

Desarrollo de páginas con Angular

Módulo 4 - Lección 3 - Lectura 01: Buenas prácticas de testing unitario

* citamos a la documentación oficial de Jasmine en Github

Suites: describe tus pruebas con “describe”

La función describe es para agrupar especificaciones relacionadas, normalmente cada archivo de prueba tiene uno en el nivel superior. El parámetro de cadena es para nombrar la colección de especificaciones y se concatenará con las especificaciones para hacer el nombre completo de una especificación. Esto ayuda a encontrar especificaciones en una gran suite. Si los nombra bien, sus especificaciones se leen como oraciones completas en el estilo tradicional de BDD.

Especificaciones

Las especificaciones se definen llamando a la función de Jasmine global llamada “it”, que, al igual que “describe”, toma una cadena y una función. La cadena es el título de la especificación y la función es la especificación o prueba. Una especificación contiene una o más expectativas que prueban el estado del código. Una expectativa en Jasmine es una afirmación que es verdadera o falsa. Una especificación con todas las expectativas verdaderas es una especificación que se encuentra sana es decir que el código que estamos probando está sano. Una especificación con una o más falsas expectativas es una especificación que falla.

Son solo funciones

Ya que describe e it son funciones, pueden contener cualquier código ejecutable necesario para implementar la prueba. Se aplican reglas de alcance de variables y código de JavaScript, por lo que las variables declaradas en un describe están disponibles para cualquier bloque dentro del conjunto.

Expectations

Las expectativas se crean con la función expect, que toma un valor, llamado real. Está encadenado con una función Matcher, que toma el valor esperado.

Matchers

Cada comparador implementa una comparación booleana entre el valor real y el valor esperado. Es responsable de informar a Jasmine si la expectativa es verdadera o falsa. Jasmine entonces pasará o fallará la especificación.

Cualquier comparador puede evaluar una afirmación negativa encadenando la llamada a esperar con un no antes de llamar al comparador.

Jasmine tiene un amplio conjunto de Matchers incluidos, puede encontrar la lista completa en los documentos del API. También existe la posibilidad de escribir Matchers personalizados para cuando el dominio de un proyecto solicite afirmaciones específicas que no están incluidas en Jasmine.

Configuración y desmontaje

Para ayudar a un conjunto de pruebas a cumplir el principio de buen diseño conocido como DRY (del inglés do not repeat yourself, es decir, el principio de diseño que nos aconseja que no repitamos código) cualquier configuración duplicada y código de desmontaje podemos ubicarlo en funciones especializadas, conocidos como eventos, y que Jasmine proporciona dichas funciones globales llamadas: `beforeEach`, `afterEach`, `beforeAll` y `afterAll`.

```
describe ("Una suite con alguna configuración compartida", función () {  
  var foo = 0;
```

`//Como su nombre lo indica, la función beforeEach se llama una vez antes de cada especificación en el describe en la que se llama`

```
    beforeEach (function () {  
      foo += 1;  
    });
```

`//y la función afterEach se llama una vez después de cada especificación.`

```
    afterEach (function () {  
      foo = 0;  
    });
```

`//La función beforeAll se llama solo una vez antes de que todas las especificaciones descritas se ejecuten`

```
    beforeAll (function () {  
      foo = 1;  
    });
```

`//y la función afterAll se llama después de que todas las especificaciones terminen`

```
    afterAll (function () {  
      foo = 0;  
    });  
  });
```

`beforeAll` y `afterAll` pueden usarse para acelerar las suites de prueba con una configuración y desmontaje costosos.