## ## Introduction

This guide will teach you how to generate variants for a product. In this example, we will be generating variants from two unique Specs, each with different SpecOptions. Generating variants creates all available combinations for a product.

## ## Prerequisites

* [Create and Assign Product within a Buyer Organization](https://documentation.ordercloud.io/use-case-guides/product-catalog-management/create-and-assign-a-product)
* [Make An Assignment](https://documentation.ordercloud.io/use-case-guides/buyer-and-seller-organization-management/make-an-assignment)

## 1. Create New Spec and Assign to Product

Specs are used to define the attributes that are configurable on a product. An example of a Spec would be the different colors available for a TShirt. If you want to be able to generate variants for the product spec, you must ensure that `"DefinesVariant": true`.

<div class="api-reference">API Reference: [Create New Spec](https://documentation.ordercloud.io/api-reference#Specs\_Create)</div>

```

POST https://api.ordercloud.io/v1/specs HTTP/1.1  
Authentication: Bearer put\_access\_token\_here  
Content-Type: application/json; charset=UTF-8  
  
{

"ID": "Color",

"Name": "Color",,

"Required": true,

"DefaultOptionID": "",

"DefinesVariant": true,

}

```  
<div class="note">Once the Spec has SpecOptions, we can also define a `DefaultOptionID`, which will display a pre-set Spec for a product.</div>

## 2. Assign to Product

Once you have created the Spec, you can assign it to the product:<br/>

<div class="api-reference">API Reference: [Save Product Assignment](<https://documentation.ordercloud.io/api-reference#Specs_SaveProductAssignment>)</div>

```

POST https://api.ordercloud.io/v1/specs/productassignments HTTP/1.1  
Authentication: Bearer put\_access\_token\_here  
Content-Type: application/json; charset=UTF-8  
  
{

"SpecID": "Color",

"ProductID": "TShirt",

"DefaultValue": "",

"DefaultOptionID": ""

}

```

## 3. Create SpecOptions

Next, we will create the SpecOptions associated with the Spec. This step generates the values of a Spec that are available on a product. Continuing the previous example, one of our Specs for the TShirt could be the Color, so the SpecOptions would be Red, Blue, etc.

When creating SpecOptions, you will first need to define the `SpecID` (in our example, Color) in order to map each value to the correct Spec.

<div class="api-reference">API Reference: [Create Option](https://documentation.ordercloud.io/api-reference#Specs\_CreateOption)</div>

```

POST https://api.ordercloud.io/v1/specs/{specID}/options HTTP/1.1  
Authentication: Bearer put\_access\_token\_here  
Content-Type: application/json; charset=UTF-8  
  
{

"ID": "001",

"Value": "red",

"ListOrder": 0,

"IsOpenText": false,

"PriceMarkupType": null,

"PriceMarkup": null,

"xp": null

}

```

## 4. Generate Variants

Once your Spec and SpecOptions have been created, the next step is to generate the variants. OrderCloud automatically composes all possible combinations available for a product based on your Spec values using the `GenerateVariants` call. Our TShirt example contains two Specs: Size and Color; therefore, the variants that will generate would be: Small-Red, Small-Blue, Medium-Red, Medium-Blue, Large-Red, Large-Blue, etc.

<div class="api-reference">API Reference: [Generate Variants](https://documentation.ordercloud.io/api-reference#Products\_GenerateVariants)</div>

![GenerateVariants](/assets/images/docs-guides/use-case-guides/GenerateVariants.gif)

## 5. List Variants

The variants that were generated can be viewed by calling `ListVariants`, which will return all variants of the product. Our TShirt example has 2 Specs (Size and Color) with 2 SpecOptions for Color and 3 SpecOptions for Size; therefore, the `ListVariants` call would return 6 Variants.

<div class="api-reference">API Reference: [List Variants](https://documentation.ordercloud.io/api-reference#Products\_ListVariants)</div>

![ListVariants](/assets/images/docs-guides/use-case-guides/ListVariants.gif)

## ## What Ifs

<strong>What if I want to enable custom text for a Spec attribute?</strong><br/>  
Variants can also allow a user to enter free text for an attribute on a product. Using business cards as an example, users could enter their personal information into a Spec field by setting the `AllowOpenText` field to `true`.

<div class="note">An additional note when creating the product/Spec assignment, if you were to enter a `DefaultOptionID` this will override the `DefaultOptionID`, if it exists, on the Spec.</div>

## ## Conclusion

Variants are used to help customers view and compare the different combinations available on a given product. You should now have an understanding for how to generate a browsing experience that allows users to specify product variations.