

## Test Plan and Report

SavvyShoppers

SavvyDevelopers

Jul 24, 2023

### System Test Scenarios:

- A. As a user, I want to be able to track the item price from a url link.
- B. As a user, I want to be able to get the item name from the url link.
- C. As a user, I want to be able to parse information from an e-commerce website with the url link.

```
@pytest.mark.parametrize('store', stores)
def test_store(store):
    store_name = store['route']
    urls = store['urls']

    for url in urls:
        query_params = urlencode({'url': url})
        route = urljoin(f'/{store_name}', f'?{query_params}')

        # Update the route construction
        url_parts = urlsplit(route)
        route = url_parts.path + '?' + url_parts.query

        response = app.test_client().get(route)
        data = json.loads(response.data.decode('utf-8'))

        # Assertions
        if response.status_code == 400:
            print(url)
            time.sleep(3)

        assert response.status_code == 200
        assert isinstance(data, dict)
        assert isinstance(data.get('name'), str)
        assert isinstance(data.get('price'), float)
```

URL links for each route will be predefined and this function will test each link and assert that a name and price can be returned.

### Unit Tests:

Ebay failed the unit tests because when we accessed their page through our unit tests, it will return a captcha page. Therefore, there was no item data to parse. Walmart also failed the unit tests due to inconsistent results. This issue has not been resolved yet.

Last modified: 2023 07 24