

# Building a Canvas App with a Shopping Cart



# Agenda

- Caching State using Variables and Collections
- Using a Collection to Track Shopping Cart Data
- Using Patch Instead of an Edit Form
- Writing Shopping Cart Data to Back to SharePoint
- Designing Reusable Components



# Designing with Variables and Collections

- Context variables
  - Created as primitive, record or table at screen scope
  - Managed using `UpdateContext` and `Navigate`
- Global variables
  - Created as primitive, record or table at app scope
  - Created and managed using `Set` function
- Collections
  - Created as tables at app scope
  - Managed using `Collect`, `Clear` and `ClearCollect`
  - Can be stored to local device using `SaveData` & `LoadData`



# Code Naming Conventions

- Consistent naming conventions for variables & collections
  - Names should indicate type, purpose and scope
  - Collection names start with **col** such as **colProductCatalog**
  - Global variable names start with **gbl** such as **gblCustomApiUrl**
  - Context variables start with **loc** such as **locCustomerFilter**

## Collections

- colLastOrderDetails
- colNavigationTable
- colProductCatalog**
- colShoppingCart
- colLastOrder

## colProductCatalog

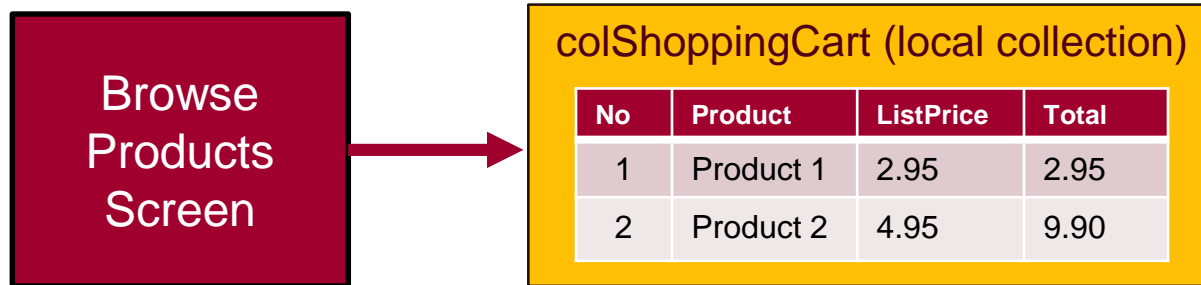
Here's a preview of the first 5 items in this collection  
[Learn about working with collections](#)

Category	Description	ListPr...	Product
Action Figures	A super hero who sometimes plays the role of a dark knight.	14.95	Batman Action Figure
Action Figures	A super action figure that protects freedom and the American way of life.	12.95	Captain America Action F

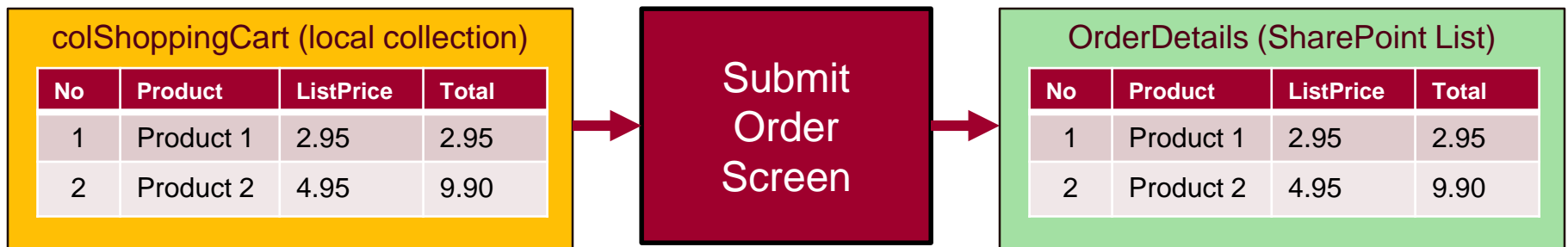


# Designing Canvas Apps using Collections

- Browse Products Screen allows user to build collection

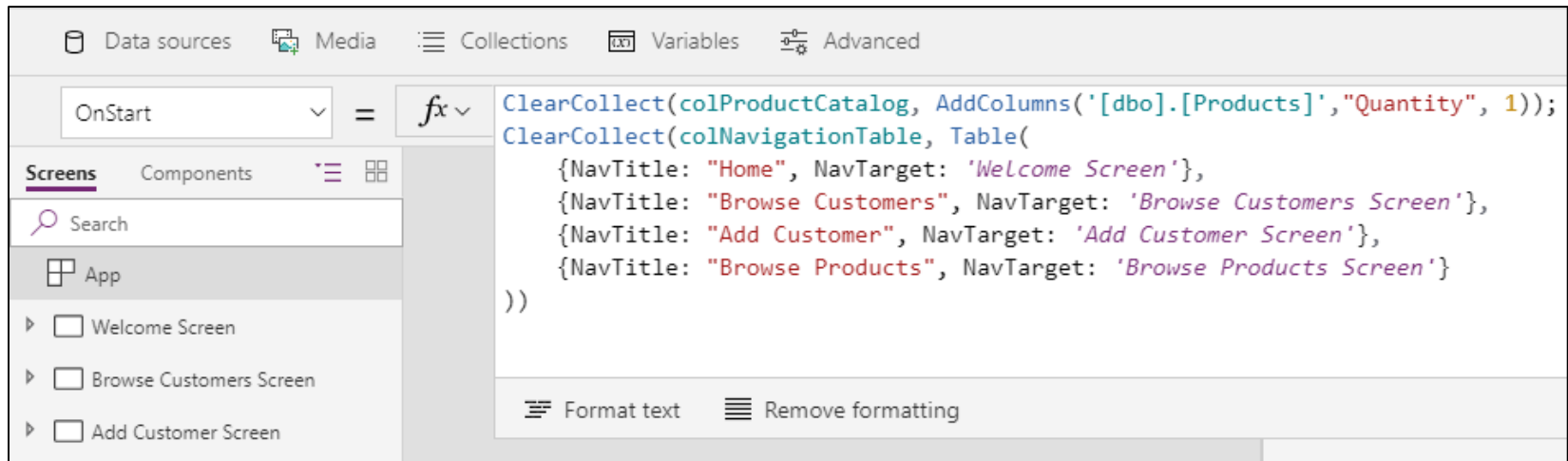


- Submit Order Screen allows user to save to SharePoint



# App OnStart

- App **OnStart** property used to initialize state in app
  - Event provides support for initializing state in app at startup
  - Commonly used to initialize global variables and collections
  - Right-click **App** in left navigation to run **OnStart** while in editor



# Agenda

- ✓ Caching State using Variables and Collections
- Using a Collection to Track Shopping Cart Data
  - Using Patch Instead of an Edit Form
  - Writing Shopping Cart Data to Back to SharePoint
  - Designing Reusable Components







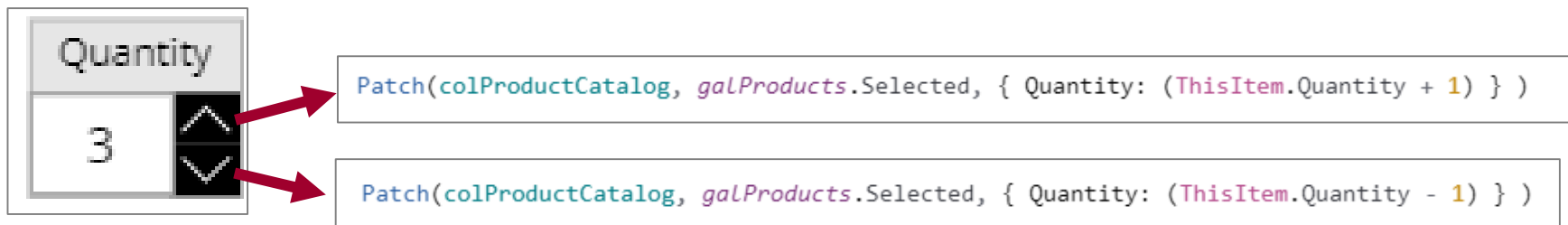
# Updatable Collection Columns

- Often helpful to add updatable column to collections

```
ClearCollect(colProductCatalog, AddColumns('[dbo].[Products]', "Quantity", 1));
```

	<b>Batman Action Figure</b> A super hero who sometimes plays the role of a dark knight.	Quantity 1	ADD TO CART
	<b>Captain America Action Figure</b> A super action figure that protects freedom and the American way of life.	Quantity 3	ADD TO CART

- Columns updated using calls to Patch







**DEMO**

# Designing A UI Experience with a Shopping Cart

# Agenda

- ✓ Caching State using Variables and Collections
- ✓ Using a Collection to Track Shopping Cart Data
- Using Patch Instead of an Edit Form
  - Writing Shopping Cart Data to Back to SharePoint
  - Designing Reusable Components



# Using Patch Instead of an Edit Form

- Sometimes edit forms are not the best option
  - **Patch** function used to create and update records in data source

```
Patch(  
    '[dbo].[Orders]',  
    Defaults('[dbo].[Orders]'),  
    {  
        CustomerId: galCustomers.Selected.CustomerId,  
        OrderAmount: Sum(  
            colShoppingCart,  
            Total  
        ),  
        OrderDate: Today()  
    }  
)
```

- **Patch** function can be used with **ForAll** to insert multiple records

```
ForAll(  
    colShoppingCart,  
    Patch(  
        '[dbo].[OrderDetails]',  
        Defaults('[dbo].[OrderDetails]'),  
        {  
            OrderId: First(colLastOrder).OrderId,  
            ProductId: ProductId,  
            Quantity: Quantity,  
            Total: Total  
        }  
    )  
)
```




# Agenda

- ✓ Caching State using Variables and Collections
- ✓ Using a Collection to Track Shopping Cart Data
- ✓ Using Patch Instead of an Edit Form
- Writing Shopping Cart Data to Back to SharePoint
- Designing Reusable Components




# Capturing Return Value from Patch

```
ClearCollect(
    collastOrder,
    Patch(
        '[dbo].[Orders]',
        Defaults('[dbo].[Orders]'),
        {
            CustomerId: galCustomers.Selected.CustomerId,
            OrderAmount: Sum(
                colShoppingCart,
                Total
            ),
            OrderDate: Today()
        }
    )
);
ClearCollect(
    collastOrderDetails,
    ForAll(
        colShoppingCart,
        Patch(
            '[dbo].[OrderDetails]',
            Defaults('[dbo].[OrderDetails]'),
            {
                OrderId: First(collastOrder).OrderId,
                ProductId: ProductId,
                Quantity: Quantity,
                Total: Total
            }
        )
    )
);
```



CustomerId	OrderAmount	OrderDate	OrderId
10	44.85	4/3/2019	14

**collastOrder**



Id	OrderId	ProductId	Quantity	Total
36	14	1	1	14.95
37	14	4	2	19.9
38	14	6	10	10

**collastOrderDetails**






# Populating UI from Cached State

CustomerId	OrderAmount	OrderDate	OrderId
10	44.85	4/3/2019	14

collastOrder

Id	OrderId	ProductId	Quantity	Total
36	14	1	1	14.95
37	14	4	2	19.9
38	14	6	10	10




collastOrderDetails

 **Order Confirmation** [Home](#) [Browse Customers](#) [Add Customer](#) [Browse Products](#)

The Order Has Been Submitted Successfully

**Order Information**  
Order ID:  
  
Order Date:  
  
Order Amount:  
  
Customer Shipping Address:

**Items purchased in Order #0014**

No	Product	List Price	Total	
1	Batman Action Figure	\$14.95	\$14.95	
2	Green Hulk Action Figure	\$9.95	\$19.90	
10	Twitter Follower Action Figure	\$1.00	\$10.00	

Shopping Cart Total: \$44.85





**DEMO**

**Writing Orders and Order Details to  
SharePoint using Patch**

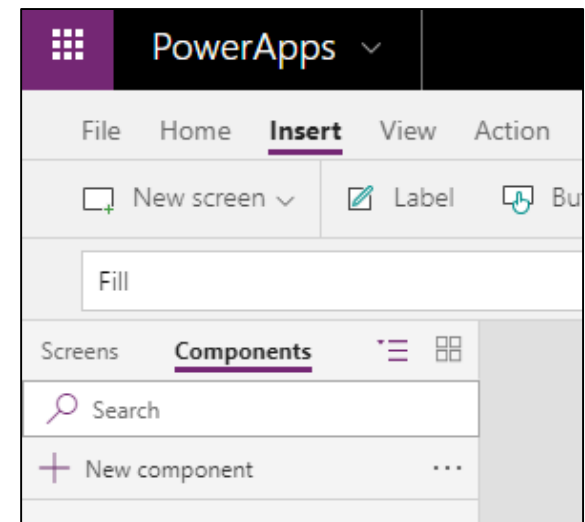
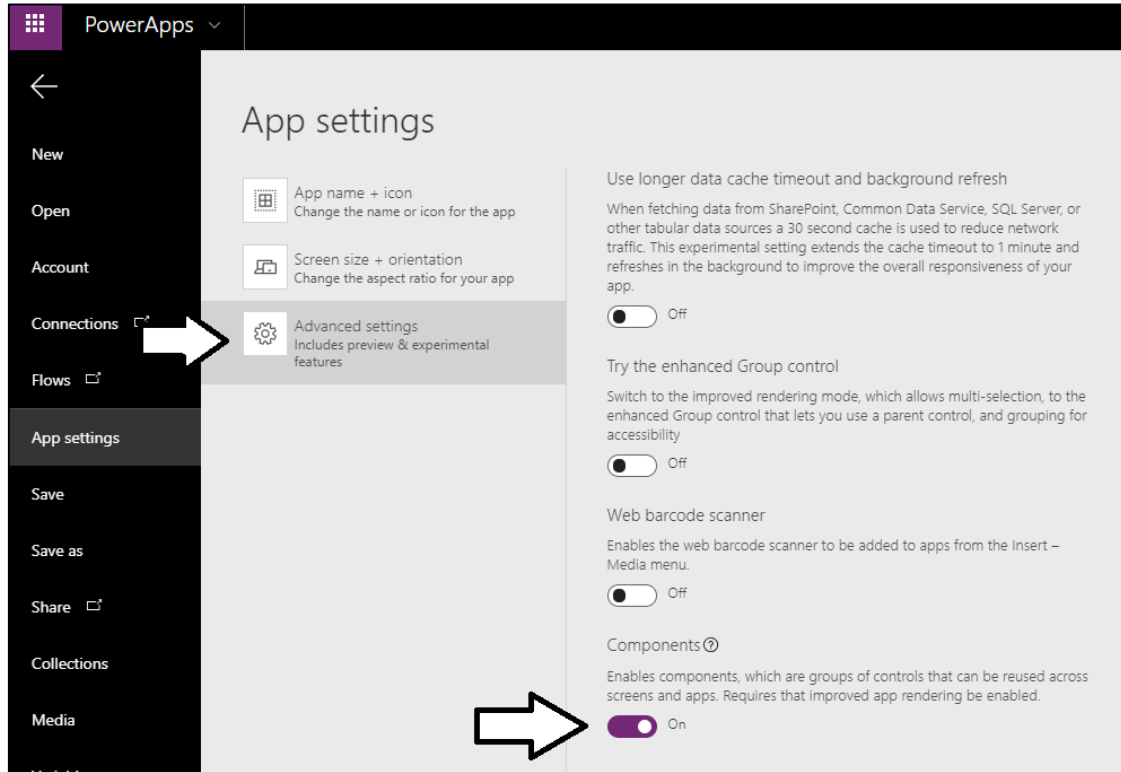


# Agenda

- ✓ Caching State using Variables and Collections
- ✓ Using a Collection to Track Shopping Cart Data
- ✓ Writing Shopping Cart Data to Back to SharePoint
- Designing Reusable Components

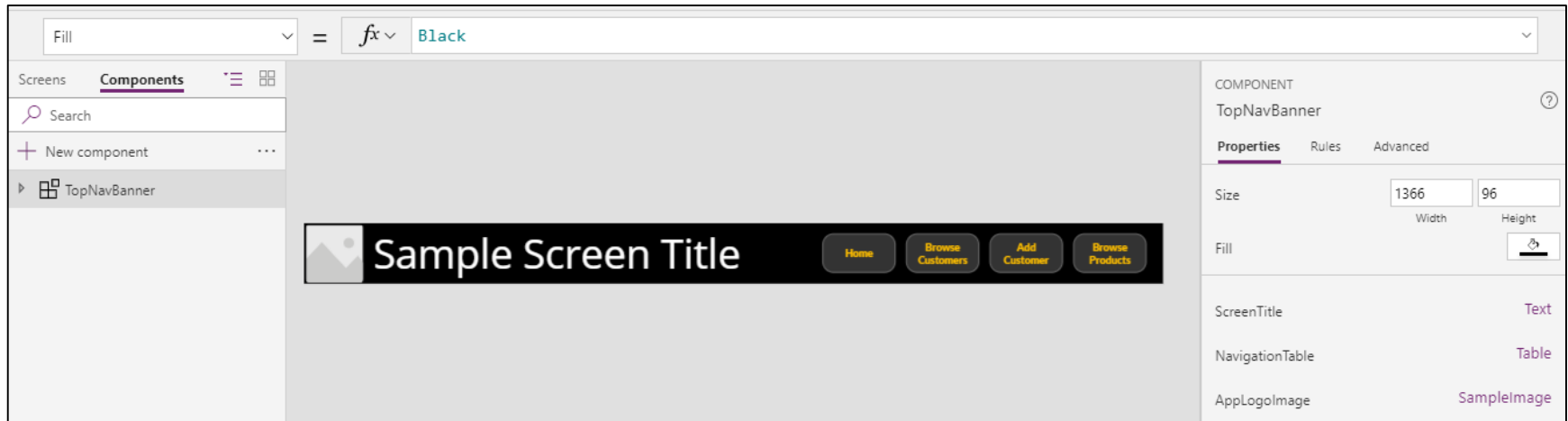
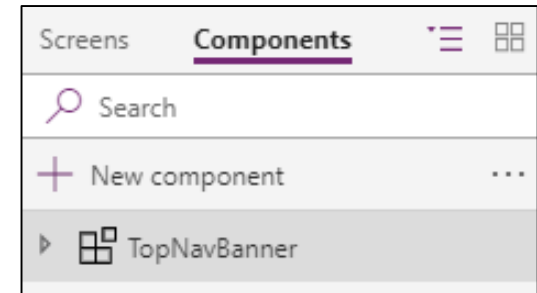


# Enabling Components in a Canvas App



# Designing Components

- Steps to using components
  - Create new component
  - Add component properties
  - Implement component UI and behavior
  - Add component to screens in your canvas apps



# Summary

- ✓ Caching State using Variables and Collections
- ✓ Using a Collection to Track Shopping Cart Data
- ✓ Writing Shopping Cart Data to Back to SharePoint
- ✓ Designing Reusable Components

