

## TECHNICAL TEST BE

### V2.0.0

November, 2021

### PRERREQUISITOS

- Tener Git instalado
- Tener Node.js Instalado (v 12.18.1 o superior)
- Clonar el repositorio con la url que has recibido vía email
- Desde la consola, Ve a la carpeta src y ejecuta el comando: npm install
- Desde la consola, ejecuta el comando: npm run test:watch, and happy code! :)

### RESTFUL FEATURE

Necesitamos que construyas una función en el archivo **src/test/handler.js** llamada "search", debe recibir un objeto como parámetro con los filtros especificados abajo. Esa función debe hacer un request de tipo GET a esta url: <https://hiring.condorlabs.io/api/countries/all> con el fin de obtener la data que vas a necesitar usar para completar este test.

#### 1. Aplicar: Paginación, Filtros, Ordenamiento

- 1.1. Request para obtener la data desde: <https://hiring.condorlabs.io/api/countries/all>
- 1.2. Filtrar por population, rango entre 1000 and 1500 personas (from: 1000, to: 1500)
- 1.3. Filtrar por región Asia (region: Asia)
- 1.4. Ordenar ascendentemente por name (sort: { name: asc | desc })
- 1.5. Ordenar descendentemente por name
- 1.6. Paginación (PageNumber: 1, PageSize: 10)

El parámetro Filters debe incluir las propiedades exactamente como están especificadas a continuación:

- pageNumber
- pageSize
- from
- to
- region
- sort

#### 2. API endpoint: Crear un endpoint para resolver el search

- 2.1. Crea una ruta bajo el path **/test** donde uses la función search creada en el paso 1.
- 2.2. El endpoint debería ser capaz de recibir los filtros de entrada, con los siguientes valores: (PageNumber: 1, PageSize: 10, from: 1000, to: 1500, region: Asia, sort: { name: desc }).

### UNIT TESTING

En el repositorio puedes encontrar unos unit test precontruidos que al inicio del test están en rojo, tu misión es ponerlos todos en verde agregando el código que satisface los requerimientos, tú NO debes modificarlos ya que están creados con el objetivo de evaluar las funcionalidades requeridas. Abre una consola y ejecuta el comando debajo para mantener los test corriendo mientras escribes el código. npm run test:watch O ejecutarlos cuando hayas finalizado la prueba. npm run test

Abre una consola y ejecuta el comando debajo para mantener los test corriendo mientras escribes el código.

```
npm run test:watch
```

O ejecutarlos cuando hayas finalizado la prueba.

```
npm run test
```

#### EXAMPLE:

En este caso, puedes ver un ejemplo de cómo jest muestra la interfaz mientras corre los tests.

```
PASS src/components/__tests__/CustomTaskListComponent.spec.jsx
CustomTaskListComponent
  ✓ should render demo component (16ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.027s
Ran all test suites.

Watch Usage
  > Press f to run only failed tests.
  > Press o to only run tests related to changed files.
  > Press q to quit watch mode.
  > Press p to filter by a filename regex pattern.
  > Press t to filter by a test name regex pattern.
  > Press Enter to trigger a test run.

shagman@shagman-next:/mnt/DAS_MAIN/PRFT/plugin-new$ npm run test
```