

## Test Cases

Covered by JUnit and Manual Testing

Test case id:1

Scenario: Verify that the xml returned by the Zillow API call is constructed as expected.

Test Step: make a get request in order to retrieve the information stored in an xml from the api

Expected Result: XML is based upon the three given Strings path, address and cityZipCode

Actual outcome: XML is constructed correctly

Test case id:2

Scenario: Verify that the build link used to make the api call is correctly constructed

Test Step: construct the build link based on the input parameters path, address and cityZipCode

Expected Result: Build link is built correctly

Actual Outcome: Build link is built as expected

Covered By Manual testing (to be automated)

Test case id:3

Scenario: verify that the CLARIFAI Api call uses the given image

Test Step: verify that Clarifai uses the correct image path

Expected result: the image path is correct

Actual result: The image path is correctly passed to the API

Test case id: 4

Scenario: Given that an image is correctly uploaded

Then the Submit button is working as expected

Test Step: Press Submit button with image inserted

Expected Result: After pressing submit the image get tagged correctly

Actual result: The submit button works as expected

## JUnit Tests

Java EE - TAIP/src/test/java/TAIP/TAIP/GetDataTest.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access Java EE

Project Explorer

- TAIP
  - src/main/java
    - TAIP.TAIP
      - DataFetch.java
      - House.java
    - src/test/java
      - TAIP.TAIP
        - GetDataTest.java
    - JRE System Library [J2SE-1.5]
    - Maven Dependencies
    - JUnit 4
    - src
    - TAIP-master
    - target
    - pom.xml
    - test

DataFetch.java House.java GetDataTest.java

```
9
10 public class GetDataTest {
11
12     private static DataFetch dataFetch=DataFetch.getInstance();
13
14     @Test
15     public void fetchDataTest() throws IOException, URISyntaxException
16     {
17         String path="/webservice/GetUpdatedPropertyDetails.htm";
18         String address="2114 Bigelow Ave";
19         String cityzipcode="Seattle WA";
20         String expectedResultDisregardingTime="<?xml version='1.0' encoding='utf-8'?'><UpdatedPropertyDetails:updatedPropertyDe
21         String actualResult=dataFetch.getData(path, address, cityzipcode);
22         assertTrue(actualResult.contains(expectedResultDisregardingTime));
23     }
24
25
26
27     @Test
28     public void fetchBuildLink() throws URISyntaxException{
29         String expectedResult = "http://www.zillow.com/webservice/GetUpdatedPropertyDetails.htm?zws-id=X1-ZWz19hkt3gwmby_9hc6e&zpi
30         String path="/webservice/GetUpdatedPropertyDetails.htm";
31         String address="2114 Bigelow Ave";
32         String cityzipcode="Seattle WA";
33         String actualResult = dataFetch.buildLink(path, address, cityzipcode);
34         assertEquals(actualResult, expectedResult);
35     }
36
37 }
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit

Finished after 1,075 seconds

Runs: 2/2 Errors: 0 Failures: 0

TAIP.TAIP.GetDataTest (Runner: JUnit 4) (1,058 s)

  - fetchBuildLink (0,001 s)
  - fetchDataTest (1,057 s)

Failure Trace

Writable Smart Insert 22:73

Ask me anything

ENG 09:28 28.11.2016

## AOP in Javascript

```
var obtainKeys= (function() {  
    var keys=getKeys() || {};  
  
    return {  
        Keys: function() {  
            return keys;  
        }  
    }  
})();  
  
//function written for logging purposes in order to retrieve confirmation  
// when the required credential are obtained in order to make the api call  
function gotKeysConfirmation() {  
    console.log("Got client id and client secret");  
}  
  
var LogInfo = AOP.aspect(gotKeysConfirmation).after(getKeys);  
  
LogInfo();
```

```

//functions written for logging purposes before and after the API is called
function beforeApiCall()
{
    console.log("Preparing to make the APi call and assign colored tags");
}

function afterApiCall()
{
    console.log("Color tags have been assigned");
}

var BeforeAssigningColorTags = AOP.aspect(beforeApiCall).before(assignColor);
var AfterAssigningColorTags = AOP.aspect(afterApiCall).after(assignColor);
BeforeAssigningColorTags();
AfterAssigningColorTags();

function assignColor (prob, index, length) {
    if (prob > 0.9) {
        return "success"
    }
    else if (prob > 0.85 && prob <= 0.9) {
        return "info"
    }
    else {
        return "warning"
    }
}

```

Full code can be found on [GitHub](#).