BRILLIANCE
BUSINESS SOLUTIONS

Episerver Digital Commerce TOOL BOX

www.BrillianceWeb.com

Downsides/Gotchas of using Projections in Episerver Find

By Mike Clausing

Episerver Find is my preferred way to write searches for Episerver. It allows me to write search queries quickly using a LINQ like syntax I am familiar with, while leveraging Episerver's cloud servers to do the heavy lifting.

While this does shift some of our server load to Episerver's servers, the standard implementation still relies on my servers to load model data, and build view models. Generally this isn't slow, but processing complex view models can hurt performance, and impact server load, especially as result sets increase in size, and complexity.

Projections (https://world.episerver.com/documentation/developer-guides/find/NET-Client-API/searching/Projections/) allow me to specify which fields Episerver Find should retrieve data for, for a given query. I try to use them wherever it makes sense, since it allows us to display results with very little server load. There are a few downsides/gotchas that I've had to work through though.

Extra Extension Methods

The LINQ-like methods that can be used in conjunction with projections are much more limited than a normal Find search allows. Even simple string manipulation extension methods are not allowed, and will result in the projection not working properly.

This is easy to test with Fiddler. If the field being requested in the projection is not included in the response, the extension method is either not getting indexed properly, or is using an incompatible extension method in the projection.

There should be an extension method for every field to be used in the results, and it should be indexed during initialization.

```
public static string MyExtensionMethod(this BlogPostPage blogPostPage)
        {
            // Implement your logic here
        }
```

```
// Add to your initialization so it is indexed properly
var searchClient = ServiceLocator.Current.GetInstance<IClient>();
            searchClient.Conventions.ForInstancesOf<BlogPostPage>()
                .IncludeField(x => x.MyExtensionMethod());
```

Index must always be up to date

Since Find searches with projections rely solely on the data in your Find index, and not the Episerver database to create your view models, your Find index always up to date.

In most use cases, this is not an issue as Find re-indexes data for you when changes are made. This isn't always the case when your extension methods require information from other models however.

IE an extension method for a ProductContent that requires data from a VariationContent.

This would require VariationContent updates to trigger reindexing for their associated ProductContents in order to keep the Find index correct.

Difficult, but not impossible to search in specific fields for multiple types

Using projections across multiple types, and searching in specific fields using inField do not currently work well together. This is due to inField being only available on ITypeSearch.

The easiest way I have found to implement it this is to is somewhat messy, but effective:

- Search for a base type which all of your models inherit from,
- Cast as needed to search in specific fields,
- Branch/Cast as needed in projections.

```
        var search = _searchClient.Search<PageData>(language)
                        .For(searchTerm.Quote())
                        .InField(x => x.Name)
                        .InField(x => x.MetaDescription)
                        .InField(x => ((AboutPage)x).SubTitleText)
                        .InField(x => ((BlogPostPage)x).FindMainBodyText())
                        .Select(x => new Article
                        {
                            Title = x.Name,
                            Description = x.PageTypeName == nameof(AboutPage)
                                ? ((AboutPage)x).SubTitleText
                                : x.PageTypeName == nameof(BlogPostPage)
                                    ? ((BlogPostPage)x).FindMainBodyText()
                                    : x.MetaDescription,
                            Link = x.FindUrl()
                        });
```

Custom tracking must be implemented

Since projections rely entirely on the data in your Find index, we need to create extension methods for hitId, hitType for Find statistics, and implement custom tracking as explained by Sebastian Sebusæter Enberget (https://blog.enberget.com/episerver-find-custom-tracking/).

```
        public static string HitId(this IContent content)
        {
            return SearchClient.Instance.Conventions.IdConvention.GetId(content);
        }

        public static string HitType(this IContent content)
        {
            return SearchClient.Instance.Conventions.TypeNameConvention.GetTypeName(content.GetType());
        }
```

Overall, these changes result in more work to implement, but I have seen really good performance when projections are implemented correctly.

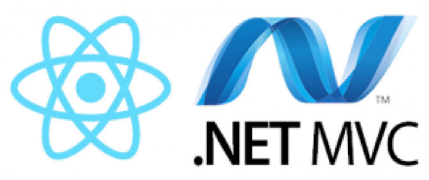
About Mike Clausing

Mike is an Episerver Certified Developer with experience in CMS and FIND, as well as a Microsoft Certified Professional (Microsoft HTML5 with JavaScript & CSS3). He specializes in B2B eCommerce development at Brilliance.

Follow Mike Clausing on in Linkdin (https://www.linkedin.com/in/michael-clausing-053ab84b/)

Part III: Making a create-react-app project in .NET in Episerver

By Drew Douglas



This post is a part of a series on integrating react and ASP.NET.

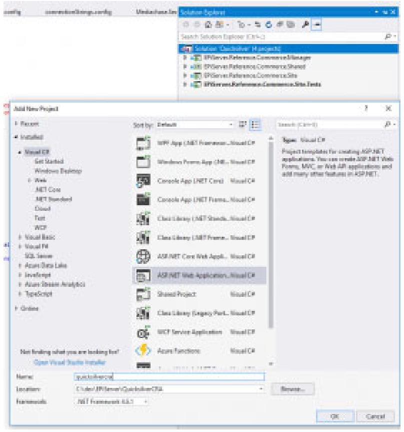
- Part I – Introduction (/resources/create-react-app-and-asp.net-mvc-5-into-episerver-introduction/)
- Part II – Install Quicksilver (/resources/install-episerver-quicksilver/)
- **Part III – Create create-react-app project**

What: create-react-app on Episerver
Why: create-react-app is easy. react and Episerver are powerful.

In my last post (/resources/install-episerver-quicksilver/) I discussed how to install Episerver Quicksilver (https://github.com/episerver/Quicksilver) site. As I note there, there's a license cost to using Episerver, but the techniques I discuss below can be applied to any ASP.NET MVC 5 site.

A create-react-app .NET web application project

We'll create a new .NET web application project to place the create-react-app project's files in. We won't compile a website there, but simply hold our create-react-app project (sort of like the node.js project type that Visual Studio offers you as a project template (https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-nodejs)). [1] [2]


Solution Explorer
Solution 'Quicksilver' (4 projects)
EPiServer.Reference.Commerce.Manager
EPiServer.Reference.Commerce.Shared
EPiServer.Reference.Commerce.Site
EPiServer.Reference.Commerce.Site.Tests
Add New Project
Recent
Installed
Visual C#
Get Started
Windows Desktop
Web
.NET Core
.NET Standard
Cloud
Test
WCF
Visual Basic
Visual F#
SQL Server
Azure Data Lake
JavaScript
Azure Stream Analytics
TypeScript
Online
Sort by: Default
WPF App (.NET Framework) Visual C#
Windows Forms App (.NET...) Visual C#
Console App (.NET Core) Visual C#
Console App (.NET Frame...) Visual C#
Class Library (.NET Standa...) Visual C#
Class Library (.NET Frame...) Visual C#
ASP.NET Core Web Appli... Visual C#
ASP.NET Web Application... Visual C#
Shared Project Visual C#
Class Library (Legacy Port...) Visual C#
WCF Service Application Visual C#
Azure Functions Visual C#
Not finding what you are looking for?
Open Visual Studio Installer
Browse...
OK
Cancel

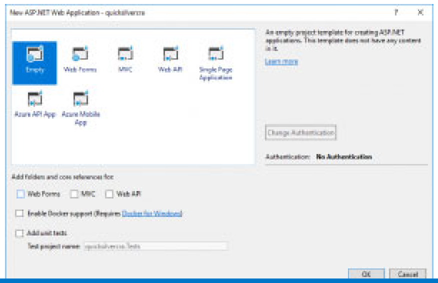
Type: Visual C#

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Name: quicksilvercra
Location: C:\dev\EPiServer\QuicksilverCRA
Framework: .NET Framework 4.6.1

(https://drew.from-wi.com/wp-content/uploads/2018/08/createProject1.png)

The project should be empty with no authentication, folders, references, or tests.



An empty project template for creating ASP.NET applications. This template does not have any content in it.

Authentication: No Authentication

Add folders and core references for:

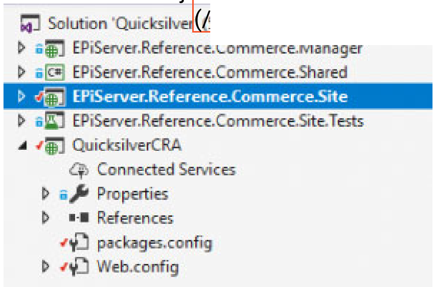
□ Web Forms □ MVC □ Web API

□ Enable Docker support (Requires Docker for Windows)

□ Add unit tests

Test project name: quicksilvercra.Tests

(https://drew.from-wi.com/wp-content/uploads/2018/08/createProjec


Solution 'Quicksilver
EPiServer.Reference.Commerce.Manager
EPiServer.Reference.Commerce.Shared
EPiServer.Reference.Commerce.Site
EPiServer.Reference.Commerce.Site.Tests
QuicksilverCRA
Connected Services
Properties
References
packages.config
Web.config

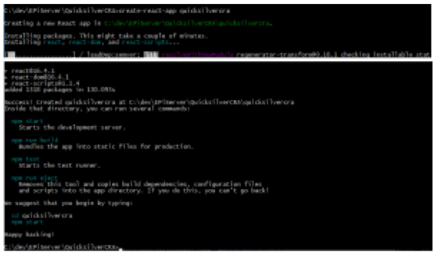
(https://drew.from-wi.com/wp-content/uploads/2018/08/empy_QuicksilverCRA_project.png)

Installing create-react-app

Init new project

We'll install create-react-app globally so we can use it to create new empty react projects. This is analogous to using Visual Studio to create new .NET projects. We'll place the react project in the solution folder (next to the quicksilver Sources folder) for now. 3 4

```
Sources\                                                                                    MS DOS
npm install -g create-react-app
create-react-app quicksilvercra
```

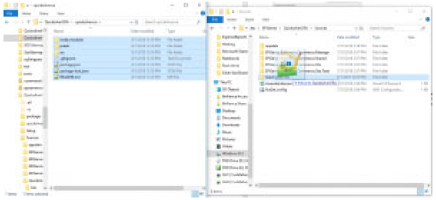


```
C:\dev\EPiServer\QuicksilverCRA>create-react-app quicksilvercra

Creating a new React app in C:\dev\EPiServer\QuicksilverCRA\quicksilvercra.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...

[..................] / loadDep:regenerator-transform: sill resolveWithNewModule regenerator-transform@0.10.1 checking installable stat

+ react@16.4.1
+ react-dom@16.4.1
+ react-scripts@1.1.4
added 1158 packages in 130.095s

Success! Created quicksilvercra at C:\dev\EPiServer\QuicksilverCRA\quicksilvercra
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd quicksilvercra
  npm start

Happy hacking!

C:\dev\EPiServer\QuicksilverCRA>
```

(https://drew.from-wi.com/wp-content/uploads/2018/08/create-react-app_install.png)

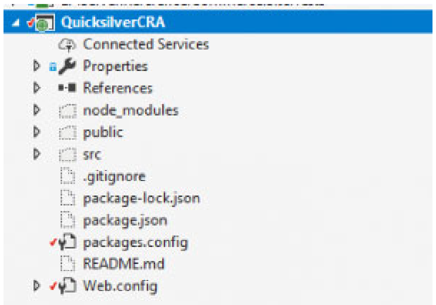
Move files to .Net project

Move all the files from the create-react-app directory into the new project and remove the empty directory create-react-app introduced.



(https://drew.from-wi.com/wp-content/uploads/2018/08/copy-create-react-app-into-project.png)

Go back to VS and show all files for the VS project we created above. Include the src and package.json files.

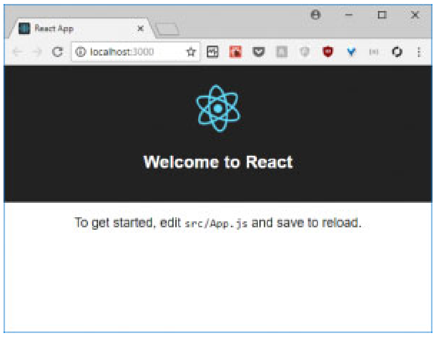


(https://drew.from-wi.com/wp-content/uploads/2018/08/quicksilver_cra_in_project.png)

Start create-react-app

At this point, you should be able to cd to the new project and start the create-react-app application.

```
Sources\QuicksilverCRA\                                                                                  MS DOS
npm run start
```



Welcome to React

To get started, edit src/App.js and save to reload.

Building a create-react-app project

The create-react-app contents are there to serve the complete react SPA. However, the only parts we'll use in the Epi site are the js and css created for our components. The dev server that create-react-app launches serves the js by running webpack to generate a transpiled file of all your js. It does this in memory and serves it from the virtual path

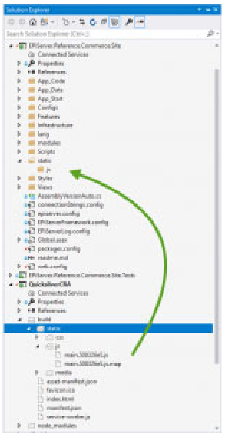
`~/static/js/bundle.js` . Then, when you're ready to deploy running

```
Sources\QuicksilverCRA\                                                                                  MS DOS
npm run build
```

creates a webpack build optimized for production and sticks it in the build/static/js folder of the create-react-app project with a the name `main.[hash].js` (where `[hash]` is a hash computed from the file's contents).

Our plan for production

The trick we want to pull off is to always use the path `static/js/bundle.js` . That way, we can reference that virtual path in our MVC project in both development and production scenarios. Using the proxy in development, the express server that serves `~/static/js/bundle.js` will serve the in-memory bundled code. When we go to production, we need to rename the `main.[hash].js` file to `bundle.js` and stick it in the website's `/static/js/` folder. [5]



(https://drew.from-wi.com/wp-content/uploads/2018/08/copy_js_to_web.png)

A proxy for Quicksilver with SSL

The point of a proxy is that we want webpack-dev-server to reload our bundle.js whenever we save our js code, but we want all other requests to be handled by our ASP.NET site.

Configure the proxy

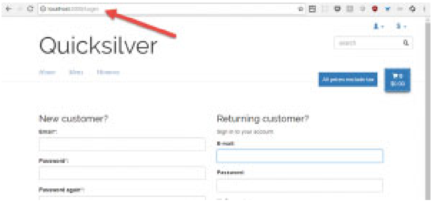
Add the new member "proxy" to the package.json file created by create-react-app. The proxy will serve the bundle from the path /static/js/bundle.js, but will proxy requests to the quicksilver site for everything else:

```
Sources\QuicksilverCRA\package.json
{
  "name": "quicksilvercra",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "react": "^16.4.1",
    "react-dom": "^16.4.1",
    "react-scripts": "1.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test --env=jsdom",
    "eject": "react-scripts eject"
  },
  "proxy": {
    "/static/js/bundle.js": {
      "target": "http://localhost:3000"
    },
    "/.*": {
      "target": "http://localhost:50244"
    }
  }
}

```

Run npm start again, start the quicksilver site in VS, and manually enter the url

```
localhost:3000/Login
```

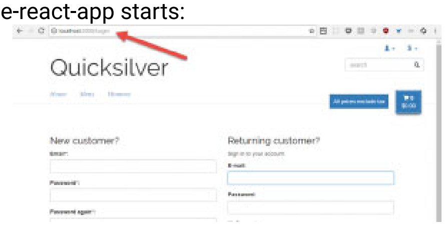
You should see the login page for the quicksilver site being proxied through the express server create-react-app has started.



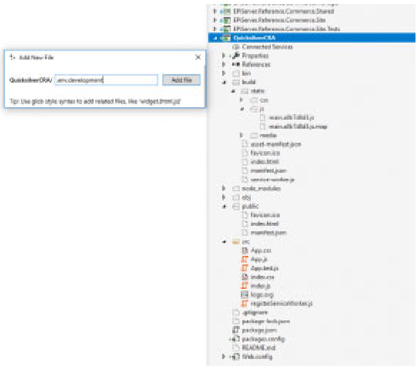
(https://drew.from-wi.com/wp-content/uploads/2018/08/quicksilver_proxied.png)

Using SSL

Note, however, that we’re not using ssl. To allow ssl, we need to tell create-react-app to use HTTPS. Add a file .env.development that holds environment variables to pass to the server create-react-app starts:



(https://drew.from-wi.com/wp-content/uploads/2018/08/quicksilver_proxied-1.png)



Add New File

QuicksilverCRA\ .env.development

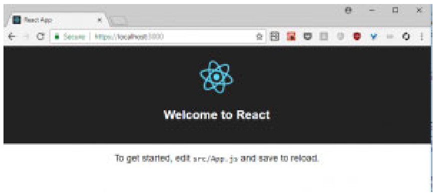
Tip: Use glob style syntax to add related files, like 'widget.(html,js)'

(https://drew.from-wi.com/wp-content/uploads/2018/08/create-development-env-file.png)

In the file, place the single environment variable assignment:

```
Sources\QuicksilverCRA\.env.development
HTTPS=true
```

Start the react app again and observe that your computer doesn't trust the certificate. Copy it to a file and import it into Trusted Root Certificates. (Note that this is a somewhat dangerous for your development environment. An attacker with this certificate can use it to (/EPiServer/CMS/?language=en#context=epi.cms.contentdata:///2540) certificates that your computer will trust. For this reason, the certificates only have a 1-month lifetime before they expire. You'll need to re-import the certificate when it gets recreated by webpack-dev-server after that. Make sure to delete the exported certificate when you are done. Restart your browser to see that the new certificate is trusted

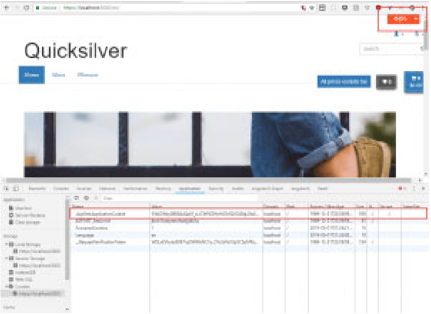

React App
Secure | https://localhost:3000
Welcome to React
To get started, edit src/App.js and save to reload.

(https://drew.from-wi.com/wp-content/uploads/2018/08/trusted-dev-server-certificate.png)

Now, we can update our proxy config to use SSL. If we change the express proxy to use https, and the Quicksilver proxy to use https, however, it still fails, because express doesn't trust the ssl setup by IIS Express. Add the option "secure": false to disable leaf validation (https://github.com/nodejitsu/node-http-proxy#options) because we the express server doesn't trust the iis express ssl.

```
Sources\QuicksilverCRA\package.json                                    JavaScript
  "proxy": {
    "/static/js/bundle.js": {
      "target": "https://localhost:3000"
    },
    "/.*": {
      "target": "https://localhost:44301",
      "secure": false
    }
  },
```

Now, we can log into the Quicksilver site using the proxy and we should see the .AspNet.ApplicationCookie and, if we're an admin, the epi quickNavigator:



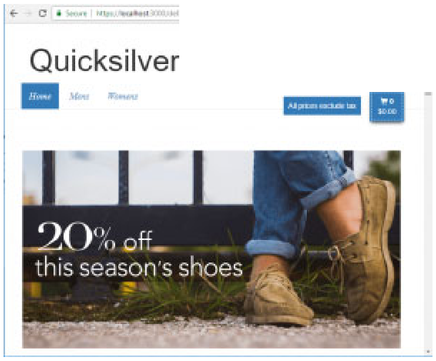
(https://drew.from-wi.com/wp-content/uploads/2018/08/login-succesful.png)

Redirecting Default

There's one more problem: we can't get to the Quicksilver root homepage through the proxy, because react is serving the app from there (and it ignores attempts proxy the root). Add a path rewrite object with the path `^/default` redirecting to `/` :

```
Sources\QuicksilverCRA\package.json
  "proxy": {
    "/static/js/bundle.js": {
      "target": "https://localhost:3000"
    },
    "/.*": {
      "target": "https://localhost:44379",
      "secure": false,
      "pathRewrite": {
        "^/default": ""
      }
    }
  },
```

Now you can browse to `/default` to get the Quicksilver homepage.

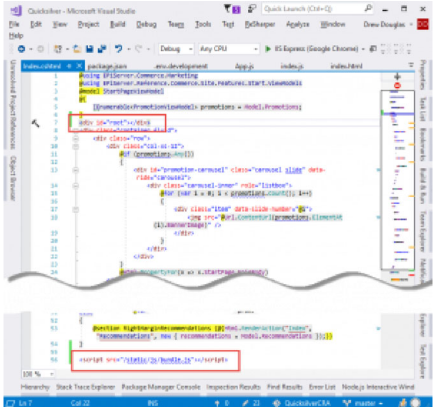


(https://drew.from-wi.com/wp-content/uploads/2018/08/quicksilver-homepage-proxied.png)

Adding our app to MVC

From here we could reference the bundle from quicksilver and display some react components by adding a `div` with `id="root"` and the script reference to `/static/js/bundle.js` to our homepage view:

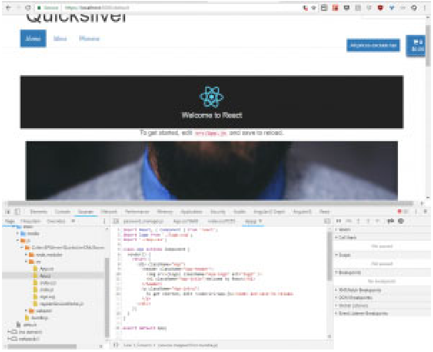
```
Sources\EPiServer.Reference.Commerce.Site\Views\Start\Index.cshtml                                   XHTML
...

<div id="root"></div>

...

<script src="/static/js/bundle.js"></script>
```



```
@using EPiServer.Commerce.Marketing
@using EPiServer.Reference.Commerce.Site.Features.Start.ViewModels
@model StartPageViewModel
@{
    IEnumerable<PromotionViewModel> promotions = Model.Promotions;
}
<div id="root"></div>
<div class="container-fluid">
    <div class="row">
        <div class="col-xs-12">
            @if (promotions.Any())
            {
                <div id="promotion-carousel" class="carousel slide" data-ride="carousel">
                    <div class="carousel-inner" role="listbox">
                        @for (var i = 0; i < promotions.Count(); i++)
                        {
                            <div class="item" data-slide-number="@i">
                                <img src="@Url.ContentUrl(promotions.ElementAt(i).BannerImage)" />
                            </div>
                        }
                    </div>
                </div>
            }
            @Html.PropertyFor(x => x.StartPage.MainBody)

{
    @section RightMarginRecommendations {@{Html.RenderAction("Index", "Recommendations", new { recommendations = Model.Recommendations });}}
}

<script src="/static/js/bundle.js"></script>
```

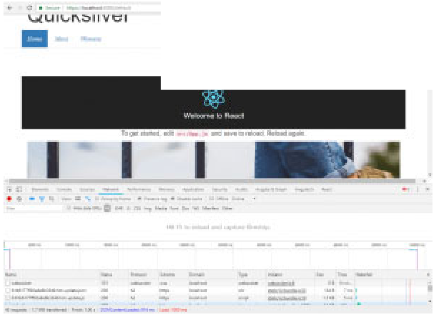
(https://drew.from-wi.com/wp-content/uploads/2018/08/react-in-mvc-view.png)

We reference the bundle in the location that the proxy server serves it from, and the `div` gives react a place to put the app on the page. [6]



(https://drew.from-wi.com/wp-content/uploads/2018/08/create-react-app-as-component-in-homepage.png)

To demonstrate that auto-reloading (https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md#available-scripts) is working, modify the component and save the file. Note that your changes are displayed without manually refreshing the page.


Quicksilver
Welcome to React
To get started, edit src/App.js and save to reload. Reload again.

(https://drew.from-wi.com/wp-content/uploads/2018/08/auto-reload-on-save-homepage.png)

About Drew Douglas

Drew is an Episerver Certified, as well as a Microsoft Certified Professional (Microsoft HTML5 with JavaScript & CSS3 Certification). As a Lead Developer at Brilliance, Drew serves as a technical lead and architect on Episerver projects. During his career, his development work has utilized many technologies, including ASP.NET Forms and MVC, NServiceBus and React. His current focus is B2B eCommerce development with ERP integrations.

Follow Drew Douglas on in Linkdin (https://www.linkedin.com/in/ddrewdouglas/)

Comet: Create-React-App for Episerver

By Jared Hackbart

React is a popular JavaScript library used to build user interfaces (UI) on the front end. You can design simple views for each state in your application, and React efficiently updates and renders just the right components when your data changes. React simplifies the development of modern user interfaces - simplifies doesn't equate to simple, however.

Facebook created an environment that comes pre-configured to facilitate building a React app called the Create-React-App. Create-React-App is a comfortable environment for learning React and is a great way to start building a new single-page application in React. Although React is simpler than alternative frameworks, diving into React can still be complicated.

Developers new to leveraging React in their front-end development for Episerver will likely spend days setting up their environment, working through various libraries, and troubleshooting techniques. This can result in lost productivity and frustration. Many React implementations fail to fully support search engine indexing by non-Google search engines, which can result in lost rankings, traffic, and revenue.

Server-side rendering is a crucial feature supported through our implementation, enabling your pages to be crawled by all search engines.

Brilliance's Comet App saves developers time in using React for front end development with Episerver (https://marketplace.episerver.com/apps/brilliance/comet-create-react-app-for-episerver/) by working through the installation nuances of:

• ReactJS.Net, React, Create-React-App
• Cache Busting
• Express Development Server

It can be a challenge to know how to get started for developers who are new to React or new to creating React components for Episerver. Brilliance's Comet app further simplifies the process and helps you get situated for success.

Click here to view the Comet app in Episerver's Marketplace. (https://marketplace.episerver.com/apps/brilliance/comet-create-react-app-for-episerver/)

For more information on the Comet app, check out our series on integrating react and ASP.NET:

- Part I – Introduction (/resources/create-react-app-and-asp.net-mvc-5-into-episerver-introduction/)
- Part II – Install Quicksilver (/resources/install-episerver-quicksilver/)
- Part III – Create create-react-app project (/resources/making-a-create-react-app-project-in-.net-in-episerver/)

Do you have questions or would like to discuss our Comet app further? Feel free to set up a time to talk! (https://go.oncehub.com/JaredHackbart)

Save developers time and headaches - focus your efforts on value-added activities for your business through Brilliance's Comet app!

- Save time getting started with React
- Search Engine indexing by non-Google search engines supported through server-side rendering
- Not worrying about cache busting, troubleshooting techniques, and other installation nuances
- Includes phone and email support with our experienced Episerver development team

About Jared Hackbart

Jared is passionate about providing clients with the right tools and applications to help their business grow. Drawing from his experiences at Canon Solutions of America & Ken Cook Co. with manufacturing companies, Jared provides solid solutions and guidance to Brilliance's clients on a daily basis.

Top 3 Questions Executives Need to Answer When Planning for Episerver Commerce

By Lori McDonald

(/Portals/3/Blog/Files/2/31/Windows-Live-Writer-Top-3-Things-to-Prepare-for-EPiServer-Co_13354-Questions_EPiServer_Commerce_2.jpg) Episerver Commerce is a powerful framework for implementing enterprise eCommerce. Just like any powerful tool, it is only as good as effective as the plan and implementation. If you are preparing for an Episerver Commerce project, here are three questions to start your planning process.

1. **What will be the source of truth for our data?**
   This is a question that you will need to answer for every data type that will be used in your Episerver Commerce site. Whether it's customer, product, order or other entity data, you will need to clearly define where data is being read from and where data is being written to. We are making the assumption here that you will be integrating the data on your site to another business system, as this is the case in the majority of instances for this type of project. The beauty of the Episerver Commerce framework is that it can be built to match your business processes, but for your Episerver Commerce project to be successful, this means your business processes need to be clearly defined up-front, including where data will be sourced from, written to and what workflow will exist in creating, updating and deleting different data types.
2. **What will be our customer model?**
   Will customer data be account-centric or user-centric? Can there be multiple users per account? If yes, what account data is shared among the users? What is the process for creating a user? Are anonymous users allowed to check out? The answers to these questions will likely be dependent on your overall goals for the project, but they are important to determine up-front. If you want multiple users per account, that can be designed into your system in the beginning, but adding it to an existing Episerver Commerce site would be much more complex.
3. **How will we structure pricing on our site?**
   Episerver Commerce's tiered pricing model can allow complex pricing structures to be implemented. To properly implement pricing, you will need to define how the pricing tiers will be structured on your site. Episerver Commerce allows you to define different customer groups that will receive different pricing based on specific characteristics. Each customer can be in multiple groups that each have their own pricing defined, and Episerver Commerce will show the customer the lowest price from any qualifying group they are in. For example, customers in the USA would be set up with the price of $150.00 for SKU 1234, customers in the state of Illinois would be set up with a price of $145.00 for SKU 1234, and account XYZ (that is in Illinois) would be set up with a price of $140.00 for SKU 1234. When a customer in account XYZ logs in, they would see a price of $140.00, but when another customer in Illinois logs in they would see a price of $145.00. The specifics of how your pricing will work will be dependent on your business model and your current pricing structure. Along with defining how pricing will be structured, there needs to be a consideration for how the pricing information will be maintained (which often leads us back to our first question.)

Brilliance Is Here to Help
The great news is that Episerver Commerce has a well-designed architecture for meeting a variety of business needs. To truly harness this for your organization, it is important to plan your project right from the start. If you are planning for your Episerver Commerce project and are looking for a company to guide you through the planning process, feel free to give us a call at 414-425-4069 x150.

About Lori McDonald

Lori graduated from Purdue University with a Bachelor’s degree in Computer-Electrical Engineering and leads Brilliance Business Solutions with over 20 years of computer engineering and software development experience. She is an Episerver EMVP, (https://world.episerver.com/blogs/Allan-Thran/Dates/2018/1/7-new-emvps/) a Microsoft Certified Professional and a regular contributor on Practical eCommerce (https://www.practicalecommerce.com/author/lori-mcdonald). Her status as a recognized industry expert has resulted in regular speaking engagements at business conferences.

Follow Lori McDonald on Linkdin (https://www.linkedin.com/in/brillweb/)

Drive Sales Through Personalized Content with Episerver Commerce

By Lori McDonald

According to Forrester Research, 70% of marketing executives believe personalization to be of strategic importance (http://www.forbes.com/sites/sap/2014/03/12/personalization-the-secret-to-better-customer-experience/) to their business. Why? Because they believed that personalization would lead to increased website traffic, more conversions, and additional revenue. In the same study, many respondents shared challenges to deliver relevant content based on real-time customer interaction.

This is one of the reasons that we love Episerver Commerce. Episerver pairs a powerful commerce architecture with rich content management features including a robust personalization engine. Episerver Commerce enables marketers to quickly create rules that dynamically personalize content in real-time through visitor groups.

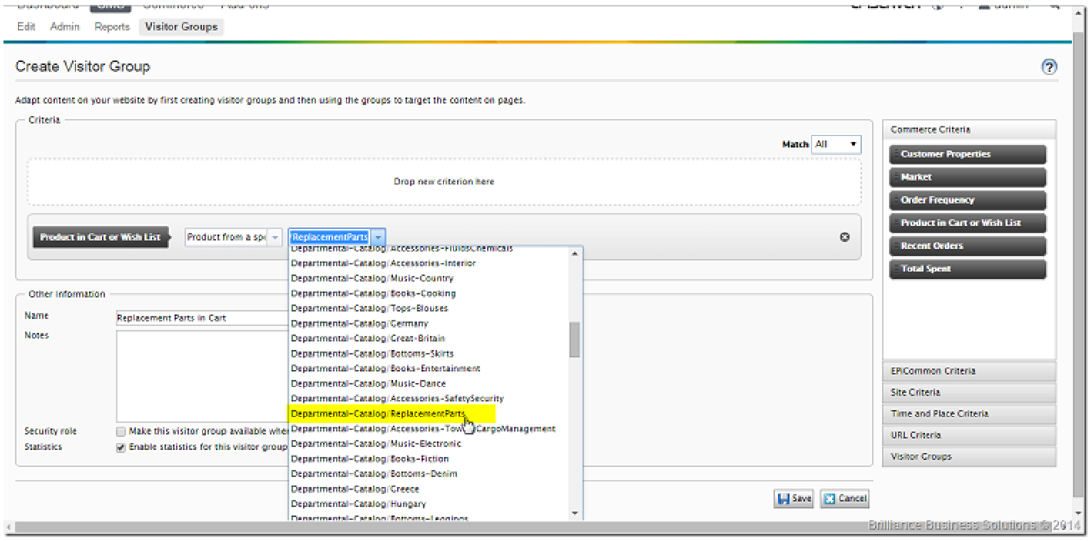
What is a Visitor Group?

A visitor group is a set of site users that are selected based on rules that site administrators set up within Episerver's web-based admin controls. You can base Visitor Groups on one or more properties.

You can define your own properties to use to select a user, or you can select from defined properties such as:

- If a specific product is in their cart or wish list
- If a product from a specific category is in their cart or wish list
- The geographic region of the customer (ZIP code, state, or country)
- How many times the customer has ordered in the last X days
- If they have spent more than X in the last Y days
- How many times the user has visited your site
- If they have visited a specific page
- If they have visited a specific category
- The Company or Job Title from the users' profile
- What URL they came from
- What landing page they started on

Here is an example visitor group, we defined to target visitors who have a product from the Replacement Parts category in their shopping cart or wish list.



Create Visitor Group

Adapt content on your website by first creating visitor groups and then using the groups to target the content on pages.

Criteria

Match: All

Product in Cart or Wish List | Product from a spe | ReplacementParts

Other Information

Name: Replacement Parts in Cart

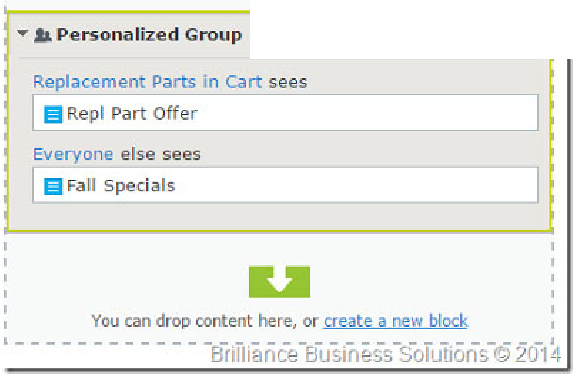
Notes:

Security role: ☐ Make this visitor group available whe

Statistics: ☑ Enable statistics for this visitor group

Edit
Admin
Reports
Visitor Groups
Drop new criterion here
Departmental-Catalog/Accessories-FluidsChemicals
Departmental-Catalog/Accessories-Interior
Departmental-Catalog/Music-Country
Departmental-Catalog/Books-Cooking
Departmental-Catalog/Tops-Blouses
Departmental-Catalog/Germany
Departmental-Catalog/Great-Britain
Departmental-Catalog/Bottoms-Skirts
Departmental-Catalog/Books-Entertainment
Departmental-Catalog/Music-Dance
Departmental-Catalog/Accessories-SafetySecurity
Departmental-Catalog/ReplacementParts
Departmental-Catalog/Accessories-TowCargoManagement
Departmental-Catalog/Music-Electronic
Departmental-Catalog/Books-Fiction
Departmental-Catalog/Bottoms-Denim
Departmental-Catalog/Greece
Departmental-Catalog/Hungary
Departmental-Catalog/Bottoms-Leggings
Commerce Criteria
Customer Properties
Market
Order Frequency
Product in Cart or Wish List
Recent Orders
Total Spent
EPiCommon Criteria
Site Criteria
Time and Place Criteria
URL Criteria
Visitor Groups
Save
Cancel
Brilliance Business Solutions © 2014

Personalizing Content



Personalized Group

Replacement Parts in Cart sees

Repl Part Offer

Everyone else sees

Fall Specials

You can drop content here, or create a new block

Marketers can quickly setup unique content to display to users in real-time, using content blocks and personalization by visitor groups.

Marketers can drag-and-drop content blocks or images and define which visitor groups will see the selected content.

Powerful Marketing Tools, Making Your Job Easier

Visitor groups and personalization controls are just one of the ways Episerver makes your site convert better, and the marketer's job easier. If you would like to learn more, or a see a demonstration of Episerver, feel free to give us a call at 414-425-4069 x150 or contact us online. (/free_consultation/)

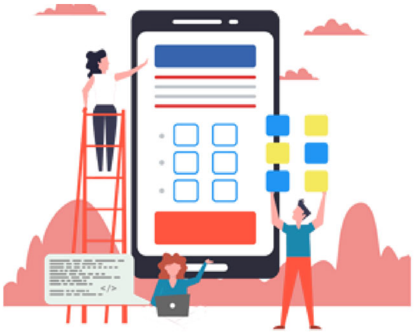
About Lori McDonald

Lori graduated from Purdue University with a Bachelor's degree in Computer-Electrical Engineering and leads Brilliance Business Solutions with over 20 years of computer engineering and software development experience. She is an Episerver EMVP, (https://world.episerver.com/blogs/Allan-Thran/Dates/2018/1/7-new-emvps/) a Microsoft Certified Professional and a regular contributor on Practical eCommerce (https://www.practicalecommerce.com/author/lori-mcdonald). Her status as a recognized industry expert has resulted in regular speaking engagements at business conferences.

Follow Lori McDonald on in Linkdin (https://www.linkedin.com/in/brillweb/)

Mobile B2B eCommerce with Episerver Commerce

By Lori McDonald



(/Portals/3/digmeta/3/13/Windows-Live-Writer/Mobile-B2B-e-Commerce-with-EPiServer_642/BusinessUsers_MobileCommerce_400_2.jpg) Designing experiences for mobile is more than how the site adapts graphically to your mobile device. While the design is important, the best mobile experiences make it easy for mobile users to reach information that is of interest to them. Many eCommerce platforms are not built for this type of experience design. Episerver Commerce provides a strong set of tools to facilitate a powerful mobile B2B eCommerce site.

Why Mobile?

In a recent survey from Salesforce, 63% of B2B marketers said that mobile is core to their business (http://www.fiercecmo.com/special-reports/research-corner-2015-year-mobile-b2b-marketers). I would question why the number isn't higher. A recent article from Capterra shows relevant statistics demonstrating that executives are actively using mobile devices to research B2B purchase decisions (http://blog.capterra.com/b2b-mobile-software-trends/). In a report from Forrester Research in November 2014, they predict that 2015 will be the year that companies investing in mobile will break away from those who are not (http://blogs.forrester.com/thomas_husson/14-11-11-mobile_leaders_will_break_away_from_laggards_in_2015).

Mobile-First Design

Episerver Commerce is architected for mobile-first design. Episerver's MVC architecture enables a clean separation between the user experience and functionality. This allows usability design to occur without hearing your developers say 'It wasn't built that way.' or 'We could do that, but it will take us 4x longer and be more expensive to maintain in the future.'

With the marketing user in mind, Episerver Commerce is to make maintaining a mobile-first design easy. You can easily type your content into the page, but it will be stored without style information being attached. This enables the style to be applied based on the design that is set up within the page template for that page type. Bottom line, you don't need to worry about HTML and CSS. You can simply edit the text and drop in the images. You are also able to quickly preview what your page will look like on different devices.

See how easy Episerver is to use

Personalized Content

Manufacturers and B2B distributors have unique challenges when it comes to eCommerce. You are selling to someone based on your relationship with them. Certainly, many consumers have a relationship with the company they are buying from, but this takes on a different form in a B2B purchase. A B2B customer's relationship is something to be protected. The relationship with your distributor and sales rep needs to be protected. This means that when a customer comes to your site, they need to see their specific pricing, their specific products, and experience your website in a way that is personalized for them.

While there are ways to get this done with other eCommerce platforms, Episerver was designed for personalization. Even to the extent that the site can dynamically personalize the B2B eCommerce experience (/resources/drive-sales-through-personalized-content-with-episerver-commerce/) based on actions the user takes on that visit.

When it comes to mobile B2B eCommerce, your mobile experience needs to offer this same level of personalization and customized value as your desktop experience. Episerver allows you to serve this same personalized experience for both your mobile and desktop users.

Content and Commerce

Using related content to drive more eCommerce revenue is a missed opportunity for B2C and B2B companies, alike. Content provides users helpful information that will assist them in their purchase research and build you up as a value-added supplier. Content also provides rich fodder for search engines to index and display your site to users who are searching for information on your products.

Manuals, Material Safety Data Sheets (MSDSs), and technical specs provide a unique opportunity for manufacturers to provide helpful content to their customers. In many cases, manufacturers struggle to organize this data in an easy to navigate way on their site. Even if they accomplish this it is rarely usable from mobile devices.

This challenge is made more difficult by most of the eCommerce and CMS platforms on the market. In general, you will find software that does eCommerce well or does content management well. It is quite rare to find software that does both well. Episerver does exactly that – content and eCommerce in a single platform, optimized for mobile devices and personalized for your end user.

Bringing this back to the need for mobile B2B eCommerce, you want to make it simple for users to find your manuals and tech specs, and quickly purchase your products, whether they are on your site from a tablet, phone or desktop. Each of these experiences may be customized, depending on your audience and the top priorities you identify for the users of different devices, but each should enable your customers to get needed content and order products from whatever device they are on.

Questions?

Are you looking to implement mobile B2B eCommerce and not sure where to start? We would love to learn more about your situation and help however we can. Fill out our free consultation form (/free_consultation/) or call us at 414-425-4069 x3.

About Lori McDonald

Lori graduated from Purdue University with a Bachelor's degree in Computer-Electrical Engineering and leads Brilliance Business Solutions with over 20 years of computer engineering and software development experience. She is an Episerver EMVP, (https://world.episerver.com/blogs/Allan-Thran/Dates/2018/1/7-new-emvps/) a Microsoft Certified Professional and a regular contributor on Practical eCommerce (https://www.practicalecommerce.com/author/lori-mcdonald). Her status as a recognized industry expert has resulted in regular speaking engagements at business conferences.

Follow Lori McDonald on in Linkdin (https://www.linkedin.com/in/brillweb/)

From the Epi Blog:

Magento vs. Episerver: Why Choose Episerver for Your B2B Ecommerce Site

Monday, October 30, 2017
Lori McDonald

I believe Episerver is a better choice than Magento for B2B companies looking to grow strategically.

There is no denying that Magento is a strong player in the ecommerce market. Magento has significant market share and a large community of developers. Magento Enterprise has strong ecommerce functionality. It can be customized to meet a variety of needs, and it offers a marketplace of extensions that enable you to enhance your store with functionality that has already been developed by others.

So why not choose Magento? I believe that a platform needs to meet the needs of all parties involved. So let's look at this from a few different points of view:

For Marketers: Content marketing is known as the most popular and most effective tactic for B2B marketers. (Source: Forrester). Content builds relationships. Relationships are critical in the world of B2B ecommerce. Magento is not a Content Management System. Can you edit content in it? Sure. Is it built for marketers strategically using content to grow their business? No. To leverage content with Magento you will likely use extensions with limited content capabilities, or you will stand up another web platform next to it.

For IT: Extensions may have conflicts with one another. Extensions may not be compatible with your next upgrade. As you customize Magento for your company's needs, you will increase the costs to maintain and upgrade it in the future. Your team will need to provide technical support for multiple platforms due to the fact that your company will need other tools to handle content, intranets and other web portals.

For End Users: The user experience will be limited by the platform. Improvements in the user experience increase technical debt, increasing the costs of maintaining the system overall. Assuming another platform is being used to handle content, users will likely see a disconnect between the content and commerce. This may evidence itself in the following ways: menu navigation changes between the commerce site and content site, marketing content doesn't lead users into related products, search results won't bring back products and content; or multiple logins may be needed to interact with access to company resources.

https://www.episerver.com/learn/resources/blog/guest-blogger-lori-mcdonald/magento-vs.-episerver-why-choose-epi-for-your-b2b-ecommerce-site/

For Business Owners: Magento Enterprise's licensing is revenue-based. While there can be a rationale for this, as a business owner myself, I don't like it. A major selling point of ecommerce is enabling companies to leverage ecommerce for profitable growth. As you increase your revenue, I would rather not be giving that up to my ecommerce platform. In addition, maintaining customizations and managing extensions through upgrades will be labor intensive. Instead of using development dollars to innovate, you end up using development dollars to keep your platform alive.

Why is Episerver a better fit for B2B Ecommerce?

For Marketers: Episerver was created around content management and offers powerful marketing capabilities that are intuitive to maintain. Episerver is in its third year as a leader in the Gartner Magic Quadrant for Web Content Management, and it is one of only 2 vendors in the top 10 (with Oracle) of the Gartner Magic Quadrant for Digital Commerce that offer native web content management capabilities. Episerver's best-of-breed approach supports connections with leading marketing platforms, like Marketo, Salesforce (CRM and Marketing Cloud), IBM Marketing Cloud (Silverpop), Microsoft Dynamics CRM, Oracle Eloqua and Hubspot. In addition, Episerver's focus on leveraging machine learning to personalize the user experience means less work for marketers with better results.

For IT: You can build every web platform your company needs on Episerver (public site, distributor site, vendor portal, etc.). With fewer systems to maintain, your job becomes easier. Episerver offers cloud and on-premise licensing options, offering you the ability to select the path that makes the most sense for you. Episerver's API enables other company applications to be driven from the data stored in Episerver. Your developers will love the architecture of Episerver. The product is technically elegant, which is one of the reasons our developers love Episerver.

For End Users: Episerver's personalization features (i.e. visitor groups, personalized product recommendations, and personalized search results) give users an experience that was built just for them – with minimal maintenance time for your marketing and development teams. Even better, you can design the experience that will be best for your user, not starting from what already exists on a given platform or extension, but simply considering what interface will be best for them. The technology gives freedom to user experience design.

For Business Owners: Upgrade costs are minimized through a strong technical architecture. You spend money on functionality that drives revenue instead of keeping your platform alive. Episerver is a platform that enables you to innovate and to compete strategically. You are no longer simply matching the functionality that your online competition is offering, you are offering your customers resources and tools that no one else does that cause them to return to you again and again.

Looking at the Bigger Picture

Amazon is a disruptor. While this was first most noticeable in retail, it is increasingly impacting B2B markets as well. B2B has an advantage as compared to retail because the buying process is more complex. This makes it more difficult to recreate online. Difficult, but certainly not impossible. Amazon Business is competing for B2B business as well.

At the end of the day, no one can out-Amazon Amazon. If you want to compete through online commerce, you need to think about how to play the game differently. I believe that content is one of the key strategies for doing that. How do you believe you can use the web to compete in a new way? If content will play a role in your strategy, I would recommend including Episerver in your platform research.

https://www.episerver.com/learn/resources/blog/guest-blogger-lori-mcdonald/magento-vs.-episerver-why-choose-epi-for-your-b2b-ecommerce-site/

Why Episerver May Be Your eCommerce Solution, And How To Launch It Quickly

By Drew Douglas

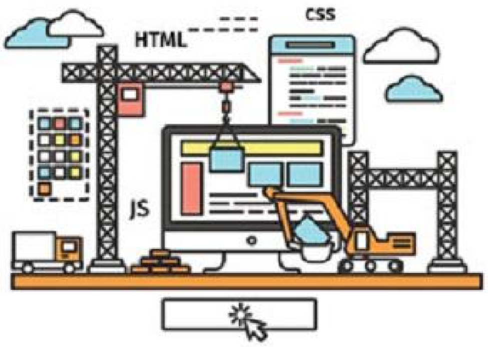
“I’m not large enough for an enterprise platform” is something we often hear customers say. That may be true: perhaps your organization is still developing an eCommerce strategy, or you know for certain that you “just need a buy button” for your products that are marketed elsewhere.

But, what happens when you start to succeed? Your needs change as you grow, and your eCommerce solution should provide the flexibility to adapt to your business.

As a developer at Brilliance who works closely with eCommerce solutions of all types, I frequently talk to customers who are limited in their choices because of their platform. In order to add new features or explore new channels or markets, these customers are forced to pursue costly workarounds or contract with outside vendors.

Before you choose an eCommerce platform, consider the architecture of the software (https://en.wikipedia.org/wiki/Software_architecture) and the potential for technical (/resources/minimizing-technical-debt-in-b2b-ecommerce/) debt. (http://martinfowler.com/bliki/TechnicalDebt.html)

A Framework For The Future: Good Architecture vs. Technical Debt



As a solution grows in complexity (more features are added) **good architecture becomes increasingly important because it helps keep the cost of new features low.**

Without good architecture, a project incurs technical debt, which means future development is more difficult (and costly) because developers have more decisions to make (https://en.wikipedia.org/wiki/Decision_fatigue) when they work with poorly designed solutions. Even though technical debt means job security for developers like me, we hate it because of the additional difficulty of finding good solutions to problems.

People argue about what exactly constitutes good architecture (https://www.google.com/search?&q=good%20software%20architecture), but its absence can be obvious: cost overruns, project delays, and obnoxious comments from developers are key signals that the project’s design is too complex or doesn’t adequately cover all the problems that need to be solved.

One principle of design that simplifies things is to have a **single point of truth** for each concept. That is, if you want to be sure of something, then there is only one place you need to check. To use a data example: the email address for a customer should be stored in only one place. The email may get displayed in different parts of the site, but when the customer changes their email, there is only one place it has to be saved.

The same is true not just for data, but also for concepts in code. For example, a customer’s true discount should only be calculated in one place. We may multiply an item’s regular price by a percentage in order to show the user how much they save, but there’s only one piece of code that calculates the final price when the user adds the item to their cart. This describes a “factored” solution where each part does something simple, but the parts working together can do something complex. (For any math geeks - think back to middle school algebra. It’s like factoring an expression with variables to break it into its simpler parts: x2 + x can be factored into x(x + 1).)

Now, using a framework doesn’t automatically give you good design (only a competent software/systems architect can do that). Rather, a framework allows you the freedom to use what you need for your business without forcing you into bad design choices for expediency’s sake. If there is a part of the framework that just doesn’t make sense for your project, you’re free to ignore it.

On the other hand, a prepackaged solution doesn’t necessarily lead to poor design. But, when problems arise that a prepackaged solution doesn't address, it's often difficult to solve one problem without creating another now or in the future. The temptation to incur technical debt is greater because finding a good design solution often means working with a core component of the platform. (/EPiServer/CMS/?language=en#context=epi.cms.contentdata:///1510)

The Episerver Framework

The framework solution that we prefer to work with at Brilliance is Episerver. We like it because it's a well-thought-out and mature framework for creating and displaying web content. Its eCommerce components are adaptable to almost any business case, which are particularly important to our B2B clients.

Episerver was initially built as a CMS and has tons of really helpful features for content creators, editors, and marketers. Episerver's developers have done a great job of creating useful abstractions around content (products, categories, blog posts, landing pages) and technical details (database access, cache management, cloud hosting) that allows us to focus on solving business problems.

These terms may or may not mean much to you, but being able to solve the problems you have with eCommerce does! Episerver incorporates proven technologies but still allows us the flexibility to add new ones.

Getting Started Fast

Buying an Episerver license gets you the framework, and then you have to build your site on top of that. At Brilliance, we understand that some clients will want a clean slate to work with, but others will be better off starting with a basic design. That's why we created Catalyst (/technology/episerver/)for Episerver Commerce.

Catalyst is a starter kit for Episerver that includes a set of templates and features robust enough for a B2B site with complex client needs, but also easily configurable for simpler B2C style sites. We've added features for account management, complex pricing, and customer self-service. What we haven't done is locked our clients into a particular payment or fulfillment system.

It's like getting some of the benefits of a prepackaged solution while keeping the architecture and flexibility of the framework. Plus, it saves a lot of time and money in the development process.

If you want to build a site with the flexibility to grow with your business and still get it up and running quickly, Catalyst is the way to go.

Schedule a free consultation (/technology/episerver/) to find out how a solution built on the Episerver framework can meet and adapt to the needs of your business and your clients now and for years in the future.

About Drew Douglas

Drew is an Episerver Certified, as well as a Microsoft Certified Professional (Microsoft HTML5 with JavaScript & CSS3 Certification). As a Lead Developer at Brilliance, Drew serves as a technical lead and architect on Episerver projects. During his career, his development work has utilized many technologies, including ASP.NET Forms and MVC, NServiceBus and React. His current focus is B2B eCommerce development with ERP integrations.

Follow Drew Douglas on **in** Linkdin (https://www.linkedin.com/in/ddrewdouglas/)

The Underutilization of eCommerce Software Investments

By Lori McDonald

I just got back from Episerver Ascend 2018, which was their biggest annual conference yet. Brilliance had five members of our team there, and it was a valuable investment. We learned from Episerver, market leaders, Episerver customers and other Episerver partners.

Underutilization of software investments

One of my takeaways from the event was how often the tools we invest in are underutilized. Based on industry research, only 25%-50% of software features are utilized following a purchase. Based on my own experience, I have found this to be true, not only for Episerver but for many platforms, especially when you have a mature software platform focused on continuous improvement.

Episerver is a platform that has devoted itself to making it easy for marketers to create effective digital experiences for their customers - in any channel and on any device. As a part of that mission, they are constantly innovating, creating new tools and offerings to improve customer loyalty and increase sales – both your average order size and your customer lifetime value.

Whether you are an Episerver customer today, or on a different platform, what gap exists between what your platform is capable of and how you are using it? What new strategies do you want to test and measure? How can you continually improve the capabilities of your site?

Let me share some ideas of ways you could be driving more revenue through your site, leveraging the investment you have already made. While this list is created from features that Episerver offers, you can look to implement these ideas no matter what platform you are on.

Included in Episerver's Base CMS Offering

- **A/B testing** – Setting up A/B tests on your site is a great way to gather data to make improvements and get better results. You define the conversion goals you want to measure like landing page opt-ins, add to cart clicks, site stickiness, average order, or specific product purchase (you can create custom conversion goals as well). With Episerver, there are no additional tools needed, and no Javascript for a 3rd party integration that will add to your site's load time. You can test a single page or a block that is used on several pages. It can automatically start using the winning variation when it determines the winner. This enables you to not guess what will create value for your site visitors (and for your company) but to use data to determine what the best choice is.

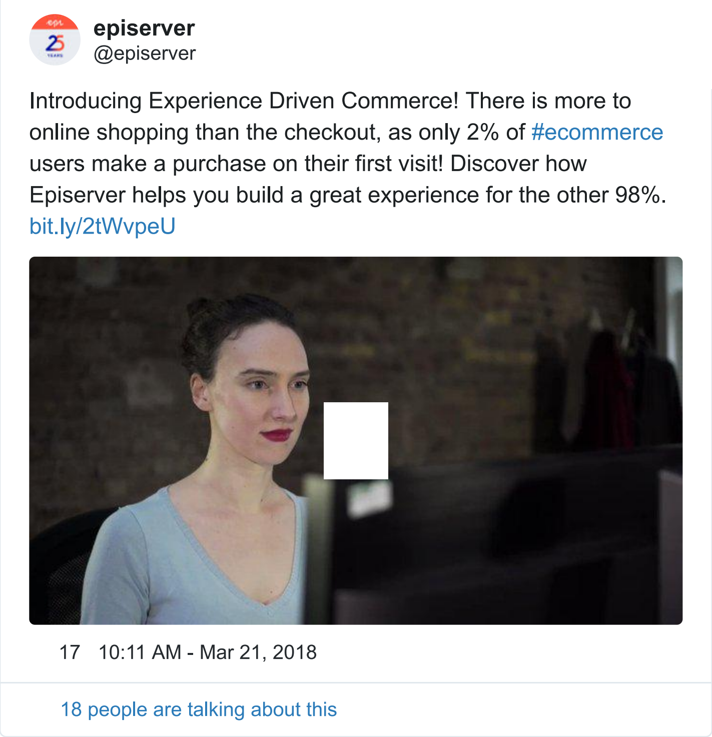
- **Visitor groups** are one of the reasons I first fell in love with Episerver back in 2013. You can define rules that will dynamically segment your users based on a variety of criteria (i.e. what page they entered your site though if they clicked on a certain page if they came to your site from Facebook, how much money they have spent with you, etc.). Then, you can use this grouping to display information they are more likely to be interested in. By making a user's experience more relevant, you will increase customer loyalty and your customer's lifetime value.

- **Automatic Landing Pages** enable you to quickly create landing pages using tags that dynamically serve the correct content without you having to create separate pages every time. Work with your development team to plan the landing pages you want and Episerver allows you to save time and create effective landing pages efficiently.

Additional Episerver Offerings that Drive Value

Episerver has been innovating and has acquired and created several additional products that may be purchased separately or can be bundled together to save you money. While these items may not be a part of your current Epi license, they leverage your existing investment to make it much more powerful and demonstrate a higher ROI.

Check out this video for a quick overview of the experience you can create for your customers:



episerver
@episerver

Introducing Experience Driven Commerce! There is more to online shopping than the checkout, as only 2% of #ecommerce users make a purchase on their first visit! Discover how Episerver helps you build a great experience for the other 98%. bit.ly/2tWvpeU

17 10:11 AM - Mar 21, 2018

18 people are talking about this

- **Product recommendations (Perform)** is driven by machine learning to determine which products your visitors will be most interested in. The "recommendations engine" is constantly improving itself based on users' actions. It isn't just computing recommendations for people in general on your site, but for each individual session, it is calculating recommendations specific to them. This has a clear return on investment that can be demonstrated, yet many companies aren't using it.

- **Personalized content (Advance)** uses machine learning to recommend content to your visitors, pointing them to other pages on your site that would be of interest. Why is this so cool? Everyone loves the idea of personalization, but few marketers have the time to invest in figuring out the best way to carve up user segments and what content is the best to recommend. That is why visitor groups are often underutilized. In the future, Episerver will provide marketers with recommended content and images suggested in the editor console when they are setting up new pages.

- **Email personalization (Reach)** uses product recommendations to personalize the emails you are sending out. Transaction emails (i.e. order receipt, shipping notification) typically have a much higher open rate (/resources/successful-email-marketing-strategies-in-the-b2b-space/) than promotional emails. Why not recommend other products they would like along with their order receipt? When you send promotional emails, what if you recommended products along with your promotional offer? There are pre-built integrations for several email service providers so you can keep the one you have.

- **Email / Marketing Automation (Campaign)** enables you to use Episerver as your email service provider, keeping all your marketing data and communications in one place. It provides workflow rules to trigger emails based on user's behavior, what segment they are in, what they clicked on in a previous email, and more.

- **Personalizing Search results (Personalized Find)** allows you to show different results to different users who both search for the same thing. When someone searches for "pink shoes" they see different results or a different order so the items most relevant to them will be at the top. While this only personalizes product searches today, personalizing content in your search results is on the roadmap.

Drive More Revenue

When you make an investment in a comprehensive eCommerce platform like Episerver and others, there's a lot of capability there. You have a lot to gain from the investment you have already made.

How can you capitalize on the tools you have to drive more revenue? What improvements can you make?

If you are interested in taking a look at how you can take ongoing incremental steps towards driving more revenue, schedule a call with us (/free_consultation/). We are happy to help!

About Lori McDonald

Lori graduated from Purdue University with a Bachelor's degree in Computer-Electrical Engineering and leads Brilliance Business Solutions with over 20 years of computer engineering and software development experience. She is an Episerver EMVP, (https://world.episerver.com/blogs/Allan-Thran/Dates/2018/1/7-new-emvps/) a Microsoft Certified Professional and a regular contributor on Practical eCommerce (https://www.practicalecommerce.com/author/lori-mcdonald). Her status as a recognized industry expert has resulted in regular speaking engagements at business conferences.

Follow Lori McDonald on in Linkdin (https://www.linkedin.com/in/brillweb/)