**Programming Test**

**Ad Server web service story:**

Mobile app developers like to monetize their app.

In order to achieve that, their app uses the AdServer which gets an appropriate ad for display to their users.

When getting an ad request, the AdServer itself sends ad requests to multiple Ad Networks, and each responds with the following: an Ad url (if there is an ad that fits), the price that it is willing to pay and 'match' – how much the Ad fits between 0-1.

The Ad-Server then picks a winner as described below.

If there is no suitable Ad from all networks, the Ad-Server returns an houseAd – which is the following url: [Inneractive Housead](https://media.glassdoor.com/sqll/755816/inneractive-squarelogo-1436766311679.png)

**The Task:**

1. Create an Ad server App:

You are given the following Ad Networks:

* 1. YourAdNetwork.
  2. SababaAdNetwork.
  3. InneractiveAdNetwork.

An Ad Network supports Json requests from the ad server.

The Json URL should include the following parameters:

* width (query parameter).
* height (query parameter).
* siteId (query parameter).
* secretKey (in request body).

The server always sends ad requests to all 3 Ad Networks in parallel for efficiency and fairness.

Sample request to Ad Network:

POST localhost:8081(/2/3)/ad?

siteId=<siteId>&width=<width>&height=<height>

Body structure:

{

"secretKey" = <String>

}

Ad network response sample:

{

"adUrl": "[www.YourAdNetwork.com/706783](http://www.YourAdNetwork.com/7063831783)" (or empty string for no Ad),

"matching": 0.8440490893032649,

"price": 0.031563785811634104

}

**Your task**

1.1 Get 1 Ad from an Ad network, and print the Ad-url to the screen.

For the matter of this task you can pick any value for siteId, width, height and secretKey.

1.2 Get 1 Ad by sending requests to the Ad Networks in parallel as described (if you cannot manage to do it in parallel, go sequential and continue..). Print the Ad-url of the network that offered the highest price to the screen.

1. The Ad server should pick a winner by either "price" or "match" according to the value "winningCriteria" param, taken from configuration file.
2. Make your Ad Server a web service. Use any HTTP server implementation (application server), and web service implementation you like (preferably REST implementation). The AdServer should support Json requests (sent by mobile Apps).

Sample request:

POST localhost:8080/adServer/ad?

siteId=<siteId>&width=<width>&height=<height>

Body structure:

{

"publisherKey" = "111111" //any String with size > 0 is valid

}

The Ad-Server validates that width, height and key size > 0, otherwise it throws AdServerException.

The AdServer then uses part 1 to send ad request for the Ad Networks, and respond with the appropriate ad:

{

"adUrl": "www.YourAdNetwork.com/7063831783"

}

1. Every 1 minute the server should print total number of winning ads

**Most importantly, model the pieces in a nice way, keep on clean and clear code...**