

# Not Fresh

```
●test("ashley banks", async () => {
  const response = await BelAir.request("ashley");

  response.data.user.grades =
    Object.keys(response.data.user.grades)
      .map(course => BelAir.computeGpa(response.data.user.grades[course]))
      .reduce((total, grade) => (total += grade)) /
    Object.keys(response.data.user.grades).length;

  const { first, last } = response.data.user.name;
  response.data.user.name = `${first} ${last}`;

  expect(response.data.user).toMatchSnapshot();
});
```

# Not Fresh

**FRESH**

```
●test("ashley banks", async () => {
  const response = await BelAir.request("ashley");

  response.data.user.grades =
    Object.keys(response.data.user.grades)
      .map(course => BelAir.computeGpa(response.data.user.grades[course]))
      .reduce((total, grade) => (total += grade)) /
    Object.keys(response.data.user.grades).length;

  const { first, last } = response.data.user.name;
  response.data.user.name = `${first} ${last}`;

  expect(response.data.user).toMatchSnapshot();
});
```

```
●test("ashley banks", async () => {
  const response = await BelAir.request("ashley");

  expect(response).toMatchSnapshot("Ashley Banks", 3.0);
});
```