

```
expect.extend({
  toHaveFullName(actual, expectedName) {
    const { first, last } = actual.data.user.name;
    const actualName = `${first} ${last}`;

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

    const pass = actualName === expectedName;

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

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

```
●test("uncle phil", async () => {  
  const response = await BelAir.request("philip");  
  
  expect(response.data.user.name.first).toBe("Uncle");  
  expect(response.data.user.name.last).toBe("Phil");  
  expect(response.data.user.role).toBe("parent");  
});
```