

Restaurant Ready!

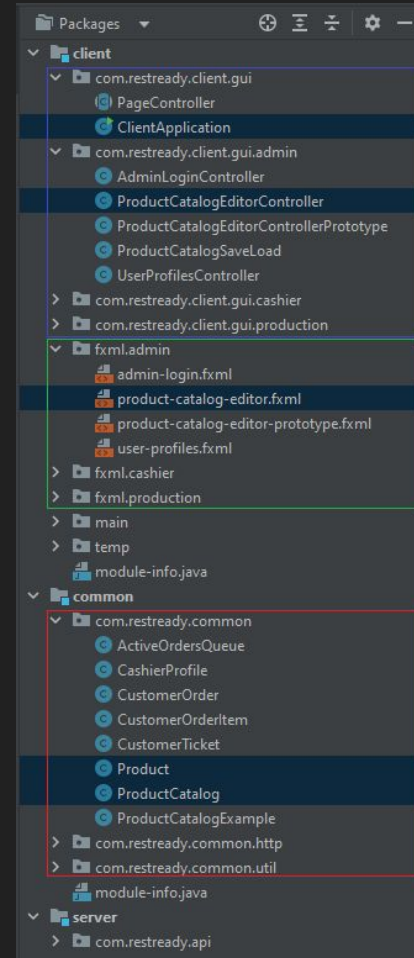
Code Review Demo

About Our Project

- It's a Point-of-Sale (POS) system
 - Product order entry + fulfillment tracking
 - Primarily for restaurants
- Client-server design with MVC architecture
 - Client = JavaFX... Standalone desktop GUI application (partially implemented)
 - Server = SpringBoot (not yet implemented)
- Three primary users interact via GUI
 - Admin = business owner
 - Defines product catalog + manages all other users
 - Cashier = waitstaff
 - Enters customer order into POS system + facilitates transactions
 - Order processing/fulfillment = kitchen
 - Receives incoming orders + updates production status

About the MVC

- Model = simple Java “bean” objects
 - Data model exists in “common” module
 - Accessible from “client” and “server”
 - Modules “client” and “server” do not access each other
 - Everything is java.io.Serializable = i/o directly to/from file
 - Later... SpringBoot provides shared Control + dictates Model
- View = FXML (file extension = .fxml)
 - Basically like HTML, but way cooler (if you like Java)
 - We use JavaFX’s SceneBuilder app to set 80% of FXML GUI
- Control = JavaFX
 - @FXML tags
 - Acquire references to GUI elements
 - “Hook-up” event handlers = triggered by user activity



Control
(JavaFX)

View
(FXML)

Model
(Java)

+SpringBoot

@FXML example: Product Data Entry + TableView

Cosmic Pizza

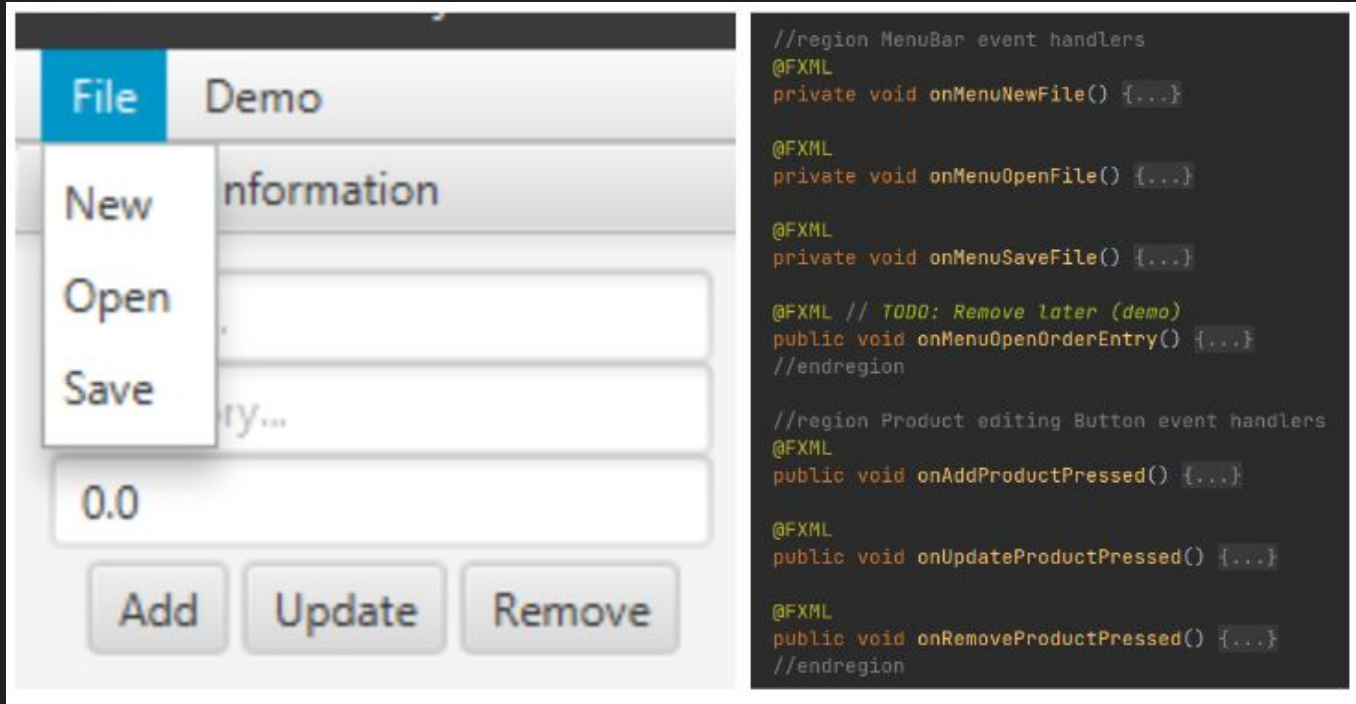
Bogos binted

1.0

```
//region FXML field references
@FXML
private TableView<Product> productCatalogTableView;
@FXML
private TextField productNameTextField;
@FXML
private TextField productCategoryTextField;
@FXML
private TextField productPriceTextField;
//endregion
```

UUID	Name	Category	Price
21c5ed89-06ee-460e-93...	Meteor Mac	Bogos binted	46.0
4696841f-d1bc-4c95-aee...	Astro Burger	Bogos binted	87.0
21f36174-32bf-44ab-98a...	Alien Toast	Bogos binted	96.0
d85e5388-0c56-40c9-b9...	Space Sliders	Vorp	54.0
9d2671c8-b05b-4e2a-b1...	Cosmic Pizza	Bogos binted	1.0

@FXML example: MenuBar (File I/O) + CRUD Buttons



Code Review: Helper Methods

- `resetProductTextFields()`
 - Called in `onMenuNewFile()`
 - Clears all of the text input fields + reset “focus”
- `updateProductFromTextFields(Product)`
 - Called in `onUpdateProductPressed()`
 - Gets data from text fields (input by user)
 - Sets data in product (table row selected)
- `createProductFromTextFields()`
 - Called in `onAddProductPressed()`
 - Gets data from text fields (input by user)
 - Returns new product with data

```
private void resetProductTextFields() {
    productNameTextField.clear();
    productCategoryTextField.clear();
    productPriceTextField.clear();
    getRoot().requestFocus(); // Clear any control focus
}

private void updateProductFromTextFields(Product product) {

    String name = productNameTextField.getText();
    String category = productCategoryTextField.getText();
    String price = productPriceTextField.getText();

    if (name.isBlank() || category.isBlank() || price.isBlank()) {
        return;
    }

    product.setName(name);
    product.setCategory(category);
    product.setPrice(Double.parseDouble(price));
}

private Product createProductFromTextFields() {

    String name = productNameTextField.getText();
    String category = productCategoryTextField.getText();
    String price = productPriceTextField.getText();

    if (name.isBlank() || category.isBlank() || price.isBlank()) {
        return null;
    }

    Product product = new Product();
    product.setName(name);
    product.setCategory(category);
    product.setPrice(Double.parseDouble(price));
    return product;
}
```

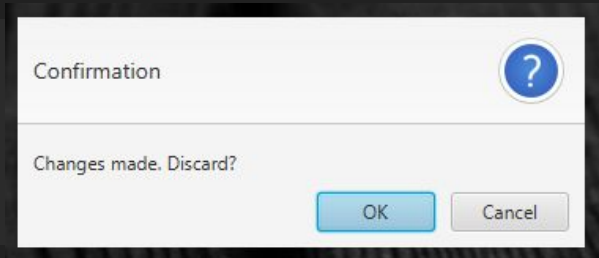
Programming Patterns 101: WTF is a “dirty flag”?

A change happened (add/update/remove)... set “dirty” flag to true:

```
productCatalog.addProduct(product);  
productCatalogDirty = true;
```

Elsewhere: confirm with user that next action overwrites changes
[like in onMenuFileOpen() / onMenuFileSave()]

```
// Open an "OK/Cancel" dialog  
if (productCatalogDirty && !confirmChangesMade()) {  
    return;  
}
```



All changes committed... set “dirty” flag as false (i.e., “clean”)

```
ProductCatalogSaveLoad.saveTo(file, productCatalog);  
productCatalogDirty = false;
```