

How to set up a Floors A/B test

This example is to set a single floor for all adUnits on a given page. For adUnit-specific floor prices (not especially necessary for a first step)

Set up the infrastructure required

- *On the SSP side:*

Set up the Prebid floors module, but instead of a fixed value, we will set a variable as the value, let's call it testFloor.

- *On the GAM side:*

First, create a key-value - let's name it "floor" - as predefined (and custom dimension if possible, it will allow for better analysis), and declare as the values each of the floors that are to be applied later on, for example, 0.05, 0.1, 0.2, 0.3, 0.4, etc.

Then, create the same number of pricing rules, and for each, target a different value of the key "floor", and set the matching floor price.

✕ New unified pricing rule

Name p.05

Targeting

Inventory	✓ All inventory	▼
Inventory URL	✓ All inventory urls	▼
Mobile application	✓ All mobile applications	▼
Geography	✓ All geographies	▼
Device category	✓ All device categories	▼
Operating system	✓ All operating systems	▼
Custom targeting		^
floor		CLEAR
is any of	0.05	AND
Search or paste a comma-sep...		
Inventory type	✓ All inventory types	▼
Video position	✓ All video positions	▼
Inventory format	✓ All inventory formats	▼

Pricing

- Set pricing for everything ?
- Set pricing for specific items

Pricing options for everything

Pricing

- Set floor prices
The lowest winning bid must be at least as high as the floor price
- Set target CPMs ?
Floor prices are dynamic but average CPM over time targets the chosen CPM

Branded € 0.05

Now that everything is in place, we can start dynamically setting up the floor price.

Split the traffic

Like in any A/B test, you need to split your traffic equally to create several test populations and define the variable testFloor with the relevant value.

For example if we want to test 4 different floors against one another:

```
let cohorts = [0.1, 0.2, 0.3, 0.4];
```

```
var testFloor = cohorts[Math.floor(Math.random() * cohorts.length)];
```

Apply the variable. Now we want to apply the relevant floor value to each population.

- *On the SSP side:*

As said at the beginning, in the Floors module, the value needs to be set as the variable "testFloor" that we defined above.

```
pbjs.setConfig({
  floors: {
    enforcement: {
      bidAdjustment: true
    },
    data: {
      currency: 'INR',
      schema: {
        fields: ['gptSlot']
      },
      values: {
        '*': testFloor
      }
    }
  }
})
```

****Make sure the provided value is a number (float) and not a string, otherwise it might not work.

- *On the GAM side:*

For GAM, we just need to target the Key-Value in order to trigger the relevant UPR:

```
googletag.pubads().setTargeting('floor', `${testFloor}`)
```

****Make sure the provided value is a string, otherwise, it might not work.

Collect the data

Data collection is the same as for any A/B test.

To follow what is happening in Prebid, you can use [Pubstack's tags field](#).

You can send the data as follows:

```
pubstack: { tags: [ "floor": `${testFloor}` ] }
```

To follow what is happening in GAM, you can use the key-value "floor" that we created.

The analysis will let you know which floor brings the most revenue. Usually, you'll want to iterate regularly.

Setting AdUnit Specific Floors

To configure adUnit specific floors, we will have to set a different floor variable for each adUnit.

Then, in the Floors Module, we simply assign the proper variable to each adUnit.

In GAM, we will need to use placement-level Key-Value targeting instead of page-level in order to activate a given UPR for a single ad slot. This gives us:

- *On the SSP side:*

The field used can be either "gptSlot" like in the example, or "adUnitCode".

```
pbjs.setConfig({
  floors: {
    enforcement: {
      bidAdjustment: true
    }
  },
});
```

```
data: {
  currency: 'INR',
  schema: {
    fields: [ 'gptSlot' ]
  },
  values: {
    'div-gpt-ad-101': testFloor1,
    'div-gpt-ad-102': testFloor2,
  }
}
})
```

- *On the GAM side:*

```
adSlot101.setTargeting('floor', `${testFloor1}`);
adSlot102.setTargeting('floor', `${testFloor2}`)
```