

# Compatibility of CPW app with shopify theme 1.0.0 ( Need to apply given steps wise changes )

27/04/2023

Shraddha Rewatkar

- **Shopify theme 1.0.0 is compatible with Variant Level Rule and also Offer list level Rules**

Follow below steps

- 1) **Step 1 : Open to the Shopify store = > Select shopify theme 1 => Go to the Edit code => Select Asset folder => Create new js file ( oscp-cpw.js ) => Add Below code in file**

```
document.addEventListener('cpw:init', function(evt) {
  var oscpProductTemplate = {
    settings: {},
    init: function(){
      var $this = this;
      if(typeof oscpCPWAppConfig == 'undefined'){

        eval(jQuery(document).find('script[type="text/oscpAppSetting"]').eq(0).text(
        ));
      }

      //$this.settings = (typeof oscpCPWAppConfig == 'undefined') ? {} :
      JSON.parse(oscpCPWAppConfig);
      $this.settings = (typeof oscpCPWAppConfig == 'undefined') ? {} :
      oscpCPWAppConfig;

      console.log($this.settings)
      jQuery(document).on('change', 'input[name="id"]', function(e){
        if(typeof $this.settings.timer !== 'undefined'){
          clearTimeout($this.settings.timer);
        }

        $this.settings.timer = setTimeout(function(){
```

```

        $this.setProductTemplate();
    }, 200);
    })
    $this.setProductTemplate();
},
setProductTemplate: function(){
console.log('setProductTemplate === ', this.settings)
    if(this.settings.priceGrid[0].hasOwnProperty('rules')){
        this.variantLevelTemplate();
    }else{
        this.offerLevelTemplate();
    }
},
variantLevelTemplate: function(){
    $('#oscpPriceGrid').remove();
    var url = new URL(window.location.href);
    var variantId = url.searchParams.get("variant");

    for(var j=0; j<this.settings.priceGrid.length; j++){
        if((this.settings.priceGrid[j].id == variantId) &&
this.settings.priceGrid[j].rules.length){
            var qtybreakHtml = "<table
id='oscpPriceGrid'><tr><th>Qty</th><th>Price</th></tr>";
            for(i=0; i<this.settings.priceGrid[j].rules.length; i++){

                var rowPrice = this.settings.priceGrid[j].price;
                var rowrules = this.settings.priceGrid[j].rules[i];
                var price =
this.getLinItemPrice((rowPrice*100), rowrules , 1);
                //var price = [10,20,30];
                var Pricerules = (rowrules.type == 'discount') ?
rowrules.value + " "+" OFF" : (this.setMoneyFormat(rowrules.value*100) + "
each" );

                qtybreakHtml += "<tr><td>"+ "Buy
"+this.settings.priceGrid[j].rules[i].qty+"</td><td>"+Pricerules+"</td></tr>";
            }
            qtybreakHtml += "</table>";

            //$('#oscpPriceGridContainer').append(qtybreakHtml);

```

```

        $('form[action*="/cart/add"]').eq(0).append(qtybreakHtml);
    }
}
},
offerLevelTemplate: function(){
    var qtybreakHtml = "<table
id='oscpPriceGrid'><tr><th>Qty</th><th>Price</th></tr>";

    for(var j=0; j<this.settings.priceGrid.length; j++){
        var Pricerules = (this.settings.priceGrid[j].type == 'discount') ?
this.settings.priceGrid[j].value + " "+" % OFF" :
(this.setMoneyFormat(this.settings.priceGrid[j].value*100) + "  each" );
        qtybreakHtml += "<tr><td>"+ "Buy
"+this.settings.priceGrid[j].qty+"</td><td>"+Pricerules+"</td></tr>";

    }

    qtybreakHtml += "</table>";

    $('form[action*="/cart/add"]').eq(0).append(qtybreakHtml);

},
getLineItemPrice: function(price, rule, rules, qty){ // Return calculated (line
item price, line item total price and saving amount)
    price = parseInt(price);
    var rowValue = (rule.type == 'discount') ? parseFloat(rule.value) :
parseFloat(rule.value)*100;
    if(rule.type == 'discount'){
        var calulatedPrice = (price-(price*rowValue/100));
        var save = rowValue.toFixed(2)+'%'
    }else{
        var calulatedPrice = rowValue;
        var save = (100*(price - rowValue)/price);
        save = save.toFixed(2);
        save = save+'%';
    }
    var calulatedTotalPrice = (calulatedPrice*qty);
    return [calulatedPrice, calulatedTotalPrice, save, rules];
},

```

```

        setMoneyFormat: function(amount){           // Provision to show price
with proper currency and template
            amount = amount/100;
            amount = amount.toFixed(2);
            return this.settings.priceTemplate.replace("0", amount);
        },
    };
    oscpProductTemplate.init();
});

```

## 2) Step 2 : Include this new created js file in theme.liquid file

Before closing tag of </head>

```
<head>
```

```
<script src="{{ 'oscp-cpw.js' | asset_url }}" defer="defer"></script>
```

```
</head>
```

## 3) Step 3 : Element Mapping in CPW App Extension with cart page and mini cart page and Checkout page

admin.shopify.com/store/demo-custom-pricing-wholesale/t

Dawn Live ...

### App embeds

Sections

Search app embeds

Custom Price Grid  
Custom Pricing Wholes...

Line item price total **cart page product price total**

.price--end

Line item price total : Add ID or Class of Original price

Line item price **product price in cart page**

div.product-option

Line item price : Click on Discounted price of the product on the cart page and select ID or Class

Cart Total **Subtotal on cart page**

.totals\_\_subtotal-value

Cart total : Click on Subtotal price (after discount) on the cart page and select ID or Class

Cart container **cart page main div container**

.cart-item

Cart Container Element : ID or Class

Cart Quantity Input **cart input quantity**

.quantity\_\_input

Quantity input :

Checkout Button

Demo - Custom

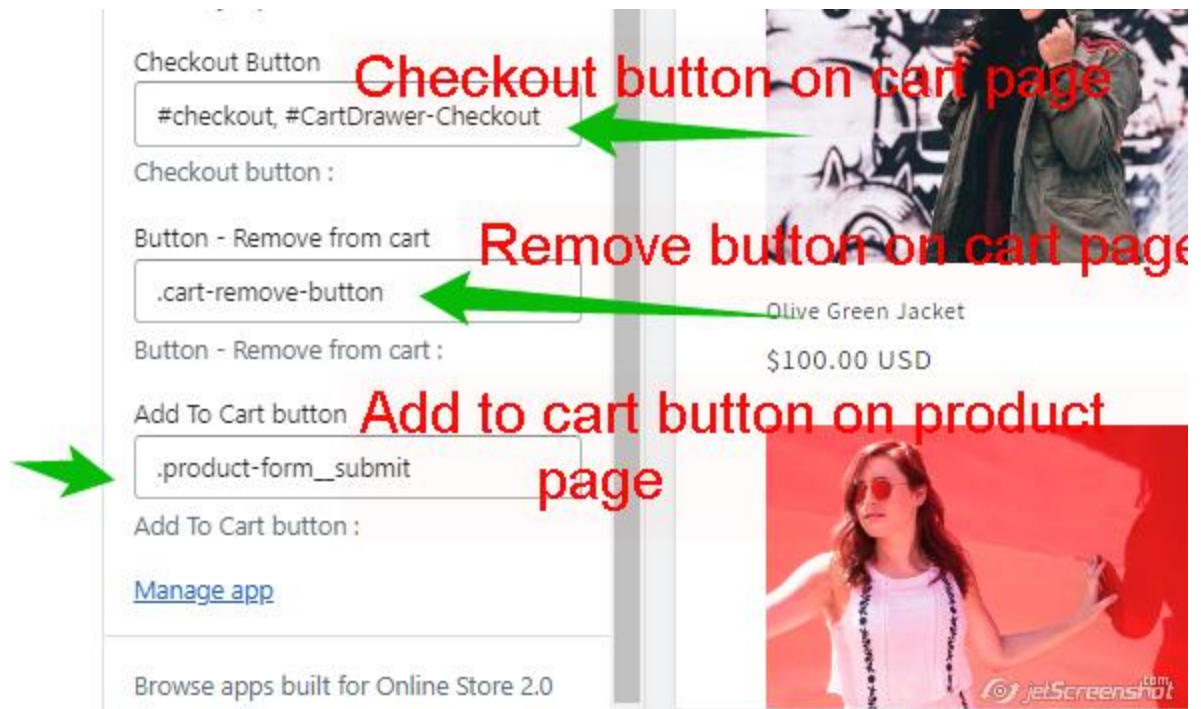
Selected

Features prod

Olive Green Jacket

\$100.00 USD

jetScreenshots.com



### Notes : Which points handle to resolve issue

- 1) Created new file (oscp-cpw.js) in theme asset directory
- 2) include this new created js file in theme.liquid file  

```
<script src="{{ 'oscp-cpw.js' | asset_url }}" defer="defer"></script>
```
- 3) In Above new file we created to handle event listener ("cpw:init")
- 4) This event will trigger from our app (CPW) and expose app data into theme
- 5) In this new file we handle both cases (variant level pricing, offer level pricing). but variant level pricing not tested yet.
- 6) To map elements we used an app provided backend interface. but there was some issue due to javascript based template with mini cart. and it was create difficulties to update values in minicart
- 7) To solve this issue We implement another event listener in app (cpw:cart-init)
- 8) This event (cpw:cart-init) will execute the cart refresh method. and executable by theme level script also.
- 9) After the mini cart drawer opened I triggered/ dispatched the "cpw:cart-init" event from the theme script to update values in minicart.  
**document.dispatchEvent(new CustomEvent('cpw:cart-init'));**