Test Plan Template

*Blizzard Test Engineer Interview*

|  |  |
| --- | --- |
| Full Name | Bill Bucher |
| Email Address | bill.bucher@gmail.com |
| Date | 9/17/2013 |

* **Scope**

The scope of this test plan includes the testing for the Item API. The following calls will be covered:

1. Individual Item API call
2. Item Set API call

Other APIs will be covered in the appropriate test plans, one for each API.

This scope makes the most sense, as changes to one of the APIs may not require testing of other APIs. Breaking up the testing by different APIs provides flexibility to the test group.

* **Test Strategy**

Functional testing of the Item API will be performed via automation. NUnit will be the framework used to execute the tests. The tests and supporting infrastructure code will be written in C#.

Unit, integration, regression, and acceptance testing will be outside the scope of this test plan, and should be covered elsewhere.

Tests will follow the same basic pattern:

1. User submits an HTTP request to the desired service endpoint
2. User retrieves the JSON output
3. User evaluates the JSON, and asserts specific data are correct based on expected output

Test execution can be performed directly in Visual Studio with NUnit and TestDriven.NET.

1. Load the test fixture (TestFixture.cs.)
2. Right click anywhere in the file, and select the TestDriven.Net ‘Run Test(s)’ option.
3. All tests should run successfully

Alternately, you can run directly via NUnit within VS.

1. To the left of the [TestFixture], select ‘Run All’ from the NUnit Fixture icon. You can run test individually as well.

Testing via automation supports the project goal of shipping product that meets the highest quality standards through consistency and repeatability of the tests.

* **Entry and Exit Criteria**

Functional testing of the Item API will commence when the following criteria are met:

1. feature is code complete
2. unit and integration tests are 100% passing
3. QA has successfully completed acceptance testing of the Item API

Testing of the Item API will be considered complete when one of the following criteria are met:

1. all functional tests have been executed against the final build of the Item API, and are passing
2. less than 100% of the functional tests are passing, but the outstanding defects are determined to not be "show stoppers" by the bug triage committee which shall meet daily during the closing days of the release to ensure that deadlines are met.
3. **Test Cases**
4. Using the WoW Item API, verify that the item name is “Finkle's Lava Dredger” for a given item with id=18803
5. Using the WoW Item API, verify that the item set name is "Deep Earth Vestments" for the given item set id=1060
6. Using the WoW Item API, verify that item set 1060 contains the expected items [76749,76750,76751,76752,76753].
7. Using the WoW Item API, verify that the items in [76749,76750,76751,76752,76753] all contain an itemSet equal to: {"id":579,"name":"Gladiator's Regalia","setBonuses":[{"description":"+35 PvP Resilience.","threshold":2},{"description":"Increases the damage done by your Arcane Blast, Fireball, Frostbolt, Scorch, Pyroblast, Arcane Missiles, and Frostfire Bolt spells by 5%.","threshold":4}],"items":[25854,25855,25857,25856,25858]}
8. Using the WoW Item API, verify that the individual items in the "Deep Earth Vestments" item set are not auctionable
9. Using the WoW Item API, verify that none of the class specific tiered armor sets are auctionable.
10. Using the WoW Item API, verify that the items in the “Gladiator’s Regalia” set are auctionable.
11. Using the WoW Item API, verify that the set bonuses for “Gladiator’s Regalia” are equal to **[{"description":"+35 PvP Resilience.","threshold":2},{"description":"Increases the damage done by your Arcane Blast, Fireball, Frostbolt, Scorch, Pyroblast, Arcane Missiles, and Frostfire Bolt spells by 5%.","threshold":4}]**
12. Using the WoW Item API, verify that the allowableClasses attribute contains the correct classes for each of the armor tokens for each armor slot (head, shoulders, chest, legs, hands) for the Tier 14 raid set. For Example, the allowableClasses for Helm of the Shadowy Conqueror (Tier 14 armor token head piece) are [2,5,9]
13. Using the WoW Item API, verify that druid (allowableClasses=[11]) is the only class that can wear items from the "Deep Earth Vestments" item set
14. Using the WoW Item API, verify that the description for "Finkle's Lava Dredger” is "Property of Finkle Einhorn, Grandmaster Adventurer”
15. Using the WoW Item API, verify that the bonus stats for "Finkle's Lava Dredger” is: [{"stat":51,"amount":15},{"stat":5,"amount":24},{"stat":6,"amount":22},{"stat":7,"amount":25}]
16. Using the WoW Item API, verify that the WeaponInfo values for "Finkle's Lava Dredger” are: {"damage":{"min":159,"max":239,"exactMin":159.28119,"exactMax":239.0},"weaponSpeed":2.9,"dps":68.66917}
17. Using the WoW Item API, verify that you can retrieve an item of quality (poor, common, uncommon, rare, epic, artifact, heirloom.) Verify that the returned data is correct for a given item.
18. Using the WoW Item API, verify that you can retrieve one of each item type (armor, weapon, consumable, gem, mount, glyph, container, pet, and recipe, projectile.) Verify that the data returned is correct for the given item type
19. Using the WoW Item API, verify that the socket info is correct for item 98828 ("Crafted Malevolent Gladiator's Silk Robe".) Expected result is: {"sockets":[{"type":"RED"},{"type":"YELLOW"}],"socketBonus":"+120 PvP Power"}
20. Using the WoW Item API, verify that item 82403 ("Windwool Boots") does not contain socket info
21. Using the WoW Item API, verify that specifying an invalid item id returns {"status":"nok", "reason": "unable to get item information."}
22. Using the WoW Item API, verify that specifying an invalid item set id returns {"status":"nok", "reason": "unable to get item set information."}

* **Dependencies**

The following tools will be required to develop and execute the automated tests for the Item API:

1. Visual Studio 2012 (professional or higher)
2. NUnit 2.6.2
3. TestDriven.NET (not required, but handy for running tests in VS.)
4. ReSharper (latest version)
5. git