

# QA Development Challenge

TEST APPROACH DOCUMENT

# FEATURE: Search weather in your city

***Owner***: Thong.Khuat

This document is a part of test strategy guideline set, for QA team. Authority belongs to OpenWheather development team. Responsibilities include:

* Owner: QA
* Reviewer: QA Lead, OpenWheather team tech lead

## Test Coverage

### In

* Search box as a top menu item – main page
* Search box functionality – main page
* Search result list – Weather in your city result page
* Search box – Weather in your city result page
* Search box functionality – Weather in your city result page
* Search foot note – Weather in your city result page

### Out

* Top menu – Main page
* Ad banner – Weather in your city result page
* Bottom menu – Weather in your city result page

## Objectives (Assumption)

This section summaries main factors of feature expectation and requirement from end-user perspective. The following ideas should be considered as guideline and acceptance criteria to pass the QA verification phase:

* Search weather function provides weather data reference service to Guest end-users with proposes:
  + Free data accessing to OpenWeather’s data to public community
  + Determine the demand of weather forecast data and the popularity of OpenWeather’s service in public community through the web-traffic statistic of this feature (data mining propose)
  + Provide weather data sample to potential pricing customers
* The feature should satisfy the following UX criteria:
  + Easy to be located from everywhere under OpenWeather’s domain
  + User can use and understand the queried info at popular knowledge level
  + Simple keyword format with standard name & code
  + Keyword input analytic must be strong enough for City & Country locating accuracy
  + Simple & friendly search guideline for exact search

## Test Approach

* 1. The manual effort of testing will base on mere Exploratory Testing Approach. Test executioner performs session based testing methodology using the starting point and coverage ideas from Objectives section. Test process must go through required Test Types & Level section and note down as Test Charter.
  2. The automation effort of testing will cover full functionality under Integration level including:
* Functionality of GET API for search service
* Functionality GUI for search box on main page – top menu
* Functionality GUI for search box on Weather in your city page
* Functionality GUI for result list on Weather in your city page

Test cases list providing notes:

* Using BDD format
* Test scenarios support API & front-end GUI perspective
* Cross browsers covering

## Test Types & Levels

* 1. Unit *– Not include in this exercise*
  2. Integration
     1. API Test for Weather in your city search service
     2. Functionality GUI Test for search result functional
  3. UAT
     1. Sanity GUI Test for main page & Weather in your city page
     2. End–To–End GUI Test for main page & Weather in your city page
     3. Load Test for search service *– Not include in this exercise*
     4. Stress Test for search service *– Not include in this exercise*
     5. Compatibility Test for front-end functional *– Not include mobile platform in this exercise*
     6. Usability Test for front-end UX look & feel

## Automation Scope

Automation framework details:

* Language: Python 3.
* Automation tool to apply:
  + API: requests - <https://pypi.org/project/requests/>
  + Front-end GUI: selenium - <https://pypi.org/project/selenium/>
* Test framework:
  + Test design: Pytest + Page Object Model + BDD
  + Test report: Allure
  + CI run: Jenkin
* Configuration:
  + MS Windows
  + Browsers support: Google Chrome, Mozilla Firefox