Folder js

4 printable files

(file list disabled)

js/ProductTypesView.js

```
class ProductTypesView {

    constructor(url, outputListId, productListView) {
        this.url = url
        this.$outputList = $('#' + outputListId)
        this.productListView = productListView
        this.typeItems = []
    }

    drawListGroup() {
        $.ajax(this.url)
            .done((response) => {
                for (const item of response) {
                    this.$outputList.append(this.createListEntry(item))
                }
                this.typeItems[0].trigger("click")
            })
            .fail((error) => {
                console.log("Ajax Call failed: ", error)
            })
    }

    createListEntry(productTypesItem) {
        let $listEntry = $("<li>" + productTypesItem.productType + "</li>")
        $listEntry.addClass("list-group-item")
        $listEntry.addClass("cursor-pointer")
        $listEntry.addClass("list-group-item-action")

        $listEntry.on('click', () => {
            this.$outputList.find('.list-group-item').removeClass("active")
            $listEntry.addClass("active")

            this.productListView.setUrl(productTypesItem.url)
            this.productListView.loadProducts()
        })
        this.typeItems.push($listEntry)
        return $listEntry
    }
}
```

js/ProductView.js

```
class ProductView {

    constructor(product) {
        this.product = product
```

```
      }

      createView(){
          const $divCol = $('<div class="col"></div>')
          const $divCard = $('<div class="card text-smaller"></div>')
          const $img = this.getImage()
          const $divCardBody = $('<div class="card-body"></div>')
          const $h5CardTitle = $('<h5 class="card-title">'+this.product.name.substring(0,10)+'</h5>')
          const $divCardText = $('<p class="card-text">'+this.product.name+'</p>')

          $divCard.append($img)
          $divCardBody.append($h5CardTitle)
          $divCardBody.append($divCardText)
          $divCard.append($divCardBody)

          return $divCol.append($divCard)
      }

      getImage() {
          return $("<img src='img/"+this.product.id+".png' />")
      }
  }
```

js/ProductsListView.js

```
  class ProductsListView {
      constructor(productsListId) {
          this.productListUrl = null
          this.$outputList = $('#' + productsListId)
          this.$errorMessage = $('#error-message-no-products')
      }
      setUrl(url) {
          this.productListUrl = url
      }
      loadProducts() {
          if (this.productListUrl != null) {
              $.ajax(this.productListUrl)
                  .done((response) => {
                      this.$outputList.empty()
                      if(response.products.length > 0) {
                          this.$errorMessage.removeClass("show")
                          for (const product of response.products) {
                              this.$outputList.append(this.createProductItem(product))
                          }
                      } else {
                          this.$errorMessage.addClass("show")
                      }
                  })
          } else {
              console.log("Error: Invalid URL for Productslist: " + this.productListUrl)
          }
      }
      createProductItem(product) {
          const productView = new ProductView(product)
          return productView.createView()
```

```
    }
}
```

js/index.js

```
$(document).ready(function() {
    let listTypesUrl = "api/?action=listTypes"
    //let productListUrl  = "api/?action=listProductsByTypeId&typeId="
    let productListView = new ProductsListView('products-list')
    let productTypesView = new ProductTypesView(listTypesUrl, "types-list",
productListView)
    productTypesView.drawListGroup()
})
```