

```
expect.extend({
  fullName(actual, fullName) {
    const { first, last } = actual;
    const actualName = `${first} ${last}`;

    const [expectedFirst, expectedLast] = fullName.split(" ");
    const expectedName = `${expectedFirst} ${expectedLast}`;

    const pass = actualName === expectedName;

    const message = pass
      ? () => `Name should not be ${actualName}`
      : () => `Name should be ${expectedName}, but is ${actualName}`;

    return { actual: actual, message, pass };
  }
});
```

Not Fresh

```
●test("carlton banks", async () => {  
  const response = await BelAir.request("carlton");  
  
  expect(response.data.user).toEqual({  
    id: "carlton",  
    role: "student",  
    name: { first: "Carlton", last: "Banks" },  
    grades: {  
      literature: "A",  
      math: "A",  
      gym: "A",  
      health: "A",  
      chemistry: "A"  
    }  
  });  
});
```