BRIGHTEDGE AUTOPILOT

SELF-CONNECTING PAGES
SERVER-SIDE IMPLEMENTATION GUIDE

Abstract

Implementation guide for AutoPilot Self-Connecting Pages with the BrightEdge AutoPilot Server-Side SDK

BRIGHTEDGE

be_implementation@brightedge.com

TABLE OF CONTENTS

BACKGROUND

AutoPilot Self-Connecting pages is a unique technology offering from BrightEdge that empowers marketers to fully optimize internal links with no ongoing manual effort. Using BrightEdge AI, the most valuable pages will help lift the entire site to expand overall organic visibility.

Self-Connecting Pages is different from other internal linking approaches as it specifically draws from the wealth of SEO data in the BrightEdge Data Cube to inform what to link to. Other internal linking approaches will usually follow a taxonomy, personalization, or information architecture driven process that does not directly unlock the upside for your organic search traffic. The outcome is a smart internal linking solution that removes the guesswork and subjectivity of internal linking in a resource efficient, automated, zero maintenance and results oriented way.

Self-Connecting Pages is built on a pre-publishing framework that offers low latency and does not require the collection of Personally Identifiable Information (PII) or pixel tracking.

Configuration is a core pillar to the solution design which provides a software defined governance of “swim-lanes” or business logic that impacts how various sections of your website should or should not interlink.

This guide provides an overview of the server-side SDK implementation process including creative considerations and sample code. Technical aspects are discussed including architecture and performance. Finally, frequently asked questions may be a helpful resource for completeness.

WHY SERVER-SIDE?

A server-side integration is the most robust and performant solution design for BrightEdge AutoPilot and the recommended integration path for customers.

The key advantages of a server-side integration are:

- **MINIMAL LATENCY** - compiled code leading to faster code execution and high-performance endpoints - **~50-70ms response time.**
- **EXTENSIBILITY** - control creative directly, support multiple creative variants within a single integration, consume data as JSON, and merge with other business data.
- **CACHING** – sits behind and leverages your own caching policies

Our Software Development Kit (SDK) offers a convenient and low effort implementation mechanism that is designed to be interoperable agnostically to the end web content management system. The SDK is currently supported across PHP, .NET, and Java programming languages.

The server-side SDK features built-in support for:

- Request/Response handling to the BrightEdge solution infrastructure
- Error handling: including timeout, network errors, invalid data responses
- Endpoint capsule response parsing
- Convenience functions to simplify response output to front end layers
- Debugging

IMPORTANT NOTE: The server-side SDK implementation approach presumes that the integration is being performed within the response rendering cycle on your web application origin servers, and with a line of sight into the full URL for any given request. If you are running a headless CMS or a serverless architecture, the SDK will likely not be applicable, alternative approaches can be explored with your BrightEdge team. This guide will still provide valuable insight into the design patterns, solution design, and technical details that would still be relevant.

IMPLEMENTATION PROCESS

The implementation process for Self-Connecting Pages follows a tried and tested approach that has clients go live within 4-6 weeks on average. The timeline is principally driven by customer development availability and existing backlog prioritization rather than the complexity of the integration itself.

Figure 1: AutoPilot Self-Connecting Pages Implementation Process Flowchart

1

TECHNICAL OVERVIEW & KICK-OFF: The AutoPilot Solutions team hosts a kick-off to perform discovery on your technical environment and process, stakeholders, and solution design considerations. We review the SDK integration code samples with your engineering resources and identify a path to integration that fits your infrastructure stack.

Activities:

- BrightEdge provides customized code samples to guide the integration based on your environment.
- BrightEdge establishes a recurring weekly cadence call to facilitate the integration process towards Go-Live.
- Discuss customer's desired creative treatment and whether customer or BrightEdge will manage the styling.

Outcomes:

- BrightEdge provisions your initial Self-Connecting Pages instance.
- The customer provides a staging hostname to support product configuration and staging validation efforts.
- (Optionally) BrightEdge provides a creative treatment for feedback and iteration. Alternatively, the customer provides styling specs if BrightEdge controls styling or begins creative development internally.

2

STAGING VALIDATION (Parallel Track): Once the SDK integration code is implemented in a lower environment – such as staging – the AutoPilot team reviews and validates the proper communication flows between your infrastructure and AutoPilot endpoints. This is often conducted over a screenshare or by the customer sharing a public facing staging URL.

Activities:

- The SDK integration is validated to properly return and display the desired link block creative treatment using static default links set by the AutoPilot team.

- BrightEdge shares sample Test URLs for the staging environment to review and surface any issues with the integration.
- BrightEdge performs any creative adjustments necessary (if using an in-line styling approach managed by BrightEdge)

Outcomes:

- The technical integration and creative treatment are approved to be scheduled for deployment to production.

3 **LINK BLOCK CONFIGURATION (Parallel Track):** The AutoPilot Integration Specialist provides a recommendation and/or initial configuration of the product based on discovery insights, analysis of your website architecture, and performance opportunities.

Activities:

- Profile segmentation is applied to the product configuration.
- Default Links are provided by the customer, subject to segmentation discussions.
- Undesirable URLs are archived as needed.
- Link anchor text is sanitized, and the source element agreed upon.

Outcomes:

- Product configuration is iterated upon in preparation for Go-Live.

4 **PRODUCTION DIAGNOSTICS:** Further to the deployment of the SDK into production, the product is technically integrated but runs in diagnostic mode. This state allows for selective sample Test URLs to be created by the AutoPilot Integration Engineer to validate the proper and expected functioning of the solution in production. In this mode, the integration is not publicly visible in your front-end experience.

- **Activities:**
  - Validation of expected solution design and functioning
- **Outcomes:**
  - Customer approves transition to Go-Live stage

5 **GO LIVE:** BrightEdge performs the final QA and deploys your configuration.

- **Activities:**
  - The BrightEdge AutoPilot team runs a final analysis and deployment to render the link blocks live into production.
- **Outcomes:**
  - BrightEdge provides a representative set of screenshots of the live state to the customer to memorialize the moment.

CREATIVE OVERVIEW

The core requirement for the Self-Connecting Pages solution to perform is to present the algorithmically determined links as standard HTML HREF links within the page source code, in a timely manner to guarantee search engine crawler consumption.

The straightforward nature of this requirement offers a large range of customization opportunities to the end customer experience. However, **it is important to note that the primary audience of the solution is search engine crawlers** and as such a simple creative is essentially all that is truly required.

PRIMARY CREATIVE CONSIDERATIONS

There are two fundamental aspects to the creative articulation of the Self-Connecting Pages solution: **Placement and Structure**.

STRUCTURE

The core structure of a Self-Connecting Pages link block can be reduced to the following elements:

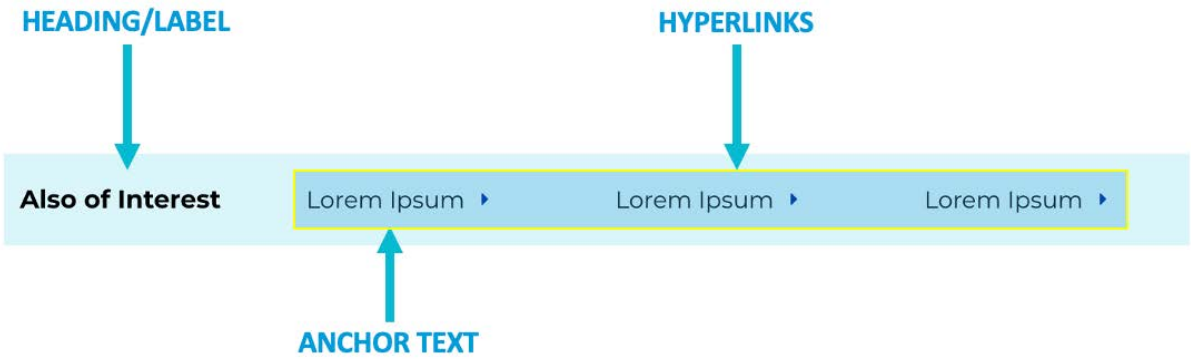

Figure 2: Elements of Creative Structure

- **Heading/Label** – frames the content from a UX perspective. E.g.: Also of Interest, Related Pages, Popular Searches…
- **Hypertext Links** – a configurable number of links delivered to the User Experience. These are established by the AutoPilot algorithms.
- **Anchor Text** – the actual text of the hypertext links that are principally sourced from the destination's page H1 heading or Page Title.

PLACEMENT

The placement or location of the link block is guided by UX and Development considerations. The below provides a generic illustration of the typical options considered for a given website experience.

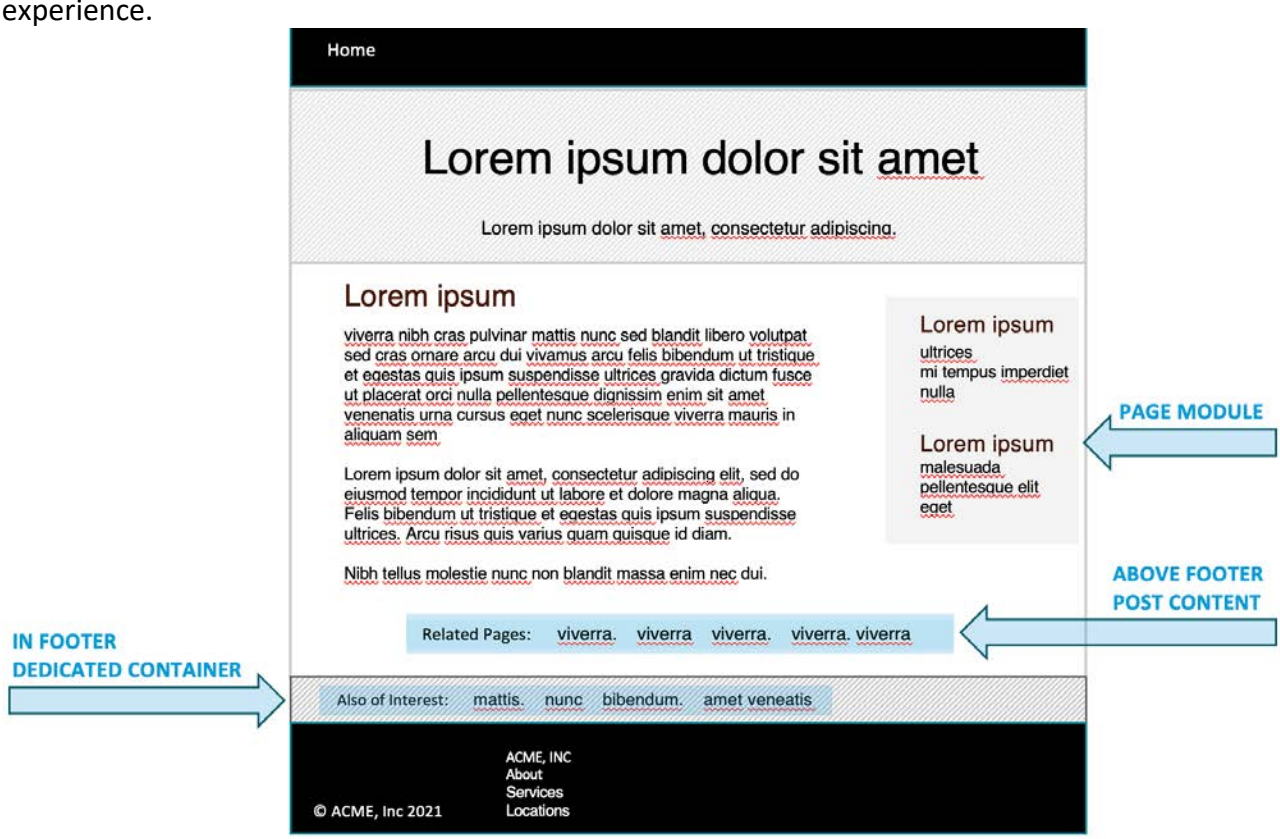

Figure 3: Typical Creative Placement Options

In practice, the dedicated container approach is overwhelmingly favored for the following reasons:

1. The AutoPilot solution is primarily deployed for the consumption by search engine crawlers, and therefore a placement away from the core visitor journey satisfies the underlying requirement for the solution's effectiveness and without compromising heavily on user experience considerations. A container just above the footer typically satisfies the use case as it is unobtrusive yet not fully obfuscated within the overall site experience.
2. From a web engineering perspective, the footer is typically a shared global template part that is re-used across a given website. This offers a single point of integration that allows the solution to be available and **the display (or not) of link blocks to be governed by the UI configuration in BrightEdge.**

Other placements illustrated are viable but warrant additional discovery in terms of the level of effort required.

The following provides a baseline example of an AutoPilot Self-Connecting Pages implementation from a creative perspective.

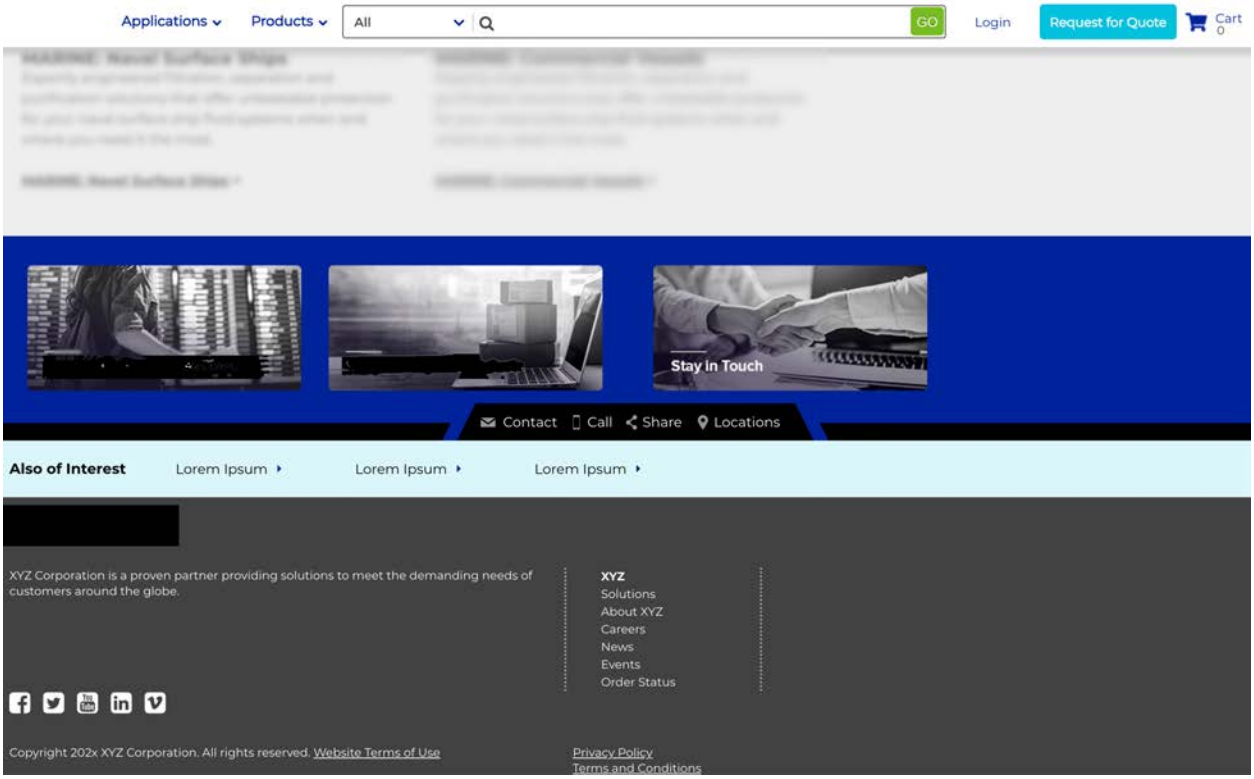

Figure 4: Baseline Example of Typical Creative

There is a range of options beyond this baseline that up-styles the creative with elements such as iconography, layouts including tiles or cards, the inclusion of images, or additional content – all of which can be discussed with your BrightEdge team to understand dependencies and best practices. Further inspiration is provided with examples shown in Appendix "Sample Creatives"

BEST PRACTICES AND LIMITATIONS

- The AutoPilot Self-Connecting Pages should be developed with responsive design principles in mind and ensure that the styling adapts to mobile viewports. If your site is not responsive or your mobile experience uses an M-dot (m.example.com) architecture, please consult with your BrightEdge team to determine the best path forward.
- A typical AutoPilot integration will start with a minimum of three (3) links; however, this can be easily increased through our UI.

  Best Practice: it is worthwhile planning creative to accommodate up to six (6) links or more from a responsiveness and fluid design perspective.

- The anchor text of the links is automatically truncated with ellipses to avoid bleeding over. The truncation threshold is configurable on a per-profile or global basis and is typically in the region of 50 characters.

  Best Practice: Plan creative to accommodate the scenario that all the links to be displayed are at the maximum threshold (e.g.: 50 characters) to mitigate user experience issues down the line.

  Best Practice*:* The class name “be-ix-link-block” is recommended to be retained with any customization to facilitate monitoring and observability of the integration by BrightEdge.

  Limitation: Only a single creative structure and styling can be supported if BrightEdge is managing the delivery of the link block creative.

  Limitation: There can be only a single link block component per URL, BrightEdge does not currently support multiple link block components per URL.

STYLING MANAGEMENT AND DELIVERY

There are three routes to manage the styling of the Self-Connecting Pages link blocks:

1. **BRIGHTEDGE MANAGED**
    - Styling is inserted in-line through our API responses
    - Entails that any creative changes after go-live need to be routed through your BrightEdge team.

    IMPORTANT: only a single creative treatment is supported with this option, variants across different pages/templates are not possible.

2. **CUSTOMER MANAGED – Barebones HTML**
    - BrightEdge's API response will return the link blocks structured as vanilla HTML content with agreed upon class names. See SAMPLE CAPSULE RESPONSE (HTML FORMATTED)
    - Customer manages styling through their own CSS stylesheets.

```
<div class="be-ix-link-block">
    <div class="be-related-link-container">
        <div class="be-label">Also of Interest:</div>
            <ul class="be-list">
                <li class="be-list-item"><a class="be-related-link" href="">Lorem ipsum dolor</a></li>
                <li class="be-list-item"><a class="be-related-link" href="">Ut enim ad minim</a></li>
                <li class="be-list-item"><a class="be-related-link" href="">Duis aute irure esse cillum</a></li>
            </ul>
    </div>
</div>
```

Figure 5: Sample Barebones HTML Response

IMPORTANT: Only a single creative structure is supported with this option, however light touch variants of the styling (color/padding/font, etc.) can be readily performed by the customer within their own CSS stylesheets.

3. **CUSTOMER MANAGED – JSON**
    - BrightEdge's API response returns structured JSON data with the link blocks to render. See SAMPLE CAPSULE RESPONSE (JSON FORMATTED STRING)
    - The customer consumes JSON to render the link block. Structure and styling are in full control of the customer.
    - This option provides the most flexible path to future enhancements and allows for complete control to build variations across the customer templates/website.

```
{
  "account_id": "f00000000012345",
  "date_published": 1635531464015,
  "key": "9999651116",
  "is_universal_capsule_enabled": "false",
  "capsule_version": "1.0.0.0",
  "engine_version": "1.0.0.0",
  "version": 2,
  "date_created": 1635531464015,
  "nodes": [
    {
      "feature_group": "body_1",
      "engine_version": "1.0.0.0",
      "content": "{\"url\":\"https://www.example.com/products/xyz\", \"anchor_text\":\"Vulputate enim nulla\"},{\"url\":\"
          https://www.example.com/products/abc\", \"anchor_text\":\"Pretium nibh ipsum\"},{\"url\":\"https://www.example.com/
          industries/real-estate\", \"anchor_text\":\"Ultricies integer quis\"},{\"url\":\"https://www.example.com/products/
          flagship\", \"anchor_text\":\"Ut eu sem\"},{\"url\":\"https://www.example.com/solutions/industries/mortgage\", \"
          anchor_text\":\"Lorem Ipsum Dolor\"}\n",
      "date_published": 1635531464015,
      "date_created": 1635531464349,
      "type": "bodystr",
      "publishing_engine": "link-block"
    },
    {
      "feature_group": "_head_open",
      "engine_version": "1.0.0.0",
      "content": "\n\n",
      "date_published": 1635531464015,
      "date_created": 1635531464409,
      "type": "headstr",
      "publishing_engine": "pixel"
    }
  ],
  "config": {},
  "publishing_engine": "ixf-compiler"
}
```


Figure 6: Sample JSON Response

The delivery mechanism should be established early in the development process as the integration code makes its way into lower development environments. The method of delivery is controlled by the BrightEdge AutoPilot Integrations team.

TECHNICAL ARCHITECTURE

Below is a high-level diagram of the AutoPilot Self-Connecting Pages architecture as integrated into a given customer environment. The server-side integration will typically take advantage of your existing web caching infrastructure and only make calls to the BrightEdge CDN infrastructure when out of cache.

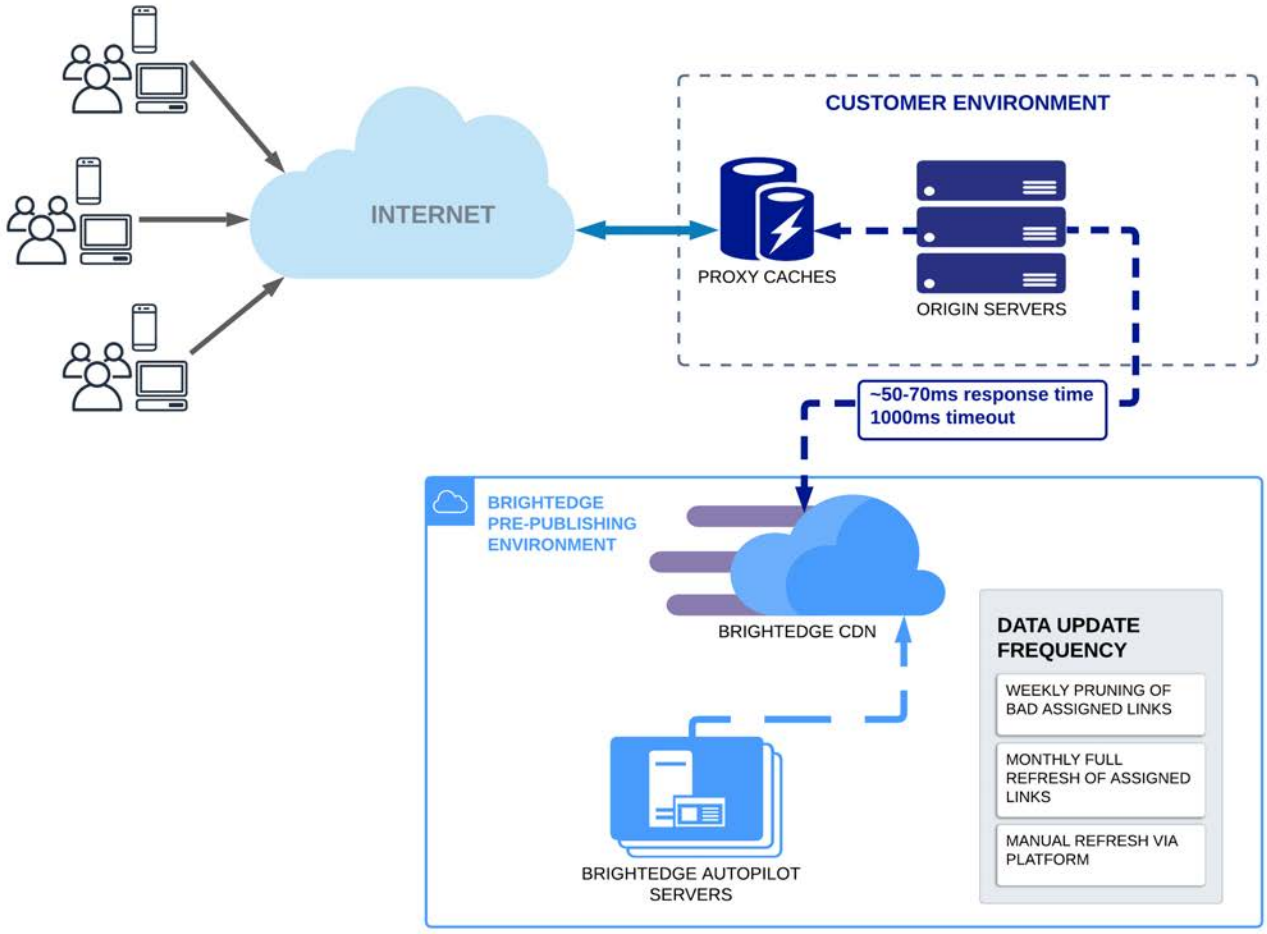

Figure 7: BrightEdge AutoPilot Technical Architecture Diagram

PRE-PUBLISHING ENVIRONMENT

The BrightEdge managed pre-publishing environment establishes the infrastructure that hosts our pre-published, pre-computed data on high performance CDN endpoints. This data is partitioned by customer and provides what we term “capsule URLs” which are relied upon by the AutoPilot solution. Each capsule is typically scoped to carry a data payload that is specific to a given customer URL. The server-side SDK effectively governs the response and request flow to these capsule URLs to integrate the solution into a customer site.

SAMPLE CAPSULE URL

Under the hood of the SDK is a simple procedure that maps the request URL from the webserver to a Capsule URL. The Capsule URL comprises of three elements:

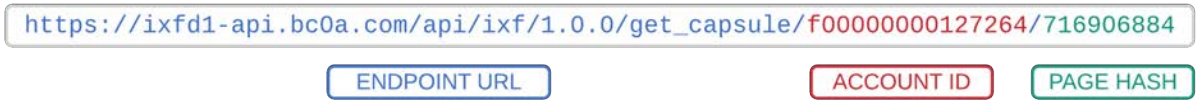

Figure 8: Capsule URL Convention

1. **Endpoint URL**: HTTPS URL to our CDN service endpoint.
2. **Account ID**: Integration specific account identifier provided to you by BrightEdge for your integration. This account ID is bound to a specific BrightEdge S3 Account to access its underlying resources for analysis and relevance.
3. **Page Hash**: An MD5-like hash of the request URI which hosts the unique URL’s computed linking data.

The Capsule URL will return a JSON response to a GET request which among other utility and debugging attributes, provides two node array items:

1. A “Head open node” which contains the relevancy tracking pixel code as JavaScript (if enabled) and CSS styling code (if enabled). This node is the same for all Capsule URLs.
2. A “Body node” which contains the computed link block data and is used for displaying the Self-Connecting Pages solution. These are unique per Capsule URL.

SAMPLE CAPSULE RESPONSE (HTML FORMATTED)

If configured by BrightEdge to respond with a pre-published HTML response, the content attribute of the Body node will return a string of escaped HTML using the styling as agreed upon. The SDK processes and injects the content attributes of the capsule nodes to render the assigned links on a given URL.


```
{
  "account_id": "f00000000127264",
  "date_published": 1594672603797,
  "key": "2095086657",
  "is_universal_capsule_enabled": "false",
  "capsule_version": "1.0.0.0",
  "engine_version": "1.0.0.0",
  "version": 2,
  "date_created": 1594672603797,
  "nodes": [
    {
      "feature_group": "body_1",
      "engine_version": "1.0.0.0",
      "content": "<div class=\"be-ix-link-block\"><div class=\"be-related-link-container index-enjory\"><h3 class=\"be-label\">Also of
      Interest</h3><ul class=\"be-list\"> <li class=\"be-list-item\"><a class=\"be-related-link\"
      href=\"https://www.brightedge.com/info/content/benefits-ab-testing-and-8-recommendations\">benefits-ab-testing-and-8-r...</a>
      </li> <li class=\"be-list-item\"><a class=\"be-related-link\" href=\"https://www.brightedge.com/info/content/one-size-fits-all-
      organic-optimization-ends\">one-size-fits-all-organic-o...</a></li> <li class=\"be-list-item\"><a class=\"be-related-link\"
      href=\"https://www.brightedge.com/info/content/how-improve-content-ranking\">how-improve-content-ranking</a></li></ul></div>
      </div>\n",
      "date_published": 1594672603797,
      "date_created": 1594672603965,
      "type": "bodystr",
      "publishing_engine": "link-block"
    },
    {
      "feature_group": "_head_open",
      "engine_version": "1.0.0.0",
      "content": "<style>\n.be-ix-link-block .be-related-link-container .be-label{display:inline-block;font-size:inherit;font-
      weight:700;text-transform:uppercase;color:black}\n\t.be-ix-link-block .be-related-link-container .be-list{display:inline;list-
      style:none;margin:0;padding:0}\n\t.be-ix-link-block .be-related-link-container .be-list .be-list-item{display:inline;text-
      transform:uppercase}\n\t.be-ix-link-block .be-related-link-container .be-list .be-list-item .be-related-link{text-
      decoration:none;color:black}\n\t.be-ix-link-block .be-related-link-container .be-list .be-list-item .be-related-link:hover{text-
      decoration:underline}\n\n\t@media (max-width:767px) {\n\t\t.be-ix-link-block .be-related-link-container .be-label, .be-ix-link-
      block .be-related-link-container .be-list{display:block;width:100%}\n\t\t.be-ix-link-block .be-related-link-container .be-list
      .be-list-item{margin-right:0;display:block}\n\t}\n</style>\n\n    <script>\n      (function() {\n      var bec =
      document.createElement('script');\n      bec.type = 'text/javascript';\n      bec.async = true;\n      bec.setAttribute(\"data-
      id\", \"bec\");\n      bec.setAttribute(\"org-id\", \"f00000000127264\");\n      bec.setAttribute(\"domain\", \"ixf_test\");\n
      bec.setAttribute(\"session-timeout\", 86400000);\n      bec.src = document.location.protocol + '//cdn.b0e8.com/conv_v3.js';\n
      var s = document.getElementsByTagName('script')[0];\n      s.parentNode.insertBefore(bec, s);\n})();\n</script>\n",
      "date_published": 1594672603797,
      "date_created": 1594672605306,
      "type": "headstr",
      "publishing_engine": "pixel"
    }
  ],
  "config": {},
  "publishing_engine": "ixf-compiler"
}
```


Figure 9: Sample HTML formatted Capsule URL response

SAMPLE CAPSULE RESPONSE (JSON FORMATTED STRING)

If your account is configured to provide a JSON formatted response, the Body node will return a content attribute with a JSON formatted string.
The content attribute is an array of links with associated metadata for manipulation in your integration code.

Each link array item is comprised of the following attribute-value pairs:

1. **url**: the URL to link to.
2. **h1**: the anchor text to display for the link destination.
3. **desc**: (optional) additional content for enhanced integrations – typically sourced from the destination URL's meta description.


```
{
    "account_id": "f00000000127264",
    "date_published": 1594685768747,
    "key": "716906884",
    "is_universal_capsule_enabled": "false",
    "capsule_version": "1.0.0.0",
    "engine_version": "1.0.0.0",
    "version": 2,
    "date_created": 1594685768747,
    "nodes": [
        {
            "feature_group": "body_1",
            "engine_version": "1.0.0.0",
            "content": "\n[{\"url\":\"https://www.brightedge.com/info/content/brightedge-retail-
            marketing\",\"h1\":\"brightedge-retail-marketing\",\"desc\":\"\"},
            {\"url\":\"https://www.brightedge.com/info/content/one-size-fits-all-organic-optimization-ends\",\"h1\":\"one
            size-fits-all-organic-optimization-ends\",\"desc\":\"\"},
            {\"url\":\"https://www.brightedge.com/info/content/benefits-ab-testing-and-8-
            recommendations\",\"h1\":\"benefits-ab-testing-and-8-recommendations\",\"desc\":\"\"}]\n",
            "date_published": 1594685768747,
            "date_created": 1594685768914,
            "type": "bodystr",
            "publishing_engine": "link-block"
        },
        {
            "feature_group": "_head_open",
            "engine_version": "1.0.0.0",
            "content": "\n    <script>\n      (function() {\n      var bec = document.createElement('script');\n
            bec.type = 'text/javascript';\n      bec.async = true;\n      bec.setAttribute(\"data-id\", \"bec\");\n
            bec.setAttribute(\"org-id\", \"f00000000127264\");\n      bec.setAttribute(\"domain\", \"ixf_test\");\n
            bec.setAttribute(\"session-timeout\", 86400000);\n      bec.src = document.location.protocol +
            '//cdn.b0e8.com/conv_v3.js';\n      var s = document.getElementsByTagName('script')[0];\n
            s.parentNode.insertBefore(bec, s);\n})();\n</script>\n",
            "date_published": 1594685768747,
            "date_created": 1594685768956,
            "type": "headstr",
            "publishing_engine": "pixel"
        }
    ],
    "config": {},
    "publishing_engine": "ixf-compiler"
}
```


Figure 10: Sample JSON formatted string Capsule URL response

IMPLEMENTATION STEPS

The server-side implementation can be broken down into three steps:

- Installation of the SDK file
- Instantiation of the AutoPilot object
- Making the AutoPilot function calls within the appropriate template areas to inject and render the solution.

SDK INSTALLATION

BrightEdge will provide an SDK library (be_ixf_client.php, BEIXFSDK.dll, or be-ixf-java-sdk.jar) that needs to be added to a folder accessible to your web server process (typically theme, webroot, library, vendor folders). To provide (or compile for you) the appropriate SDK file, we need to know a few aspects of your environment against which we will integrate:

1. Language: PHP, .NET, or Java
2. Version: e.g.: PHP 7.4, .NET Framework 4.8
3. For .NET, we need to know:
    a. Framework type: Webform or MVC
    b. Newtonsoft version

SDK INSTANTIATION

Sample code for the core supported programming languages are provided hereafter. The instantiation of the SDK is typically invoked in your theme file for PHP, controller file for .NET, and class file for Java. In either event, the instantiation should occur only once per request cycle and have access to the server's REQUEST_URI variable (or appropriate server variable depending on the underlying web server language) with the full request path.

By default, all URL query string parameters are ignored to mitigate inadvertent content duplication. If you are serving query parameter URLs to search engines for indexing, the whitelist configuration variable needs to be set for each query string key separated by the pipe character (|).

```
BEIXFClient::$WHITELIST_PARAMETER_LIST_CONFIG => "ixf",                    // PHP
ixfConfig.setProperty(IXFConfiguration.WHITELIST_PARAMETER_LIST, "ixf");   //Java
ixf_config.SetProperty(IXFConfigKeys.WHITELIST_PARAMETER_LIST, "ixf");     //.NET
```

For .NET and Java implementations, we may need to set a few optional parameters (PAGE_ALIAS_URL, CANONICAL_HOST) in the controller or class file to generate the appropriate Capsule URL request.

TEMPLATE INTEGRATION CALLS

Once the AutoPilot object is instantiated, three functions should be called to make up a successful integration:

- GetHeadOpen
- GetBodyOpen/GetBodyString
- Close

In your site template (typically the header template part), you should make a function call to retrieve the GetHeadOpen in the <head> section, and the GetBodyString and Close in the location where you want the links to show up (typically in the footer template part).

LANGUAGE SPECIFIC SAMPLE CODE

IMPORTANT: The following is sample code for overview purposes only. Your BrightEdge AutoPilot Implementation Engineer will provide customized integration code following the AutoPilot Kick-Off session.

PHP SAMPLE CODE

INSTANTIATION CODE

```php
<?php
//Your access to and use of BrightEdge AutoPilot - Self Connecting Pages is governed by the
//Infrastructure Product Terms located at: www.brightedge.com/infrastructure-product-terms.
//Customer acknowledges and agrees it has read, understands and agrees to be bound by the
//Infrastructure Product Terms.

//BE IXF: save the be_ixf_client.php file to your server, then use "require" to include it in your template.
require 'be_ixf_client.php';
use BrightEdge\BEIXFClient;

//BE IXF: the following array and constructor must be placed before any HTML is written to the page.
$be_ixf_config = array(
    BEIXFClient::$CAPSULE_MODE_CONFIG => BEIXFClient::$REMOTE_PROD_CAPSULE_MODE,
    BEIXFClient::$ACCOUNT_ID_CONFIG => "enter_account_id_here",

    BEIXFClient::$API_ENDPOINT_CONFIG => "https://ixfd1-api.bc0a.com",
    //BEIXFClient::$CANONICAL_HOST_CONFIG => "www.domain.com",
    //BEIXFClient::$CANONICAL_PROTOCOL_CONFIG  => "https",

    // BE IXF: By default, all URL parameters are ignored. If you have URL parameters that add value to
    // page content.  Add them to this config value, separated by the pipe character (|).
    BEIXFClient::$WHITELIST_PARAMETER_LIST_CONFIG => "ixf",

);
$be_ixf = new BEIXFClient($be_ixf_config);
?>
```

TEMPLATE FUNCTION CALLS

```php
<head>
<?php
    //BE Head: place getHeadOpen just inside of the HTML head, used for to append SEO-related header elements.
    print $be_ixf->getHeadOpen();
?>
</head>
    <!--BE Footer: The following <div> block needs to be placed in the location where the link block will be displayed-->
    <div class="be-ix-link-block">
    <?php
        print $be_ixf->getBodyString("body_1");
        print $be_ixf->close();
    ?>
    </div>
    <!--BE IXF: end-->

```

.NET SAMPLE CODE

INSTANTIATION CODE

```
using IXFSDK;
using IXFSDK.Util;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace kentico-demo
{
    public static class BrightEdge
    {
        public static string GetContent(string URL)
        {
            return GetContent(URL, null);
        }
        public static string GetContent(string URL, string urlParameters)
        {
            IXFConfiguration ixf_config = new IXFSDKConfiguration();
            ixf_config.SetProperty(IXFConfigKeys.CAPSULE_MODE, IXFConfigKeys.REMOTE_PROD_CAPSULE_MODE);
            ixf_config.SetProperty(IXFConfigKeys.ACCOUNT_ID, "[enter_account_id_here]");
            ixf_config.SetProperty(IXFConfigKeys.API_ENDPOINT, "https://ixfd1-api.bc0a.com");
ixf_config.SetProperty(IXFConfigKeys.CHARSET, "UTF-8");
            //everything in the URL address bar
            ixf_config.SetProperty(IXFConfigKeys.PAGE_ALIAS_URL, CMS.Helpers.URLHelper.GetAbsoluteUrl(URL) + urlParameters);
            //ixf_config.SetProperty(IXFConfigKeys.CANONICAL_HOST, "www.kentico-demo.com");
            //ixf_config.SetProperty(IXFConfigKeys.CANONICAL_PROTOCOL_CONFIG, "https");

            /* If url parameters are used in the site in a way that affects page content, add
            those content-related URL parameter names to the following whitelist.  Separate
            each parameter name with a pipe (|) character. */
            ixf_config.SetProperty(IXFConfigKeys.WHITELIST_PARAMETER_LIST, "ixf");

            IXFSDKClient _ixf_sdk_client = new IXFSDKClient(ixf_config, HttpContext.Current);

            return "<!-- BrightEdge Integration - Start - " + URL + " -->\r\n" +
                   _ixf_sdk_client.GetHeadOpen() + "\r\n" +
                   _ixf_sdk_client.GetBodyOpen() + "\r\n" +
                   _ixf_sdk_client.Close() + "\r\n" +
                   "<!-- BrightEdge Integration - End -->\r\n";
        }
    }
}
```

TEMPLATE FUNCTION CALLS

```
<head>
    @{
        //BE IXF: place GetHeadOpen at the top of the <head>
        if (ixf_sdk_client != null)
        {
            @Html.Raw(ixf_sdk_client.GetHeadOpen());
        }
    }
    <title></title>
</head>


    //BE IXF: The following <div> block needs to be placed in the location where the link block will be displayed
    //BE IXF: For your website, the location is above/below...
        <div class="be-ix-link-block">
           @{ if (ixf_sdk_client != null)
               {
                    @Html.Raw(ixf_sdk_client.GetBodyString("body_1"));
                    @Html.Raw(ixf_sdk_client.Close());
               }
            }
        </div>
        //BE IXF: End
        <footer>

        </footer>
```

JAVA SAMPLE CODE

INSTANTIATION CODE

```java
import com.adobe.cq.sightly.WCMUsePojo;
import com.brightedge.ixf.IXFSDKClient;
import com.brightedge.ixf.util.*;
public class BrightEdgeUtil extends WCMUsePojo {
    private IXFConfiguration ixfConfig;
    private String headOpen;
    private String bodyString;
    private String pagePath;
    private String hostname;
    private String country;
    @Override
    public void activate() throws Exception {
        String currenturl = this.getRequest().getRequestURL().toString();
        hostname="www.xxxx.com";
        this.pagePath=get("pagePath",String.class);
        this.country=get("country",String.class);

        IXFConfiguration ixfConfig = new IXFSDKConfiguration();
        ixfConfig.setProperty(IXFConfiguration.CAPSULE_MODE, IXFConfiguration.REMOTE_PROD_CAPSULE_MODE);
        ixfConfig.setProperty(IXFConfiguration.ACCOUNT_ID, "[ENTER_ACCOUNT_ID]");
        ixfConfig.SetProperty(IXFConfiguration.API_ENDPOINT, "https://ixfd1-api.bc0a.com/");
    //BE IXF: Update CHARSET if needed
        ixfConfig.setProperty(IXFConfiguration.CHARSET, "UTF-8");
    //BE IXF: By default, all URL parameters are ignored. If you have URL parameters that add value to page content.
    // Add them to this config value, separated by the pipe character (|).
        ixfConfig.setProperty(IXFConfiguration.WHITELIST_PARAMETER_LIST, "ixf");
    //Log Level options:LOG_LEVEL_ALL,LOG_LEVEL_INFO,LOG_LEVEL_WARN,LOG_LEVEL_ERROR,LOG_LEVEL_DEBUG
        ixfConfig.setProperty(IXFConfiguration.LOG_LEVEL, IXFConfiguration.LOG_LEVEL_ERROR);

ixfConfig.setProperty(IXFConfiguration.PAGE_ALIAS_URL,"https://"+hostname+"/"+this.pagePath);
        //Optional parameters
        //ixfConfig.setProperty(IXFConfiguration.PROXY_HOST, hostname);
        //ixfConfig.setProperty(IXFConfiguration.PROXY_PORT, "1234");
        //ixfConfig.setProperty(IXFConfiguration.PROXY_PROTOCOL, "https");

        IXFSDKParameters parameters = new IXFSDKParameters(this.getRequest());
        IXFSDKClient client = new IXFSDKClient(ixfConfig, this.getResponse(), parameters);

        headOpen=client.getHeadOpen();
        bodyString=client.getBodyString("body_1");
        closeString=client.close();
    }
    public IXFConfiguration getIxfConfig() {
        return ixfConfig;
    }
    public String getHeadOpen(){
        return headOpen;
    }
    public String getBodyString(){
       return bodyString;
    }
    public String getCloseString(){
       return closeString;
    }
}
```

TEMPLATE FUNCTION CALLS

```
    <head>
    <!-- BE IXF: BE IXF: Place getHeadOpen just inside of the head tag -->
        <sly data-sly-use.brightEdgeUtil="${'com.domain.site.commons.use.BrightEdgeUtil' @pagePath=currentPage.Path}"/>
        ${brightEdgeUtil.headOpen @context='unsafe'}
    </head>
    <body>



    <footer>
    <!-- BE IXF: The following <div> block needs to be placed in the location where the link block will be displayed
    BE IXF: For your website, the location is above/below ...-->
        <div class="be-ix-link-block">
        ${brightEdgeUtil.bodyString @context='unsafe'}
        ${brightEdgeUtil.closeString @context='unsafe'}
        </div>
    </footer>
```

SECURITY CONSIDERATIONS

The security of our customer data is our top priority. When BrightEdge was founded in 2007, we built our SaaS (Software as a Service) technology with security as a founding principle, choosing to undertake the most stringent global security standard, ISO 27001. Our independent auditor for ISO 27001 certification is the British Standards Institution, the same reputable security auditor that certifies many prestigious large technology companies around the world.

We use multiple IaaS (Infrastructure-as-a-Service) providers with the highest reputation in Cloud Infrastructure Service to host our software. AWS (Amazon Web Services) has been historically our primary infrastructure service provider. We recently started using GCP (Google Cloud Platform) as our secondary infrastructure service provider.

Both AWS and GCP were recognized as the “Leaders” in 2018 Gartner’s Magic Quadrant for Cloud Infrastructure as Service. The combination of the best Cloud infrastructures enhances our SaaS Platform to be the most reliable and scalable in our industry.

The BrightEdge Platform is a Privacy Shield certified software and is also GDPR (General Data Protection Regulation) compliant. We have Data Protection Agreements with all our Infrastructure Service providers, and they are also ISO 27001 certified and GDPR compliant as well. AWS and GCP data centers are audited regularly by an independent auditor for their SOC1 and SOC2 compliance, and we periodically review their SOC1/SOC2 audit reports to ensure their commitment to security compliance (SOC1/SOC2 audit reports from AWS and GCP are available upon request).

We perform security scans on every released version of our AutoPilot SDKs. We use automated code analysis solutions from Veracode to assist us with:

- Proactively detecting security bugs in our source codes during development
- Automating static code analysis with CD/CI integration
- Identifying the security vulnerabilities in applications through dynamic code analysis
- Responding to customer’s reports on software vulnerabilities in our source codes

IP WHITELISTING REQUIREMENTS

For the AutoPilot solution to assess relevancy and serve data to your pages, our website crawlers need to be able to analyze your publicly available website URLs. Oftentimes, web filtering solutions (or content delivery networks) are deployed as a standard best practice by our customers to control for unwanted bot activity and may block access from our crawlers unwittingly.

To ensure the solution integrity, we highly recommend whitelisting the following IP ranges in your relevant infrastructure to ensure full coverage of your website content:
38.146.39.162/31
38.146.39.164/30
38.146.39.168/29
38.146.39.176/32
69.166.9.2/31
69.166.9.4/30
69.166.9.8/29
69.166.9.16/32
155.254.16.85/32
155.254.16.86/31
155.254.16.88/29
155.254.16.96/30
192.96.218.25/32
192.96.218.26/31
192.96.218.28/31
209.236.227.130/31
209.236.227.132/30
209.236.227.136/29
209.236.227.144/32

FREQUENTLY ASKED QUESTIONS

- **What data sources are used by AutoPilot?**
  - BrightEdge Data Cube
  - Google Search Console (when integrated into the BrightEdge platform)
  - Web Analytics (variety of vendors - when integrated into the BrightEdge platform)
  - Content IQ crawler (Site Audit) and a dedicated AutoPilot crawler.
  - *(Optionally)* a JavaScript tracking pixel, known as the Relevancy Pixel
- **What criteria is used to consider the links relevant – or how are the links chosen?**
  - The criteria for relevance are derived from two principal sources: 1) content analysis of the page hosting the links and the eligible targets to link to according to the configuration 2) our SEO data, principally the BrightEdge Data Cube and Google Search Console provides us indicators of what URLs have the most opportune value to link to from an SEO perspective – primarily this boils down to pages that have striking distance ranking opportunities. The processing order is linked, so that we first establish relevance on point 1 to make sensible link recommendations, and are then internally prioritized against the algorithm in step 2 to arrive at the final link assignments (and subject to the # of links we are displaying on the page).
  - If there are not sufficient relevance established candidates to make up the required # of links to display, fallbacks are configured to meet the display need (“default links”, configurable in the platform).
- **How do you monitor the performance of the AutoPilot solution?**
  - Post deployment in production an automated in-platform dashboard provides a monthly snapshot of solution performance.
  - A regularly scheduled deep dive of solution performance is delivered by the BrightEdge team, the initial report is typically delivered 45 days from the go-live date to allow for the availability of a new Data Cube dataset to be published.
  - Solution reporting tracks the evolution of key ranking metrics for URLs targeted by Self Connection Pages by comparing trends between the Data Cube data before go-live ("baseline") to each subsequent month of fresh Data Cube data since go live.
- **How is the text (anchor text) of the link determined?**
  - AutoPilot leverages either the H1 tag or Title tag of your pages to establish the anchor text to reinforce your on-page keyword optimization efforts and decisions. The configuration provides for sanitization rules that can be applied to these inputs to keep them clean and focused.
- **Do you support images or other artifacts such as reviews, pricing, etc. in the link block?**
  - The server-side SDK integration offers the possibility to incorporate other key attributes that may enhance the creative or user experience by providing a

structured JSON response that allows your developers to enrich or modify the presentation at will using other data sources.

- **Can we have different creatives across our site?**
  - Different creatives can be supported when the styling is customer managed (see section "Styling Management and Delivery")
- **How often do the links change?**
  - There are two core cadences within the product that could lead to links changing on a given page:
    1. Monthly complete recalculations based on new incoming Data Cube data
    2. Weekly crawl adjustments to eliminate any links that have errors (4xx, 5xx http status codes) or redirect through a 30x status code.
  - Links can also be modified and deployed ad-hoc from the product user interface within the BrightEdge platform.
- **What manual overrides are available to end users?**
  - While most customers enjoy the automated, hands-off nature of the solution – we know that some instances or events call for manual intervention. Self-Connecting Pages offers a feature rich User Interface within the BrightEdge platform that enables credentialled users to make fine grained modifications to the product configuration, including:
    1. Simple:
       - Preventing a specific page from being linked to (Archiving)
       - Replacing specific links on a given page with an alternative link.
       - Overriding the anchor text or descriptive content of a specific link.
       - Disabling the link block on a per-page basis.
       - Re-crawling on-page elements for a given page – for example after performing specific on-page optimizations.
       - Managing the default links used by the product.
       - Creating additional profile segments.
       - Deploying any changes made from the above from the preview area to be live on the website.
    2. Advanced:
       - Forcing a given link to always be displayed in the link blocks – at a profile or global scope.
       - Excluding sections, using URL pattern matching, of the domain from being linked to (targets) or from hosting the link block (sources).
       - Increase or decrease the number of links shown in the link block at a per profile or global scope.
       - Modifying anchor text selection, truncation, and sanitization rules at a per profile or global scope.

- **Are the links going to be internal and pointing to the same site or some might be external?**
    - The Self-Connecting Pages solution is designed to enable linking to any pages contained within your apex/root domain. This includes linking to content on other subdomains if they belong to that same apex/root domain name.
- **What happens when there are no relevant links identified?**
    - If an insufficient number of links are algorithmically identified to produce a complete link block (due to very niche topics or profile exclusion rules for example), the product will source at random from the list of configured default links. This process is repeated until the link block reaches its configured set number of links to display.
- **Do you support X language or country?**
    - AutoPilot Self-Connecting Pages supports English language markets that have a dedicated country/language Data Cube in the BrightEdge platform.
    - For best results, each supported country/language market should have a dedicated BrightEdge platform account provisioned – and a distinct integration account ID configured.
    - Please contact your BrightEdge team to discuss the best approach and currently supported markets.
- **What level of SLA is available?**
    - AutoPilot Self-Connecting Pages adheres to a 99.9% uptime SLA.
- **Do/Can we supply any product data or other data feeds to your solution?**
    - Currently, AutoPilot does not consume custom data feeds into the solution. However, the relevancy algorithms incorporate your BrightEdge Google Search Console and Analytics integrations as enrichment signals to guide relevancy.
- **Do you support X CMS?**
    - We support all flavors of CMS through generic design. We have Java, .NET, and PHP SDKs that provide a convenient integration approach for the majority of off-the-shelf and even bespoke CMS builds. This includes but is not limited to: WordPress, Drupal, Magento, Joomla, Kentico, Sitecore, Sitefinity, Umbraco, EPiServer, Adobe AEM, Hybris, Salesforce Commerce Cloud, and more.
- **Can you implement into SPAs?**
    - In principle, SPAs can be integrated in as long as SEO friendly URLs are being created for your various SPA states. However, without further engineering, the user experience will not reflect unique link blocks when transitioning between states as the SDK is not getting fired in tandem. Please consult with the BrightEdge team to assess the feasibility and an appropriate integration path.
- **Do you support native modules or plugins?**
    - We currently do not provide CMS native modules/plugins.
- **Do you support m.dot mobile sites?**
    - There is limited support for m dot websites as the approach is mostly being discouraged by the major search engines.
- **Why do so many pages show default links?**

  - For AutoPilot Self-Connecting Pages to produce non-default link blocks, the underlying URL needs to have been crawled and processed by our product. The solution is principally geared towards qualifying URLs for which our SEO data holds clear signals against which we can prioritize. This means that to have non-default link blocks, the URLs need to be ranking and present in our Data Cube data at a minimum.
  - Temporal content URLs such as high churn product URLs and/or real-time content are unlikely to develop maturation to receive non-default link blocks if they dissipate from existence in under a month.
  - Alternatively, the following factors may be at play:
    1. Our crawlers are blocked by your web filtering infrastructure (usually returning a 403 status code) and therefore not discoverable.
    2. The URL is canonicalized elsewhere or set to noindex.
    3. A sufficient crawl was not achieved for those URLs or discovered through alternate sources including GSC and BrightEdge ContentIQ.
- **How do you handle out of stock pages?**
  - If sold out inventory is 301/2 redirected or 404'd, our crawlers will expunge them as linking candidates through our weekly crawls.
- **How do you handle redirected or dead pages?**
  - Weekly crawls are automated to catch any links that have errors (4xx, 5xx http status codes) or redirect through a 30x status code. These pages are then no longer linked to by the product.
  - Similarly, pages that are redirected or not found will get pruned from our production data during those weekly crawls and will stop presenting data to your integration.
- **What is the Relevancy Pixel and is it required?**
  - The Relevancy Pixel assists the machine learning algorithms to detect topical adjacencies and concepts that might not typically be derived from pure learning and natural language processing algorithms. The pixel tracks page sequences from your website visitors to identify frequent patterns that in turn provide a signal of additional relevance into our learning algorithms. No PII is collected, however, some IP logs are temporarily stored to assist in debugging. The Relevancy Pixel has a very subtle overall impact and is considered entirely optional and readily disabled by the BrightEdge team before or after integration.
- **What is my AutoPilot account ID?**
  - Your AutoPilot account ID typically takes the form such as f00000001234 and is a required variable to enable the AutoPilot integration.
  - The AutoPilot account ID is most often tied to a given BrightEdge S3 account.
  - Generally, there is only one AutoPilot Account ID in use per customer, but some scenarios may call for multiple such IDs.
  - Contact your BrightEdge team to receive your Account ID, or for further questions.

APPENDIX: SAMPLE CREATIVES

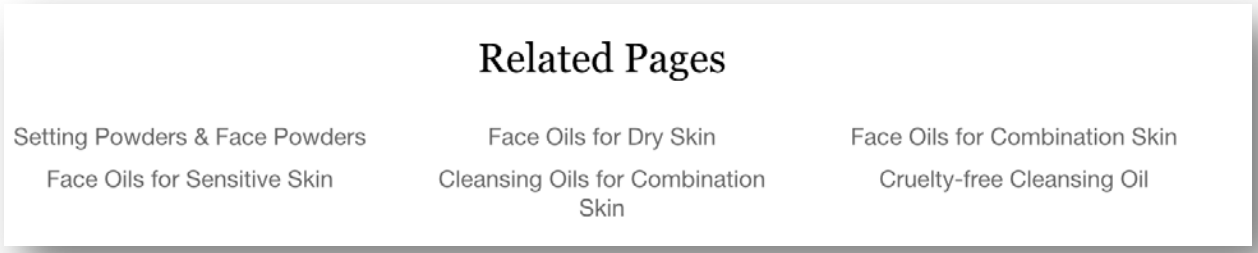

Figure 11: Three column, below content but above footer

Figure 12: Pill button list

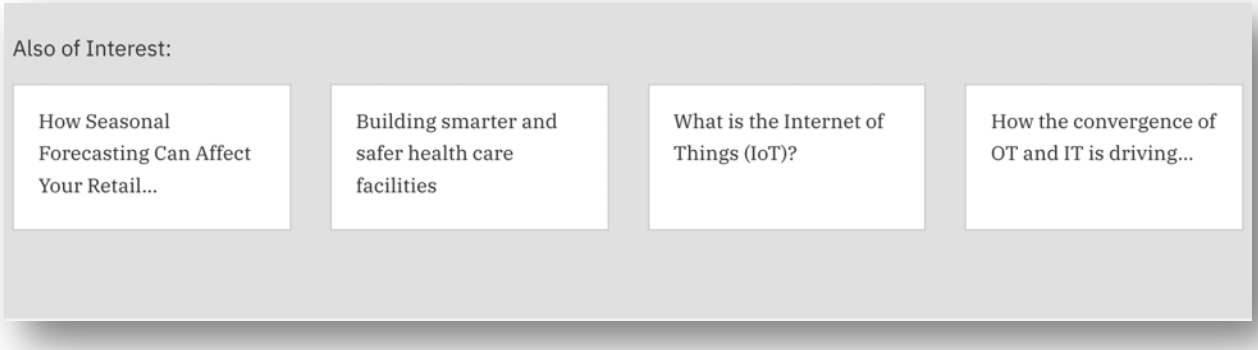

Figure 13: Simple Tile Layout

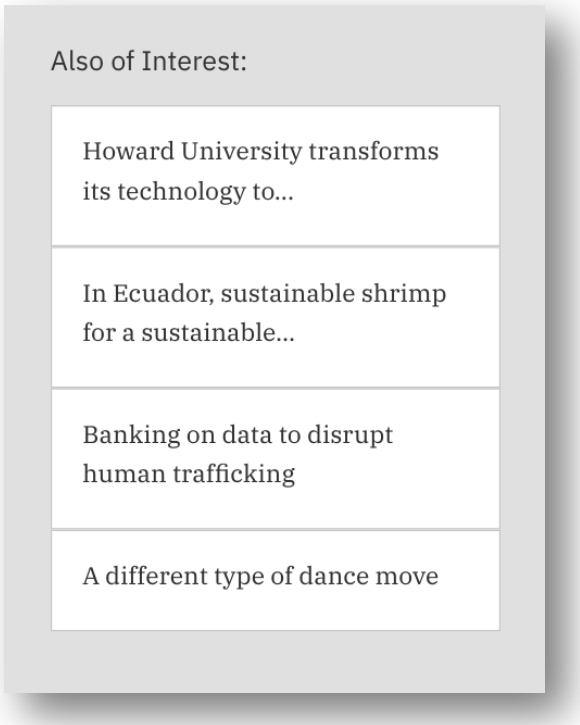

Figure 14: Simple Tile layout collapsed for Mobile device viewports

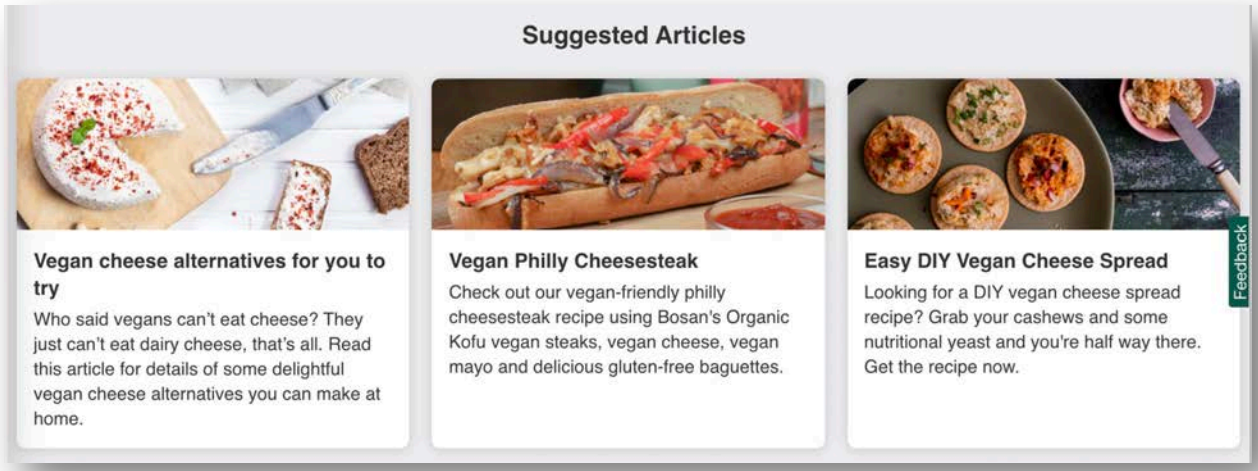

Figure 15: Card design including an "image cap" and supplemental card content