

What are BASIC Principles?

The evil test bastards are coming to increase the test complexity and grind you down. Fight back with the BASIC principles!

Black-Box	Annotative	Single Door	Independent	Copy
				
<p>Test the outcome of the component, not the internals</p> <ul style="list-style-type: none">• Test through REST or public API• Strive for less mocking• Use test doubles at the boundaries of your component	<p>Each test has a predictable and declarative structure</p> <ul style="list-style-type: none">• A Test name should contain 3 repeatable parts• Apply the AAA pattern• Assertion should be checked in a declarative style• Test contains no more than 7 statements	<p>Each test check one action and one corresponding outcome</p> <ul style="list-style-type: none">• Each test simulates ONE application action like a single function call• Each test checks one outcome that corresponds to this call. For example - The modified app data, or the response• Apply the minimum amount of assertions needed to prove that the outcome is successful	<p>A test is a tiny universe that shares no state with others</p> <ul style="list-style-type: none">• A Test is a 7 statement long problem• Avoid sharing objects or data with other tests• A test should fail or succeed regardless of others - Don't assume a specific execution order• Share code sparingly using small helpers	<p>Include within the test anything that is NECESSARY to understand the test intent</p> <ul style="list-style-type: none">• The test outcome should be inferred easily• Avoid placing meaningful information outside of the test• Copy code if necessary• Focus the test - Take non-meaningful info to an external helper or hook• Use factories to craft long structures - Pass meaningful overrides as params

Visit my workshop at NodeConfRemote.com to learn more

