Using the Objects Repository

VeriSoft framework supports object repositories. It is a very similar solution to the page factory pattern.

General

In order to use the Object repository feature you will need to do the following:

1. Create an object repository, which is a JSON file with a specific format.
2. Annotate your WebElement objects in your page object accordingly.
3. Initialize the WebElements (not needed if you derive your page object from BasePage)
4. Use the WebElement objects like in any other page object.

Repository Logic

Each object in the object repository has a unique id. This unique id is called `objectId`. For each `object` there can be several fitting locators. For each `objectId` there is a defined array of locators, each locator has a type, a value, and a grade. A higher grade means a stronger locator. You can use each locator strategy more than once. For instance, you can define 2 XPath locators.

When calling the WebElement, the framework will sort all the locators from the strongest (with a higher grade) to the weaker (with a lower grade) and try to apply the `findElement` method. **Note!** Currently, there is no support for a list of WebElements. Once a locator strategy finds an object on the DOM and the WebElement is created, the framework uses this locator for the current action. This logic repeats every time the WebElement is being used.

Structure of the object repository JSON file:

Currently, you will need to follow these instructions precisely:

The file structure look like this:

```json
{
  "objectsRepository": [
    {
      "objectId": "WIKI-MAIN-SEARCH",
      "pageName": "WikiMainPage",
      "locators": [
        {
          "type": "id",
          "value": "searchInput",
          "grade": 80
        },
        {
          "type": "xpath",
          "value": "//badXpath",
          "grade": 90
        },
        {
          "type": "css",
          "value": "#NoCss",
          "grade": 50
        }
      ]
    },
    {
      "objectId": "WIKI-SEARCH-BAR",
      "pageName": "WikiMainPage",
      "locators": [
        {
          "type": "xpath",
          "value": "//input[@name='search']",
          "grade": 10
        },
        {
          "type": "id",
          "value": "BadId",
          "grade": 80
        },
        {
          "type": "css",
          "value": "#BadCss",
          "grade": 50
        }
      ]
    }
  ]
}
```

The Page Object

In the page object, the only difference is the way we reference the WebElement:

```
@ObjectRepositoryItem(id = "WIKI-MAIN-SEARCH")
private WebElement searchBar;
```

If you extend VeriSoft's `BasePage` (`WebBasePage`, `MobileBasePage`, `BasePage`) you don't need anything else. If you do not extend either of these classes you will have to call `ObjectReporsitoryFactory.initObjects` in your constructor like this:

```
public ObjectRepositoryStandAloneWikipediaPageObject(WebDriver driver) {
    ObjectReporsitoryFactory.initObjects(driver, this);
}
```

A full-page object which uses the object repository should look like this:

```
public class ObjectRepoWikipediaMainPage extends WebBasePage {

    @ObjectRepositoryItem(id = "WIKI-MAIN-SEARCH")
    private WebElement searchBar;

    public static final String pageUrl = "https://www.wikipedia.org/";


    /**
     * C-tor. Initializes generic properties such as timeOut and pollingInterval
     *
     * @param driver a WebDriver object to store and use
     */
    public ObjectRepoWikipediaMainPage(WebDriver driver) {
        super(driver);
    }

```

```java

    @Override
    public boolean isOnPage() {
        return searchBar.isDisplayed();
    }


    public ObjectRepoWikipediaResultPage searchForTerm(String term){
        searchBar.sendKeys(term);
        new Actions(driver).sendKeys(Keys.ENTER).build().perform();
        return new ObjectRepoWikipediaResultPage(driver);
    }


    public ObjectRepoWikipediaMainPage gotoPage(){
        driver.get(pageUrl);
        return this;
    }
}
```

You can find this code, as well as a stand-alone page object here.

A test will look like this:

```java
 @Test
 @DisplayName("Search Wikipedia with Page Objects using the Object Repository")
 @Description("This test has a description")
 public void searchWikipediaWithPageObjectsRepository(VerisoftDriver driver) {

    String phraseToSearch = "Test Automation";
    ObjectRepoWikipediaMainPage wikipediaMainPage = new ObjectRepoWikipediaMainPage(driver);
    ObjectRepoWikipediaResultPage resultPage =
                                 wikipediaMainPage.gotoPage().searchForTerm(phraseToSearch);

    // Note!! Verisoft Assert
    Asserts.assertTrue(resultPage.isOnPage(), "Should be on the result page");
    Report.info("We reaeched the result page");

    Asserts.assertTrue(resultPage.getPageTitle().toLowerCase(Locale.ROOT)
            .contains(phraseToSearch.toLowerCase()),
                        "Title should containt the pharse " + phraseToSearch);
}
```

Repository file Location

The default path for the file is src/test/resources/objectRepository.json

To support another location or **multiple locations** for the object repository file in the VeriSoft framework, you can modify the implementation by following on of two options:

Option 1: Use `root.config.properties` Property

1. **Set a Property in the Configuration File**: Add a property in `root.config.properties` file (located in the test/resources folder) for the path to the desired object repository file. the property name is `object.repository.path`

```
object.repository.path=src/test/resources/myObjectRepository.json
```

Option 2: Pass the Path in the Constructor

- **Define a Custom Page Class**: Create a specialized page class that accepts an object repository path in its constructor.

```java
public class SpecificObjectRepoWikipediaMainPage extends WebBasePage {
    @ObjectRepositoryItem(id = "WIKI-MAIN-SEARCH")
    private WebElement searchBar;
    public static final String pageUrl = "https://www.wikipedia.org/";

    /**
     * Constructor that initializes the object repository with a specific path
     *
     * @param driver a WebDriver object to store and use
     * @param objectRepositoryPath the custom path to the object repository file
     */
    public SpecificObjectRepoWikipediaMainPage(WebDriver driver, String objectRepositoryPath) {
        super(driver, objectRepositoryPath);
    }

    @Override
    public boolean isOnPage() {
        return searchBar.isDisplayed();
    }

    public ObjectRepoWikipediaResultPage searchForTerm(String term) {
        searchBar.sendKeys(term);
        new Actions(driver).sendKeys(Keys.ENTER).build().perform();
        return new ObjectRepoWikipediaResultPage(driver);
    }

    public SpecificObjectRepoWikipediaMainPage gotoPage() {
        driver.get(pageUrl);
        return this;
    }
}

```

- **Use Custom Page Class in Your Tests**: Instantiate your page class with a specific object repository path in your test.

```java
@Test
@DisplayName("Search Wikipedia with Page Objects using Custom Object Repository Path")
public void searchWikipediaWithSpecificRepo(VerisoftDriver driver) {
    String phraseToSearch = "Test Automation";
    SpecificObjectRepoWikipediaMainPage wikipediaMainPage = new SpecificObjectRepoWikipediaMainPage(
        driver, "path/to/your/objectRepository.json"
    );
    ObjectRepoWikipediaResultPage resultPage = wikipediaMainPage.gotoPage().searchForTerm(phraseToSearch);
    Asserts.assertTrue(resultPage.isOnPage(), "Should be on the result page");
}

```

Default Object Repository Location

If no specific path is provided either via configuration or through the constructor, the framework falls back to the default path (`src/test/resources/objectRepository.json`).

Want to learn more about the mechanism? Read about it here.