

# Creating a Scaler

One of Shareably's strategies as a publisher is to purchase traffic on various social media platforms. Integral to the success of this strategy is designing a scaler that is capable of scaling up the spend on profitable ads while shuttering or scaling down the spend on unprofitable ads.

In this exercise, you'll be using JS and React to:

1. Fetch the performance of mock ads from our API server
2. Evaluate the performance and suggest new budgets for each ad
3. Display the recommended budgets on a UI powered by React

This take-home coding challenge should take approximately 4-6 hours. Feel free to use any external libraries and resources as long as you are not actively involving other people (browsing Stack Overflow is acceptable). You must use Javascript and React for this assignment.

Please submit a Github repository with instructions on how to run your solution.

## [1] Fetching Performance from the API

You'll be fetching data from the API server located at `https://interview-api.sbly.com`. To authenticate use the access token `SHAREABLY_SECRET_TOKEN` in either the `Authorization` header or in the query parameter `accessToken`.

### Ad Performance

Use the GET `/ad-insights` endpoint to fetch ad performance data.

Parameter	Description	Example
date	The date to fetch ad data for. The date should be formatted in YYYY-MM-DD format and valid dates range from 2020-01-01 to 2020-01-05 <code>REQUIRED</code>	2020-01-01
metrics	A comma separated list of metrics you want returned by the server. See below for a complete list <code>REQUIRED</code>	spend,impressions

### Available Metrics

Metric	Description
spend	The total amount spent
revenue	The total dollar amount brought in by the ad
impressions	The total number of times the ad was shown to users
clicks	The total number of times a user click on the ad

## Ad Budget

Use the GET `/ad/AD_ID` endpoint to get the current ad budget for an ad.

Parameter	Description	Example
AD_ID	The ad id to fetch current ad budget for. <b>REQUIRED</b>	081944fe-aa73-6165-3f4c-3ab87a1539fe

## [2] Evaluating the performance of campaigns

Using the data from the API and the rules below, generate the new budget of each ad.

1. For each ad item, compute the profit margin of each ad for each day. Profit margin is defined as  $[(\text{revenue} - \text{spend}) / \text{spend}]$ .
2. Then, for each ad item, compute the weighted average of profit margin across all available days of data. Compute this weighted average by taking an ad's profit margins and weighing on both spend and recency. Use the ad's  $[\text{spend}]$  as the weight for spend and use  $[\text{Math.pow}(0.5, \text{"days from most recent performance"})]$  as the weight for recency. Combine these weights multiplicatively for the finalized weights. See <https://www.investopedia.com/terms/w/weightedaverage.asp> for more details regarding computing a weighted average. Refer to the example below.
3. Then, for each ad item, take  $[1 + \text{weighted average of profit margin}]$  and multiply that with the current budget to get the next budget. If the most recent profit margin was  $> 0$ , do not decrease the budget.

[Step 2 Example]

Spend: \$10

Most recent performance date: 01/05/2020

Date of performance in question: 01/04/2020

Spend is \$10 so the weight for spend is 10.

The performance is dated 01/04 which occurred one day prior to the most recent performance date.

Based on this, the weight by recency is  $\text{Math.pow}(0.5, 1) = 0.5$ .

Combining the two multiplicatively brings you to the final weight of this performance  $10 * 0.5 = 5$ .

## [3] Displaying the Recommended Budget

Design a simple UI showcasing the work done above. Display each ad performance with its id, weighted profit margin, current budget, and proposed budget, along with any other information you think might deem relevant. You must use the React library for this portion.

We look forward to your solution!