**TABLE OF CONTENTS**

**CHAPTER**                                                                                                                     **PAGE**

**What is Amplience ?..............................................................................................................**

**Use of Amplience?..................................................................................................................**

**Pre-Requisites.............................................................................................................................**

**Amplience with SFCC Integration…….........................................................................**

* **Salesforce Commerce Cloud Integration…………………………………………………**

**1.content slots Integration…………………………………………………………….**

**2.Content Assets Integration…………………………………………………………..**

* **Setting up a SFCC integration……………………………………………………………...**

**Getting to know Amplience………………………………………………………………….**

* **Dynamic Content ……………………………………………………………………………...**

**1. Dashboard…………………………………………………………………………….**

**2. Assert Tab……………………………………………………………………………..**

**3. Content Library………………………………………………………………………**

**4. Scheduling Tab………………………………………………………………………**

**5. Development Tab…………………………………………………………………….**

**6. Settings………………………………………………………………………………..**

* **Content HUB…………………………………………………………………………………….**

**1.Content Hub Tools……………………………………………………………………**

**2. Impersonation………………………………………………………………………..**

**Getting started..........................................................................................................................**

* **Model Content ...............................................................................................................**
* **Create Content...............................................................................................................**
* **Publish Content.............................................................................................................**
* **Consume Content.........................................................................................................**

**Concepts………………………………………………………………………………………**

* **Schema ..........................................................................................................................**
* **Partial Schemas.............................................................................................................**
* **Content Type..................................................................................................................**
* **Slots................................................................................................................................**
* **Extensions.....................................................................................................................**
* **Integrations ...................................................................................................................**
* **Webhook........................................................................................................................**
* **Scheduling……………………………………………………………………………………...**
* **Visualization..................................................................................................................**
* **Preview...........................................................................................................................**
* **Content modelling……………………………………………………………………………..**
* **Publishing and scheduling………………………………………………………………….**
* **Versioning………………………………………………………………………………………**
* **Caching………………………………………………………………………………………….**

**Content Rendering Service………………………………………………………………….**

* **Using the content rendering service……………………………………………………….**
* **Handlebars………………………………………………………………………………………**
* **Handlebars templates…………………………………………………………………………**
* **Partials…………………………………………………………………………………………...**

**Amplience Limitations……………………………………………………………………….**

**Glossary....................................................................................................................................**

* **Hubs...............................................................................................................................**
* **Status.............................................................................................................................**
* **Archive...........................................................................................................................**

**1.What is Amplience ?**

Amplience is an API-first, headless content management platform for enterprise retail. Content and experience solutions for headless commerce enable us to do more, better, faster.

ReferenceLink: [Headless CMS Platform](https://amplience.com/product/)

**2.Use of Amplience?**

 Used to create, modify, preview, publish, schedule the content on the page.

**3.Prerequisites**

* Person should have access to the Amplience portal - [Dynamic Content](https://content.amplience.net/#!/ariat/authoring/content-items)
* Person should have access to the SFCC business manager - [Commerce Cloud Account Manager](https://account.demandware.com/dwsso/XUI/?realm=/&goto=https://account.demandware.com:443/dwsso/oauth2/authorize?response_type%3Dcode%26client_id%3D5c4fe03a-9cbb-3e25-8350-f12008e6ba55%26scope%3DtenantFilter%2520openid%2520profile%2520roles%2520externalId%2520email%26state%3DpWqwcXYHaihTV2fymAYwLE0FGz7yv4a6eiwNqfc5kCU%253D%26redirect_uri%3Dhttps://aaml-011.sandbox.us01.dx.commercecloud.salesforce.com:443/on/demandware.servlet/dw/oidc/openid_connect_login%26nonce%3DsD66SZ9JR-R8wBzApi4y7tUg5oD5pYYThm3Gp3udfI4%26prompt%3D#login/)
* Person should do SFCC Integration configuration on Amplience portal which requires below details.

1.Amplience Dynamic Content Auth - Client-Id and Client-Secret.

2.Salesforce Commerce Cloud Auth - Client-Id and Client-Secret.

3.SFCC Api path and version.

4.Content rendering service - VSL URL and service template label.

* Person should do OCAPI Configuration in SFCC business manager.

**4.Amplience with SFCC Integration**

**Amplience** is used for authoring, scheduling and publishing the content to SFCC.

          In **SFCC** we can see the published content on the page.

* **Salesforce commerce cloud integration:**There are two ways of integrating Dynamic Content with SFCC:

1.Mapping slots in Dynamic Content to **content slots** in SFCC.

2.Generating **content assets** in SFCC from content items in Dynamic Content.

**1.Contet Slots:**The Dynamic Content Salesforce Commerce Cloud (SFCC) integration makes it easier for developers to integrate Dynamic Content projects with SFCC. The integration is designed to reduce the amount of setup that developers are required to do and to make the integration process simpler for Dynamic Content users and those using and configuring SFCC.

<https://amplience.com/developers/docs/integrations/commerce/salesforce/slots/>

**2. Content Assets:** The Dynamic Content Salesforce Commerce Cloud (SFCC) integration can be set up to create SFCC content assets from content items in Dynamic Content. Once the integration is configured, all the user needs to do is create and publish content items of the specified content types from the Dynamic Content production view. The integration will create or update the corresponding content assets in SFCC. You can also specify mappings between properties in Dynamic Content content items and fields in SFCC content assets.

[SFCC Content Asset Integration | Amplience Developer Portal](https://amplience.com/developers/docs/integrations/commerce/salesforce/assets/)

* **Setting up a SFCC integration:** On this page we'll show you how to add a SFCC integration from the Dynamic Content app. Some of the set up is the same for both content slot and content asset integrations, but to configure a content asset integration requires some additional steps, as explained on the content assets page.

[Setting up a SFCC integration | Amplience Developer Portal](https://amplience.com/developers/docs/integrations/commerce/salesforce/setup/)

**5. Getting to know Amplience**

**Dynamic Content:**  The content production features of Dynamic Content include everything you need to create, schedule and publish content. We provide a Content Library that makes it easy to view and organize your content, and preview and editing features to help you get the content ready to be scheduled for publication.

         Dynamic Content URL: [Dynamic Content](https://content.amplience.net/#!/ariat/authoring/content-items)

**The Dashboard:**The Dashboard provides you with a configurable view of your content,   allowing you to quickly focus on the content items and editions that you need to work with. Your dashboard opens when you log in to the Dynamic Content app and you can also view it at any time by clicking "Dashboard" in the masthead at the top of the window.

 ReferenceLink:[The Dashboard | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/basics/dynamic-content/dashboard/)

**The Assert Tab:**The Dynamic Content Assets tab streamlines your content production workflow by allowing you to quickly perform some key asset management tasks without the need to switch to Content hub. Using the Assets tab you can easily access the Asset Library to perform tasks such as uploading and publishing assets.

ReferenceLink:<https://amplience.com/developers/docs/user-guides/basics/dynamic-content/assets-tab/>

**Content Library:**Content is viewed, edited, created and organized in the Content Library that is opened when you choose the "Content" tab in the Dynamic Content app.

   ReferenceLink:[Content Library | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/basics/dynamic-content/content-library/)

**The scheduling Tab:**heduling tab lets you work with your events and editions, in a calendar view, as a list or in a timeline. The calendar view provides a month by month overview of your events, while the list will show the events in a list. The timeline displays events and editions in a time period you choose, from one week to one year.

ReferenceLink:[The Scheduling tab | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/basics/dynamic-content/scheduling-tab/)

**Development Tab:**The Dynamic Content Development tab is a launch page to developer documentation and resources. It is developers who customize Amplience to how your organization operates, providing the structures used by content producers and schedulers.

ReferenceLink:<https://amplience.com/developers/docs/user-guides/basics/dynamic-content/development-tab/>

**Settings:**This section describes the settings you can use when working within Dynamic Content.

ReferenceLink:[Settings | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/basics/dynamic-content/settings/)

**Content Hub:** Content Hub is the foundation of the Amplience platform and is where all assets, including images and video are stored. It underpins the services of Dynamic Media, which includes powerful features for manipulating images and video, and acts as the media library for assets used in Dynamic Content.

         Content HUB URL: [Amplience: Log On](https://ondemand.amplience.com/dam?alerts=true)

**Content Hub Tools:**There are a number of asset management tools that you can access from the Content Hub Tools section. On this page, we'll provide an overview of each of these tools and how they are used within the platform.

     ReferenceLink:[Content Hub tools | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/basics/content-hub/tools/)

**Impersonation:** Impersonating accounts and users is a feature within Content Hub that enables you to perform activities such as, configuration, troubleshooting and assisting users with tasks.                                                                                   ReferenceLink:[Impersonating accounts and users | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/basics/content-hub/impersonation/)

**6. Getting started**

**Model Content :** The first step when creating your website or app is content modeling- where the wireframes and requirements that make up your user experience are converted into a content model. This model is then used to create a content type.

**Example:** Take the example of Banner, here we break down the banner into the elements -   Heading, Text, Image, CTA.

Reference Link: [Step 1: Model | Amplience Developer Portal](https://amplience.com/developers/docs/start/model/)

**Create Content:** In Step 1 you created the content type. In this step you'll use that content          type to create some content.

**Example:** Now the ContentType is Banner - we have to provide the inputs for this elements - Heading text, image, CTA.

 Reference Link: [Step 2: Create | Amplience Developer Portal](https://amplience.com/developers/docs/start/create/)

**Publish Content:** Publishing is a process that works with content items and takes the content objects from the Dynamic Content back-office and places them in our delivery systems, ready to be consumed by applications via our delivery APIs.

**Example:** Now the banner is published, you can retrieve it from the CMS. Now we can quickly view the JSON content for the banner directly from the content form.

Reference Link: [Step 3: Publish | Amplience Developer Portal](https://amplience.com/developers/docs/start/publish/)

**Consume Content:**Now that you've created and published the content, you can use the content delivery SDK to retrieve it from the CMS and then render it in the browser.

Reference Link: [Step 4: Consume | Amplience Developer Portal](https://amplience.com/developers/docs/start/consume/)

**7. Concepts:**

**Schemas:** A schema defines the data model for a content type.We use an open standard called JSON Schema to provide you with a powerful and flexible way of modeling your data. JSON schema describes the format of JSON structures using a set of keywords that you combine to create complex structures.

ReferenceLink: [The schema editor | Amplience Developer Portal](https://amplience.com/developers/docs/dev-tools/guides-tutorials/schema-editor/)

**Partial Schemas:** Partials are a type of mix in schema that contain just a definitions section. These definitions can be referenced in multiple schemas, particularly useful if you have some complex properties that you want to store and refer to in one place. You can also share definitions within the same schema.

ReferenceLink:<https://amplience.com/developers/docs/schema-reference/mixins/#partials>

**Content Type:**A content type is a blueprint that describes a specific data structure users can create and manage inside the CMS, such as a page, banner, blog post etc.

ReferenceLink: [Working with content types | Amplience Developer Portal](https://amplience.com/developers/docs/dev-tools/guides-tutorials/content-types/)

**Slots:** Content can be scheduled, published and consumed using slots. Slots are containers for content and will model the way that content is organized on your website, app, or within an e-commerce system.

ReferenceLink:[Slots | Amplience Developer Portal](https://amplience.com/developers/docs/dev-tools/guides-tutorials/slots/)

**Extensions:** Write and register your own code to extend the functionality of the Amplience platform.

ReferenceLink: [Extensions | Amplience Developer Portal](https://amplience.com/developers/docs/integrations/extensions/)

**Integrations:** Connect Amplience with a range of commerce tools and platforms using off the shelf integrations.

Reference Link: [Integrations and extensions - Amplience Developer Portal](https://amplience.com/developers/docs/integrations/)

**Webhook:** A webhook is a way for Dynamic Content to automatically send messages or data to a third party system.

Reference Link: [Webhooks | Amplience Developer Portal](https://amplience.com/developers/docs/integrations/webhooks/)

**Scheduling:** Dynamic Content lets you plan and schedule your content in an easy to use scheduling calendar, using the concepts of events and editions. While the events and editions represent when you want content to be published, you specify the content itself and where it should be delivered using slots.

ReferenceLink: [Scheduling overview | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/schedule-content/overview/)

**Visualization:** Visualizations are an easy to use and effective way of previewing your content and ensuring it meets your requirements before you save it to an edition to be scheduled for publishing, or at any time during the content production process.

ReferenceLink: [Visualizations | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/produce-content/visualize/)

**Preview:** Content previews allow you to view your website or app exactly as it will appear at a chosen point in time. The preview app can be as complex as you'd like, simulating your entire app or just particular pages on your website, for example. In many cases the preview will be linked to your ecommerce system.

ReferenceLink: [Previewing content | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/schedule-content/preview/)

**Content modeling:** How to model your data structures, define your rules and enforce validation.

 Reference Link: [Content modeling | Amplience Developer Portal](https://amplience.com/developers/docs/concepts/content-modeling/)

**Publishing and scheduling:** Publish the right content at the right time.

  Reference Link:<https://amplience.com/developers/docs/concepts/publishing-scheduling/>

**Versioning:** Working with the versions of content you need.

  Reference Link:<https://amplience.com/developers/docs/concepts/versioning/>

**Caching:** Our dynamic commerce experience platform supports high performance customer journeys through the Dynamic Content and Dynamic Media products. They both use Content Delivery Networks (CDNs). Understanding our caches will help to design successful high performance applications.

  Reference Link: [Caching | Amplience Developer Portal](https://amplience.com/developers/docs/concepts/caching/)

**8.Content Rendering Service**

The content rendering automates the process of combining JSON content with handlebars templates to generate HTML that can then be added to a page on your website. The service provides an API that takes a content or slot ID and a handlebars template as parameters and returns HTML in the response. The HTML can then be combined with CSS styling to display on your website or stored in an e-commerce system.

 Reference Link: [The content rendering service | Amplience Developer Portal](https://amplience.com/developers/docs/apis/content-rendering-service/)

**Using the content rendering service:** Both slots and individual content items are stored as JSON content and retrieved using their unique ID.

The steps to retrieve content from the Dynamic Content using the content rendering service are as follows:

* Send a Get request to the content rendering service with a slot or content ID and a handlebars template as parameters
* If the content is found then HTML is returned in the response

**Handlebars**: Handlebars is an off the shelf templating technology that allows developers to specify the structure of the HTML generated from a piece of JSON content.

**Handlebars templates:** Templates are uploaded to Content Hub as .txt or .html files. You can also create the templates directly in Content Hub using the Text Editor app. A template must be published so that it can be used by the content rendering service.

**Partials**: Partials is a handlebars feature that allows one template to load another. This lets you create a structure where you have one or more top level templates and separate templates for each content type, so you can specify the same template in each call to the content rendering service.

**9.Amplience Limitations**

  Limits for the Dynamic Content APIs and general usage of the Dynamic Content app. Limits for Content Hub and apps are also included.

  Note that most of these limits are hard coded and cannot be changed. Generally you should not be close to these limits.

     Reference Link:[Limits | Amplience Developer Portal](https://amplience.com/developers/docs/apis/limits/#content-delivery)

**10.Glossary**

* **Hubs:**Hubs are the environments in Amplience dynamic content.

   example: Ariat Dev, Ariat Prod, Ariat Prod AE.

* **Status:** statuses to help organize and prioritize your work. You can apply statuses to content items in a hub. Each status needs a label and a color, and you need to save each status before adding a new one. You can have a maximum of 100 statuses.

Reference URL:[Dynamic Content](https://content.amplience.net/#!/ariat/settings/statuses)

* **Archive:** Archiving a content item does not affect any previous work you have done with this content. If the content item is published, the published version is still consumed as normal. Any versions of the content that have been saved to an edition are still available to add to slots and scheduled for publication. Archived content can no longer be edited, published or copied and does not show up in any search results.

[Organizing content | Amplience Developer Portal](https://amplience.com/developers/docs/user-guides/produce-content/organize/#:~:text=To%20archive%20a%20content%20item,Archive'%20from%20the%20Save%20menu).

* **Filters:** Used to filter the content by Content Type, Assignedto, Status, publish status, LastModify, Locale in dynamic content.