**Simple Ad Server**

This doc contains requirements for a coding exercise that involves building a simple HTTP-based ad server.

**Overview**

You will be building a simple web application that allows a user to create ad campaigns. You should demonstrate that your code meets the functional requirements described below via unit and integration tests. There should be instructions for deploying and running the application, ideally expressed via code/configuration and not prose.

Use the language with which you feel most comfortable and share your project via a Git repository in GitHub.

Be prepared to walk through your code, discuss your thought process, and talk through how you might monitor and scale this application. You should also be able to demo a running instance of the host.

**Functional Requirements**

Create Ad Campaign via HTTP POST

A user should be able to create an ad campaign by sending a POST request to the ad server at http://<host>/ad.

The body of the POST request must be a JSON object containing the following data:

{

"partner\_id": "unique\_string\_representing\_partner',

"duration": "int\_representing\_campaign\_duration\_in\_seconds\_from\_now"

"ad\_content": "string\_of\_content\_to\_display\_as\_ad"

}

The server should enforce the following invariant upon receiving a request to create a new campaign.

\* Only one active campaign can exist for a given partner.

If an error is encountered, the ad server must return an appropriate response and indicate the problem to the user.

The response format is left up to the discretion of the implementer.

Storing campaign data in memory or a cookie is totally fine.

**Fetch Ad Campaign for a Partner**

A partner should be able to get their ad data by sending a GET request to the ad server at http://<host>/ad/<partner\_id> Response can be delivered as a JSON object representing the active ad

If the current time is greater than a campaign's creation time + duration, then the server's response should be an error indicating that no active ad campaigns exist for the specified partner.

**Bonus:**

\* Add a URL to return a list of all campaigns as JSON.

<http://localhost:8080/list>

\* Add support for multiple ad campaigns per partner.

http://localhost:8080/multiple/ad