

```
using System;
using System.IO;
using System.Linq;
using RestSharp;
using System.Net;
using NUnit.Framework;
using Newtonsoft.Json.Linq;

namespace V3_Coding_Test
{
    public class Assignment
    {
        //hold instance of rest client class
        private static RestClient client;

        [OneTimeSetUp]
        // a method to initialize the RestClient field with a base URL
        public void InitializeClient() => client = new RestClient("https://
        openlibrary.org");

        //This test is performed to obtain the total number of books matching a
        //specific title and return their keys for years starting from 2000.
        [Test]
        public void CheckGetBooks()
        {
            // a request to the Open Library API to search for books with the
            // title "Goodnight Moon Base"
            var request = new RestRequest("search.json").AddQueryParameter
            ("title", "Goodnight Moon");

            // a call to the Open Library API to get the response to the above
            // request
            var response = client.Get(request);

            Assert.AreEqual(HttpStatusCode.OK, response.StatusCode);
            // a variable to hold the JSON response content as a JToken object
            // after json is deserialized
            var content = JToken.Parse(response.Content);

            // a variable to hold the total number of books found in the search
            // results with the title "Goodnight Moon"
            var TotalNumberOfBooks = content.SelectToken("numFound").ToString
            ();

            Console.WriteLine($"Total number of books with the title matching
            exactly [Goodnight Moon]: " +
            $"{TotalNumberOfBooks}");
        }
    }
}
```

```
// a variable to hold an array of JToken objects representing the search results
var Books = content.SelectToken("docs");

foreach (var Book in Books)
{
    if (Book.SelectToken("publish_year") != null &&
        Book.SelectToken("publish_year").Count() > 0 && (int)
        Book.SelectToken("publish_year")[0] >= 2000)
    {
        Console.WriteLine(Book.SelectToken("key"));
    }
}

}

//This test is performed to check whether the response attained from the api matches the one provided
[Test]

public void CheckResponseMatch()
{
    // a request to the Open Library API to search for books with the title "Goodnight Moon Base"
    var request = new RestRequest("search.json").AddQueryParameter("title", "Goodnight Moon Base");

    // a call to the Open Library API to get the response to the above request
    var response = client.Get(request);

    Assert.AreEqual(HttpStatusCode.OK, response.StatusCode);

    // a variable to hold the JSON response content as a JToken object after json is deserialized
    var contents = JToken.Parse(response.Content);

    var expectedJson = File.ReadAllText("C:/Users/abdul/Documents/Study/Software tessting/API Testing/New folder/V3_Coding_Test/V3_Coding_Test/Resources/ExpectedResponse.json");
    var expected = JToken.Parse(expectedJson);

    // a boolean variable to hold the result of comparing the actual response with the expected response
    bool isEqual = JToken.DeepEquals(contents, expectedJson);

    if (isEqual)
```

```

    {
        Console.WriteLine($"The response for Goodnight Moon Base
            matches the expected response.");
    }
    else
    {
        Console.WriteLine($"The response for Goodnight Moon Base does
            not match the expected response:");
        contents = contents.SelectToken("docs");
        expected = expected.SelectToken("docs");
    }

    // Create an array of expected properties to compare.
    string[] properstiesName = new string[] { "key", "type", "seed",
        "title", "title_suggest", "title_sort",
        "edition_count", "edition_key", "publish_date", "publish_year",
        "first_publish_year", "number_of_pages_median",
        "isbn", "last_modified_i", "ebook_count_i", "ebook_access", "has_fulltext", "public_
        scan_b", "readinglog_count",
        "want_to_read_count", "currently_reading_count", "already_read_count", "cover_edit
        ion_key", "cover_i", "publisher",
        "language", "author_key", "author_name", "subject", "publisher_facet", "subject_face
        t", "_version_", "subject_key"};

    // Iterate through each expected property and compare it with the
    // corresponding actual property.
    foreach (var proper in properstiesName)
    {
        foreach (var expect in expected)
        {
            foreach (var content in contents)
            {
                expected = JToken.Parse(expectedJson);
                expected = expected.SelectToken("docs");

                if (content.SelectTokens(proper) != null &&
                    expect.SelectTokens(proper) != null)
                {
                    if (!JToken.DeepEquals(content.SelectToken(proper),
                        expect.SelectToken(proper)))
                    {
                        Console.WriteLine($"Difference found in
                            property : {proper}\n");
                    }
                }
            }
        }
    }
}

```

```
Console.WriteLine("actual");
Console.WriteLine(content.SelectToken(proper)

Console.WriteLine("expected");
Console.WriteLine(expect.SelectToken(proper) +
```