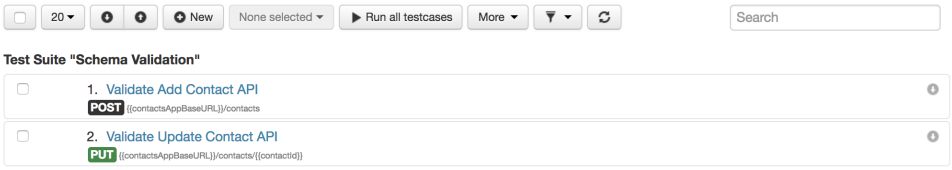
Schema Validation

Schema Validation is also a powerful feature of vREST. You may validate your API test case's responses against JSON schema. You may define this JSON schema in

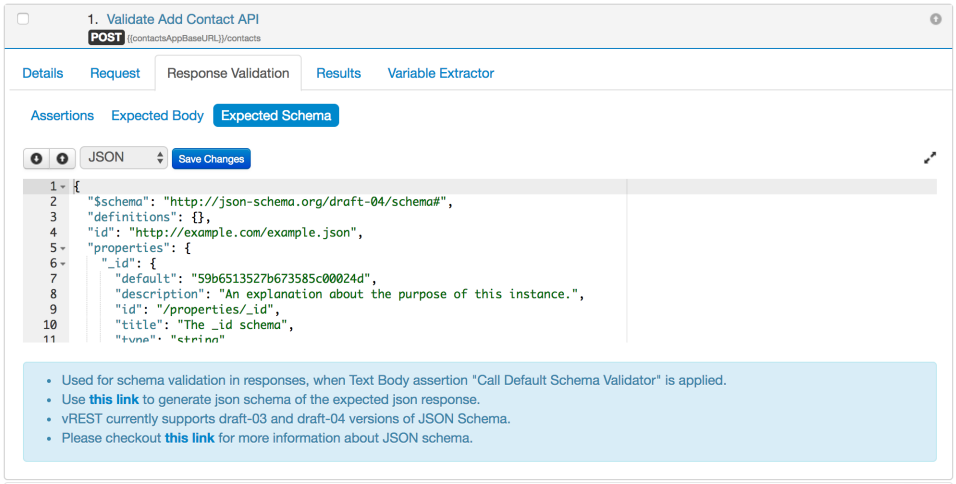
1. **Expected Schema** sub-tab under **Response Validation** tab of each test case
2. And in **Project Configuration** >> **Schemas section**

Now we will see, how you may validate your test cases using JSON Schema definition.

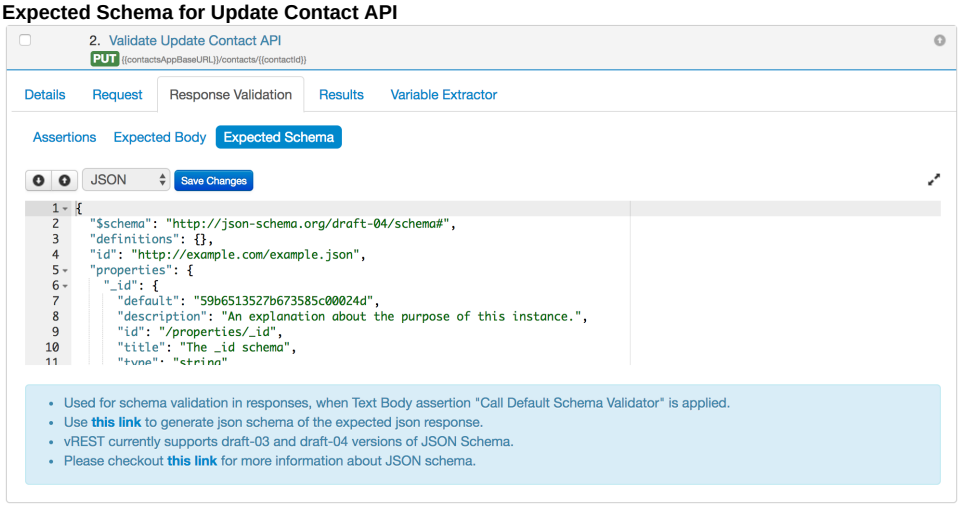
1. **Create test cases for your test application**
   a. Suppose we have two test cases for our example test application (Contacts Application).
   b. One test case is validating the Add Contact API and another test case is validating the Update Contact API.
   c. A sample screenshot is available below:



2. **Validate those test cases via JSON Schema by defining the schema in Expected Schema Tab**
   a. Now you may validate these test cases by defining the expected JSON schema for the test cases responses.
   b. You can define the JSON schema in Expected Schema sub-tab under Response Validation tab as shown below:
   Expected Schema for Add Contact API



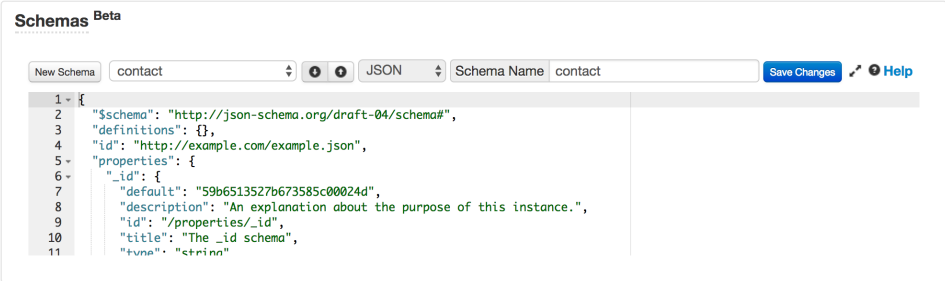
   Expected Schema for Update Contact API



   Note: Both APIs consumes the same JSON schema as the response structure of both APIs are same.

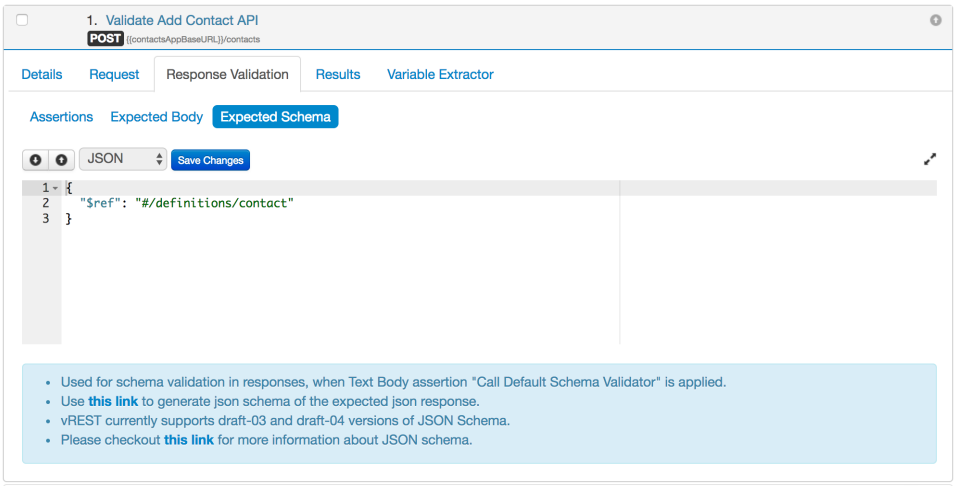
   Now if you execute these two test cases, then both will pass. Now if we have hundreds of such test cases, then we will be defining the same schema in each test case. To overcome this issue we may define the schema globally in Project Configuration tab and re-use the same schema in our test cases.
3. **Re-use the common schema by defining them in Project Configuration >> Schemas Section**
   In Project Configuration >> Schema Section, you may define all your schemas related to your APIs. Then you may use them in your test cases for response validation. In your test cases, you just need to refer the global definitions defined in Schemas section. This section is also useful when you import test cases via swagger definitions.

Let's define our Contact schema in **Project Configuration** >> **Schemas Section** as shown below:



Now, we may use this schema in our test cases by referencing it in Expected Schema tab of our test cases as shown below:

Updated Expected Schema for Add Contact API:



Updated Expected Schema for Update Contact API: