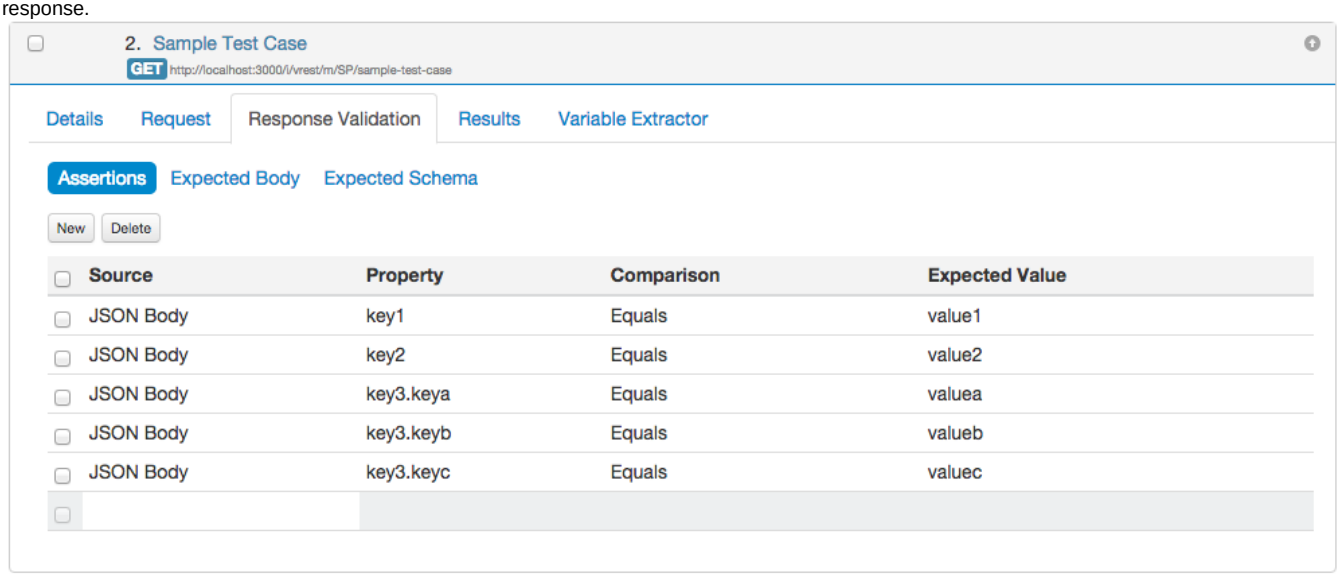
Benefits of using Response Validators

If you understand well, how Default Validator works, then in majority of cases, you don't need to define any other JSON Body assertions. You need to define the Expected Body if you choose Default Validator in the Text Body Assertion. Suppose, I want to validate an API which returns the following response. We are taking a simple response here, how you can validate the complex responses through Default Validators are covered later in this article.

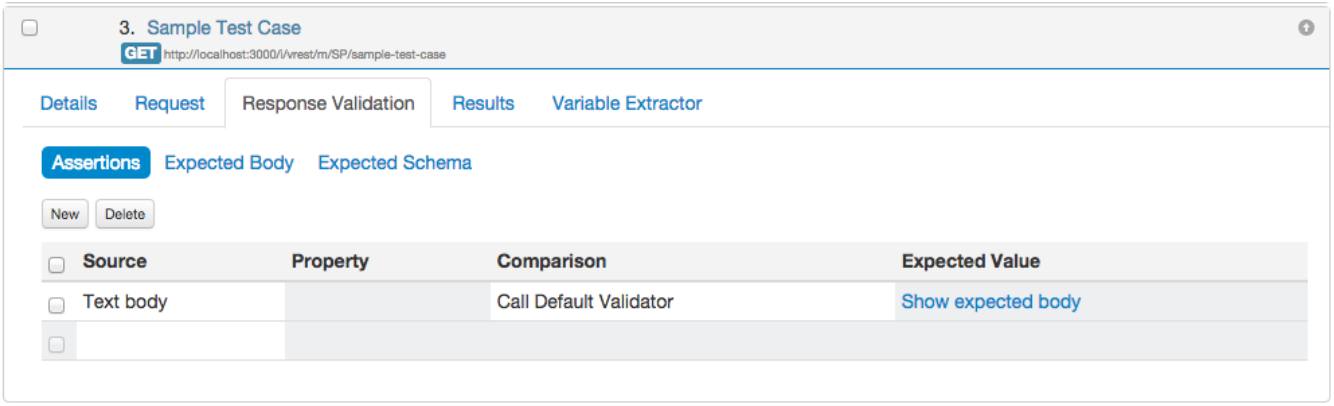
Sample Test Case Response

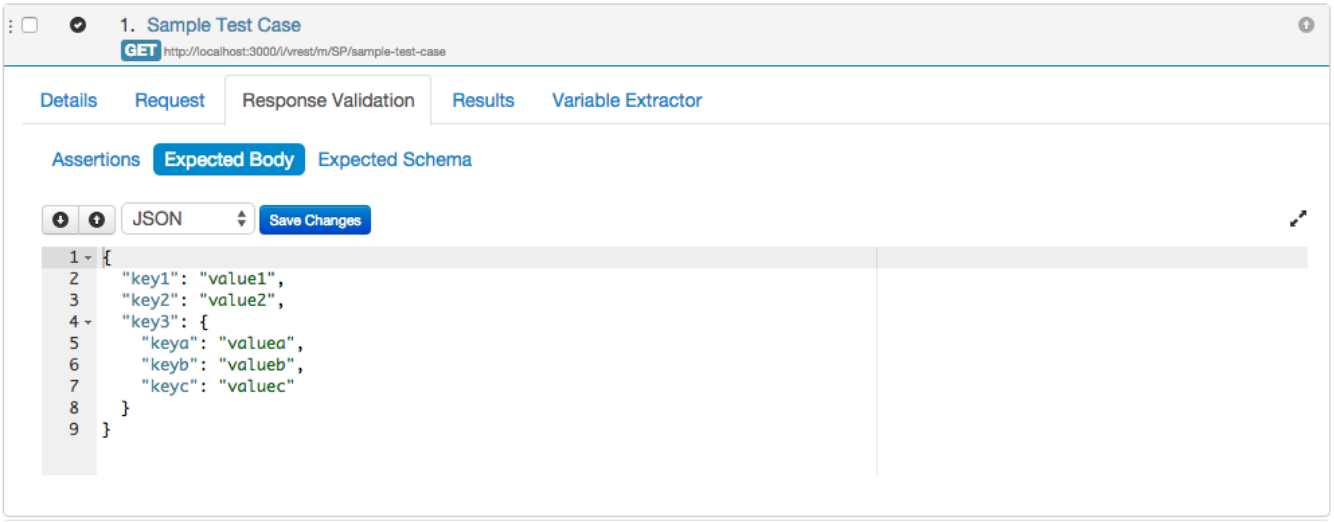
```
{
  "key1": "value1",
  "key2": "value2",
  "key3": {
    "keya": "valuea",
    "keyb": "valueb",
    "keyc": "valuec"
  }
}
```

Now, if you don't want to validate the response with Default Validator, but instead you have defined the following JSON Body assertions to validate the response.



With Default Validator, you can convert these multiple JSON Body assertions into one assertion, but you need to set the Expected Body in this case.





So, benefits of using this approach are:

1. **Easier to Define**
   Even if you don't know the expected response body in advance then just set the empty object ({}). When you execute this test case, you will have an option to copy actual response body to expected response body.
2. **Easier to maintain**
   If your test case response changes in future then you can use function "Copy Actual to Expected" to update the expected body after reviewing the Diff Report with a single click.
3. **Easier to visualise changes**
   Diff report is more compact and pretty way to visualise the test case failures.