

Ad Pods

Ad pods provide publishers with an opportunity to sequence together multiple video ads using a single VAST request. The functionality simulates a TV commercial break with multiple ad slots. Ad Pods work with VAST 3.0 compliant ad networks ONLY and could be customized or dynamic.

Setting up an Ad Pod

In SpringServe, an Ad Pod is a type of supply tag with features available to manage and customize your Ad Pod. If your account is Ad Pod enabled the "Pod Settings" feature will be shown when adding new supply. You will need to go through all of the steps to create a tag including adding demand and any targeting you might want to have. If you wish to create a supply tag without an ad pod go to the Pod Settings tab and select None. Ad Pods



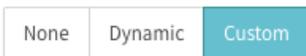
Dynamic Ad Pod

The easiest way to take advantage of the ad pod capabilities available in SpringServe is to set the default ad pod setting to "Dynamic". This will enforce two macros in the export tab that you could use to specify your Maximum duration (**pod_max_dur**) of the ad pod and the ad slot settings (**pod_ad_slots**). The ad slot setting needs to be populated with a range if it is being used. For information on how to populate the macros properly go here: [Macros](#). If you leave the macros blank SpringServe assumes a Maximum Pod duration of 180 seconds - 3 minutes - with a maximum Ad slot allotment of 10, assuming you have enough demand to fill 10 slots and 3 minutes. If there is not enough demand to fill the entire 3 minutes your pod will complete and the content will resume.



Custom Ad Pod

Should you choose to customize your ad pod you could do so by selecting Custom in the Pod Settings tab.



This will allow you to add a custom maximum duration for your ad pod and the ad slots within it. The Pod Max Duration is the total maximum length of your ad break (the pod) in seconds. The total amount of ads, in seconds, will not exceed this time limit. Additionally, you could customize the number of Ad's (Ad Slots), no more than 10, within the ad break. This part is optional but suggested if you want to customize the experience with specific ad lengths and

Ad Slot duration:



ads. If you select

then SpringServe will look for ads that pass all targeting and fit within the **Pod Max Duration** that is

Ad Slot duration:



set. If you select

you will have the option to select a minimum or a maximum for each ad in the pod. The durations will be taken into account and anything outside of the requirements will not serve. It is easy to add or remove the entire ad pod or ad slots individually by clicking on the red buttons as needed.

Pod Max Duration:

120 Sec

Ad Slot duration:

	Any	Custom	Min:	Max:	
1	Any	Custom			⊖
2	Any	Custom			⊖
3	Any	Custom			⊖

+ Ad Slot

⊖

Ad Pod Demand Set up

An Ad Pod request calls 12 ads all at once so it is in your best interest to have at least 12 demand tags available in the waterfall. We suggest using tiers and setting a priority value on each tag in order of importance, with directly sourced tags at the top. Note that if your demand stack includes DirectConnect tags, all 12 requests may go to underlying demand and other tags in the waterfall may not fill.

Using your Ad Pod

When you have completed and saved the Ad Pod settings along with the rest of the tabs that are required to create tags you will be able to export the tag that houses your Ad Pod. Use this Tag anywhere you would normally use supply tags to create a unique experience for your users.

For more information about Ad Pods and to enable your account with the feature please contact your account manager.

SSAI Workflow

Most SSAI vendors work in a similar manner. They are all VAST compliant and the general workflow goes like this:

1. Player calls SSAI vendor for content
 - a. The SSAI vendor is configured to know where the HLS or DASH content streams reside (usually a CDN or S3)
 - b. The HLS or DASH content has cue points to tell the SSAI vendor when to insert ads
2. When a cue point is about to be reached, the SSAI vendor calls out to SpringServe for ads.
3. When SpringServe gets the call we:
 - a. Find all demand partners that pass targeting
 - b. Call them all server side and follow any wrapper VASTs until we get to an asset
 - c. Fill the pod as much as possible (adhering to the tiers created in SpringServe)
 - d. Stitch in our impression and other beacons for tracking
 - e. Return the resulting VAST to the SSAI vendor
4. The SSAI vendor then takes the assets from the VAST, transcodes them and stitches them into the HLS/DASH stream

Example: MediaTailor Integration:

To configure an SSAI vendor is pretty straight forward. At a minimum it requires two fields: content source and ad decision server.

Content Source:

This is the CDN or S3 bucket where the SSAI vendor can access the HLS or DASH streams. Here is what that looks like in MediaTailor:

Video content source [Info](#)

The origin server that's providing content to AWS Elemental MediaTailor.

<https://s3.amazonaws.com/springserve.test/springserve/mediataylorstest>

Ad Decision Server

This is the ad server's VAST endpoint (in this case SpringServe) that you want the SSAI vendor to call in order to know which ads to play.

Ad decision server [Info](#)

The URL for the ad decision server (ADS).

`https://vid.springserve.com/vast/378094?w=1920&h=1080&cb=[avail.random]&ip=[session.client_ip]&ua=[session.user_agent]`

SpringServe makes this even easier by showing you which macros you can populate in our tag export.