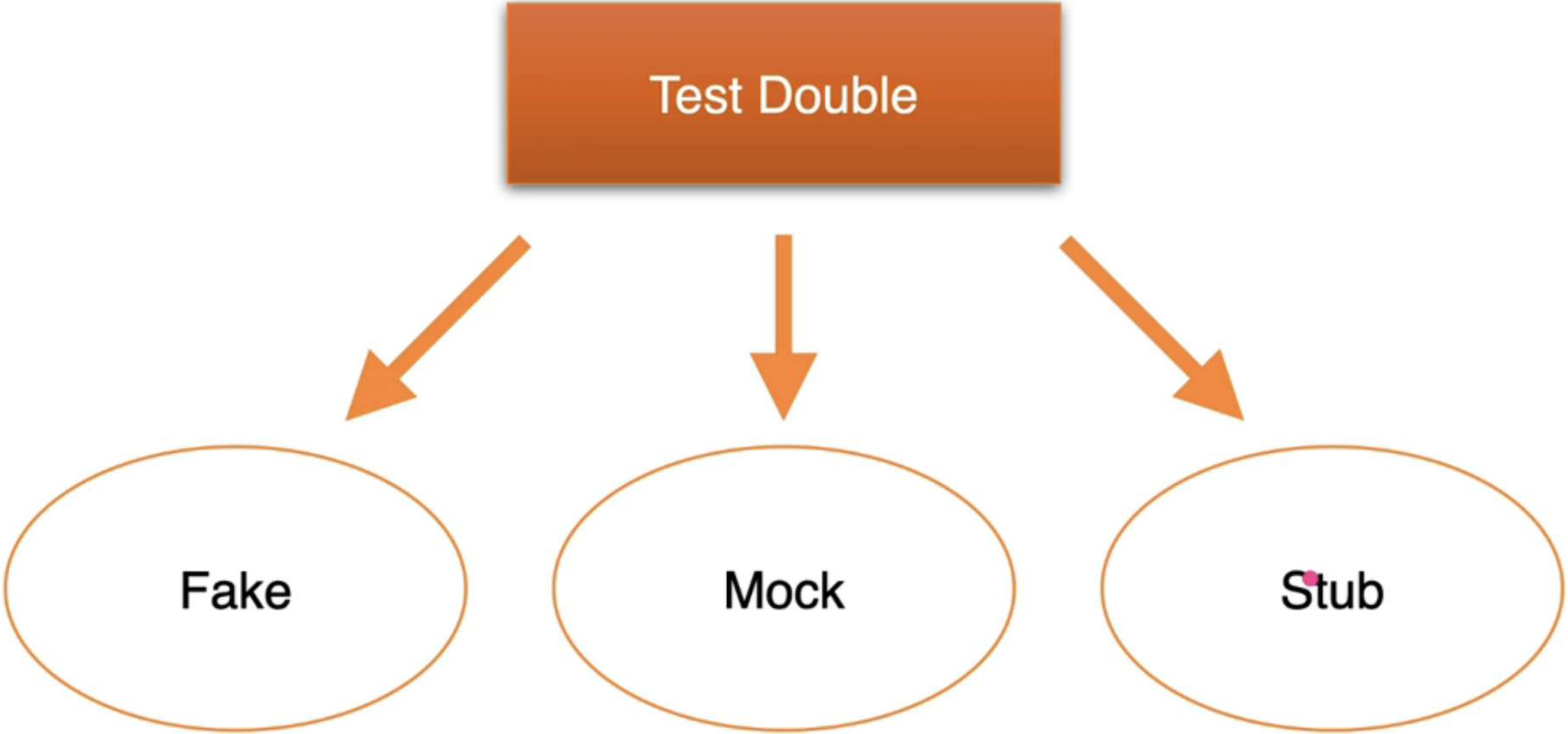
- API Controllers
  - The action returns the correct type of response
  - Invalid parameters return the correct error
  - The action calls the correct method on the repo
  - If the response is a model, verify the model type

Unit Tests

Tests 1 thing

Integration Tests

Tests more than 1 thing



Test Double

Fake

Mock

Stub

XUnit

NUnit

MSTest