List of test cases and their mapping for the Best Buy API:

|  |  |
| --- | --- |
| **Manual test in Postman** | **Respective automation tests** |
| **GET PRODUCTS TEST** |  |
| Retrieve list of all products | [TestCase("/products", TestName = "Check all products response status and content type")] |
| Retrieve list of all products sorted e.g. by price etc. | [TestCase("/products?$sort[price]=-1", TestName = "Check product with highest price")] |
| Retrieve all products and check total, limit and skip etc. | [TestCase("/products", TestName = "Check all products total, limit and skip are correct")] |
| Retrieve all products with different search criteria e.g. by shipping limit, category etc. | [TestCase("/products?shipping[$gt]",10, TestName = "Check product with shipping Price not more than")]  [TestCase("/products?category.name=", "Coffee Pods", TestName = "Check product with category")]  [TestCase("/products?name[$like]=\*star+wars\*&price[$lt]=30", "star+wars", 30, TestName = "Check with name and within range")] |
| Retrieve product by id | [TestCase("/products/9132294", TestName = "Check product with id is correct")] |
| **POST PRODUCTS TEST** |  |
| Create new product | [TestCase("/products", "test name","test type","test upc",150,"test description","test model",TestName = "Create new product")] |
| Create with missing parameters | [TestCase("/products", "test name", "test type", "test upc", 150, "test description", TestName = "Create new product with mandatory data missing")] |
| Create with invalid parameters | [TestCase("/products", "test name", "", "test upc", 150, "test description", "test model", TestName = "Create new product with type less than1 char long")]  [TestCase("/products", "test name", "11", "", 150, "test description", "test model", TestName = "Create new product with upc less than 1 char long")]  [TestCase("/products", "test name", "11", "11", 150, "", "test model", TestName = "Create new product with decription less than1 char long")]  [TestCase("/products", "test name", "11", "11", 150, "11", "", TestName = "Create new product with model less than 1 char long")] |
| **GET CATEGORIES TEST** |  |
| Retrieve list of all products | [TestCase("/products", TestName = "Check all products response status and content type")] |
| Retrieve all categories with different search criteria e.g. by name | [TestCase("/categories?name[$like]=\*TV\*", "TV", TestName = "Check all categories witn Name containing TV")] |
| **POST CATEGORIES TEST** |  |
| Create new category | [TestCase("/categories", TestName = "Create category")] |
| Create with missing parameters | [TestCase("/categories", "test1", TestName = "Create category with missing data")] |
| Crate duplicated record | [TestCase("/categories", "test1","test1", TestName = "Create duplicate category")] |
| **GET STORES TEST** |  |
| Retrieve list of all products | [TestCase("/products", TestName = "Check all products response status and content type")] |
| Retrieve all categories with different search criteria e.g. by name | [TestCase("/categories?name[$like]=\*TV\*", "TV", TestName = "Check all categories witn Name containing TV")] |
| **POST STORES TEST** |  |
| Create new store | [TestCase("/stores", "New Store", "BigBox", "123 Fake St","", "Springfield", "MN", "55123", "Mon: 10-9; Tue: 10-9; Wed: 10-9; Thurs: 10-9; Fri: 10-9; Sat: 10-9; Sun: 10-8", TestName = "Create new store")] |
| Create with missing parameters | **TBD** |
| **GET SERVICES TEST** |  |
| Retrieve list of all services | **TBD** |
| Retrieve all SERVICES with different search criteria e.g. by name | **TBD** |
| **POST SERVICE TEST** |  |
| Create new service | **TBD** |
| Create with missing parameters | **TBD** |
| **GET heath check** | [TestCase("/healthcheck", TestName = "Health Check")] |
|  |  |
|  |  |

The majority of tests are for the product GET/POST APIs as other APIs are similar with this one. Also framework must be extended with

1. More negative cases like POST with duplicates, invalid body data e.g. some fields missing or invalid as well as creating duplicates and sending incorrect parameters.
2. Coverage for Stores, was skipped as similar to categories

README:

1. Run BestBuy on localhost:3030
2. Install VS Community 2017
3. Open Solution
4. Restore packages and build
5. If missing libraries select Manage Nuget packages and check following packages are installed:

* IdentityModel 1.12.0
* Newtonsoft.Json 12.0.1
* NJunut 3.10.1
* RestSharp 106.6.10
* System.Net.HTTP 4.3.4

1. Check packages are in the References with correct versions
2. After build is successful open Test -> Windows-> Test Explorer and execute the tests loaded

All tests are located in \BestBuy.API.Tests\TestCollection\BestBuyAPITests.cs