1. Following is Definition from WireMock Docs.



* 1. Suppose you’re developing S1 Service in your company.
  2. S2, S3, S4 are external Services which will be consumed by S1.
  3. S2, S3 services are paid and for each request you have to pay something.
  4. S4 is yet to be developed.
  5. So, when performing integration testing S1 while developing, we may face some issues like
     1. S4 Service is not still developed and our S1 service using S4.
     2. We have to pay for each hit which may cost us a lot.
     3. Maybe API owner has provided SandBox but the responses may not be deterministic. There are chances of network failure.

1. **Solution**: Mocking. We mock these external services with which S1 has to interact. Some solutions are as follows
   1. We have some options like
      1. **Mockito**: To mock a class object.
      2. **MockMVC**: By Spring Boot
      3. **TestTemplate**:

**But they are not near to production like behavior**. These are the hard-coded things you have to do.

* + - * You have to simulate the failure scenarios.
      * How to check the circuit breaker.
      * How to check out Timeout.  
        WireMock plays very important role in all of these cases.

1. What we’re going to discuss in this course.  
   Graphical user interface, text

   Description automatically generated with medium confidence