

Challenge QA Automation

Instrucciones:

- El challenge se compone de 2 partes, por un lado, realizar un test a nivel API (Backend) y por otro, un test a nivel Frontend.
- El lenguaje de programación puede ser Java o Kotlin.
- Para el caso de frontend, utilizar Selenium WebDriver.
- Preferentemente utilizar: Maven, Junit, TestNG, HttpClient, HttpURLConnection, Rest Assured.
- Se permitirá solo el uso de herramientas de tipo openSource.
- Se deberá proveer todo lo necesario para la ejecución del test como comandos, doc de referencia, consideraciones, etc.
- Luego de realizar los test, subir el código a un gitlab o github personal público.
- (Opcional) - Generar reporte con resultado de ejecución de los tests.

Test Frontend:

- Realizar un único test donde se realicen los pasos abajo mencionados y se incluya en dicho test las aserciones indicadas en el punto 6.

Pasos:

- 1- Generar una clase, con un método de test que instancie un chromedriver.
- 2- Ingrese a la home de Frávega (www.fravega.com).
- 3- Busque "Heladera".
- 4- Filtre por "Heladeras" desde los filtros de la sección izquierda de la página.
- 5- Filtre por "Samsung" desde los filtros de la sección izquierda de la página.

Assert:

- 6- Guarde la grilla de resultados como lista de WebElement y aserte que:
 - Cada uno de los elementos contenga "samsung" en su title.
 - La cantidad de elementos de la lista coincida con los resultados mostrando por el frontend.
 - En el breadcrumb de la página (atributo name="breadcrumb") se visualice "Heladeras con Frezzer"

El test debe heredar de una clase base (TestBase) donde deben estar los annotation @BeforeTest y @AfterTest.
Es deseable utilizar el patrón "Page Object" para el modelado de las páginas.

Test Backend:

- Realizar un único test donde se realicen los pasos abajo mencionados y se incluya en dicho test las aserciones indicadas en el punto 4.

Pasos:

- 1- Obtener una lista de cervecerías que contengan el texto "lagunitas" en su nombre.
Para ello, se debe ejecutar el siguiente servicio, indicando el texto a buscar en el queryParams "query".

GET - <https://api.openbrewerydb.org/breweries/autocomplete>

- 2- De la lista de resultados del punto 1, tomar aquellos que contengan en la key "name", el valor "Lagunitas Brewing Co".

- 3- A través del siguiente servicio, obtener el detalle de cada cervecería de la lista del punto 2 y tomar solo el que contenga "state" = "California".

GET - <https://api.openbrewerydb.org/breweries/{id}>

Assert:

- 4- Sobre la cervecería resultante, assertar lo siguiente:

"id" = 761
"name" = "Lagunitas Brewing Co"
"street" = "1280 N McDowell Blvd"
"phone" = "7077694495"

Seccion Funcional:

Instrucciones:

Dada la siguiente especificación funcional, generar los casos de prueba que verifiquen el buen funcionamiento del requerimiento solicitado para un sitio web.

- Realizar un máximo de 10 casos de prueba por funcionalidad.
- Agregar todos los campos para el armado de los casos que considere necesario.
- Agregar si lo desea o requiere datos de prueba.

Search.

Se modificará el motor de búsqueda e indexación del sitio.

Se debe verificar que el resultado de búsqueda responda a la keyword (palabra) ingresada.

El buscador por keyword posee las siguientes características:

- La keyword a buscar, puede estar tanto en título como en descripción de los productos.
- El motor de búsqueda posee diccionario de sinónimos (ejemplo -> si busco "pley 4", realiza la búsqueda de "play 4")
- No es posible buscar sin ingresar una keyword
- Las keywords a buscar deben poseer un mínimo de 3 caracteres
- La keyword buscada se visualiza en el resultado de búsqueda
- Al lado del textbox para ingresar la keyword, se incluye un botón "buscar".