# (魔改造)Vue.jsでのコード

## Vueコンポーネント定義

```
import Vue from 'vue'
import { Component, Prop } from 'vue-property-decorator'
import { filter, emitter } from '../stores'
import template from '../templates/FilterPallet.pug'

@Component(template({}))
export default class FilterPallet extends Vue {
  @Prop() type: string
  selected: number[] = []
  properties = filter.definitions[this.type]
    .sort( compareFn: (a, b) => a.priority - b.priority)

  bg(id: number) {
    const property = this.properties.find( predicate: param => id === param.id)

    return !property
      ? null
      : property.hex_code
        ? { backgroundColor: `#${property.hex_code}` }
        : { backgroundImage: `url(/images/pattern-icons/${property.slug}.png)` }
  }

  setQuery(): void {
    const condition = filter.condition.filter( callbackfn: query => query.type !== this.type)
    filter.condition = this.selected.length
      ? condition.concat({ type: this.type, ids: this.selected })
      : condition

    emitter.$emit( event: 'filter-change')
  }

  handleForceUpdate(): void {
    const condition = filter.condition.find( predicate: query => query.type === this.type)
    const ids = condition ? condition.ids : []

    this.selected.splice( start: 0, this.selected.length, ...ids)
  }

  created(): void {
    emitter.$on( event: 'filter-force-update', this.handleForceUpdate)
    this.handleForceUpdate()
  }
}
```

## Vueテンプレート定義 (Pug/Jade)

```
ul.color-handle-index
  li.color-handle-element(
    v-for='property in properties',
    :key='property.id',)
    label
      input.checkbox(
        type='checkbox',
        :value='property.id',
        v-model='selected',
        @change='setQuery',)
      span.checkbox-icon(:style='bg(property.id)')
```

# (魔改造)Vue.jsでのコード

## Vueテンプレート定義 (Pug/Jade)

```
.no-product-items(v-if='productsNotFound')
  p 商品が見つかりませんでした。
    br
    |検索条件を変更して再検索してください
  button.ui-btn(@click.prevent.stop='undo') 直前の検索条件に戻る
product-group(
  v-for='group, index in groups',
  :key='group.id',
  :name='group.name',
  :price='group.price',
  :items='group.items',)
```

## Vueテンプレート定義 (Pug/Jade)

```
export default class App extends Vue {
  items: Item[] = filter.apply()
  visibleLength = 8
  modal = false
  attach = false
  mobile = window.innerWidth <= 750

  get productsNotFound(): boolean { return !this.groups.length }

  get groups(): {name: string, price: string | number, items: Item[]}[] {
    const visible = this.items.filter( callbackfn: (item, index) => index < this.visibleLength)

    return filter.definitions.lineup
      .sort( compareFn: (a, b) => a.priority - b.priority)
      .dedupe( predicate: (a, b) => a.name === b.name)
      .map( callbackfn: refiner => {
        return {
          name: refiner.name,
          price: refiner.price,
          items: visible.filter( callbackfn: product => product.refiners.lineup[0].name === refiner.name)
        }})
      .filter( callbackfn: lineup => lineup.items.length)
  }

  undo(): void {
    logger.undo()
    filter.condition = logger.current

    this.items.splice( start: 0, this.items.length, ...filter.apply())
    emitter.$emit( event: 'filter-force-update')

    if (this.mobile) this.modal = true
  }
```