## Test double encapsulation

In file “PhoneBook.cs” there’s a method “LoadContacts” which consumes a JSON file.  
That process fits into my test double “PhoneBookSpy” -> theres logic after which I check a property.

Could that LoadContacts-method be a stub inside of a spy test double?  
Technically there’s a need for a simulated return-a-hardcoded-value (stub) process inside of the spy.

I am simulating the ”PhoneBookOperations” Contacts using a spy, which in turn relies on a stub?  
(Simulating the process->data and simulating the file input seem like two different things)

Testing responsibility

I am testing the property “Contacts” for the class “PhoneBookOperations” in the context of   
returning the correct value for that property.

1. Literally “Contacts => Geeft alle contacten terug.”
   1. No literal definition of specific acceptance criteria (“all” is abstract, not specific)
   2. Does “all” permit nothing, or is null an unacceptable answer in any test case?
2. Delving into the inner workings…
   1. A file input stream (url to deserialized json)
      1. Is the testing responsability recursive? Do I test any underlaying mechanic?
      2. Opposite to that, do I stub such things or declare them out of scope?
   2. Testing does not leave side effects
      1. How to test things, like I/O that have to cause side effects? (Cleanup stage)
   3. The current application state contains no detectable file for the example url
      1. How would I test being able to read a present file -> altering the state  
         (Adding a file to test being able to read a file would leave a test-state)