Project Agora

Laissez Faire Advertising

Introduction

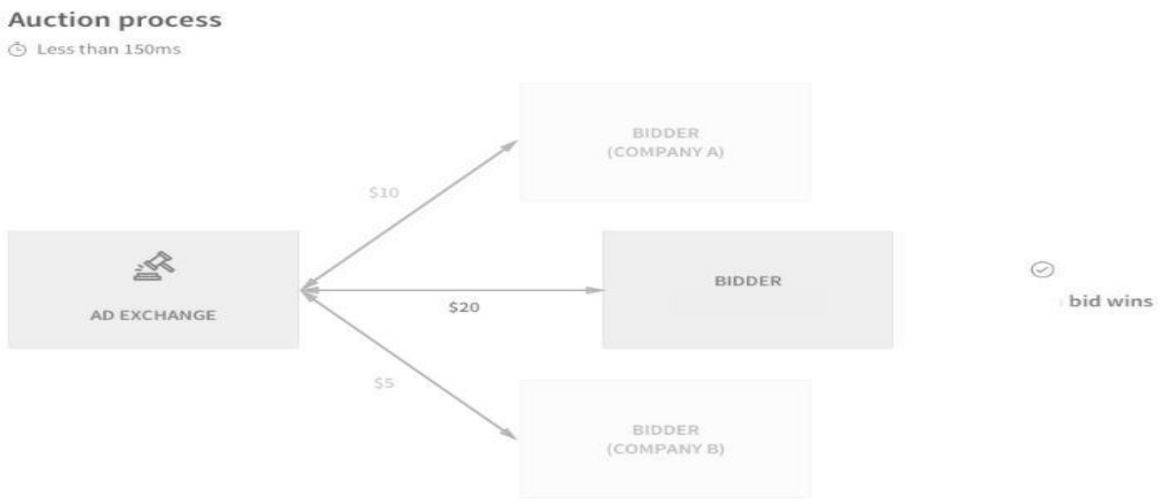
- Please read the instructions carefully before starting.
- **Please make sure you read the deliverables section before submitting your assignment.**
- The instructions below include the expected time of finishing this assignment. If you spend significantly more time than expected then please check if your implementation does more things than the assignment asks for.
- Please keep your solution as simple as possible! We are impressed by simple and elegant solutions.
- If something is not clear to you, please do not hesitate to get in touch with us for clarifications. We'd love to help you out!

Objective

You are a software engineer for an advertising technology company called Project Agora. The company wants to enter the world of real-time bidding (RTB) and you are asked to implement a real-time bidder for mobile advertising campaigns.

What is a real-time bidder?

A bidder is simply a platform which allows advertisers to submit bids to buy mobile ad space in real-time. A bidder receives ~~bid requests~~ from 3rd-party ad exchanges and responds back with a bid response. This bid then competes with bids from other bidders in a real-time auction at the exchange. The highest bid (in terms of bid price) wins and gets to show its ad. You can see this process illustrated below.



For this assignment we will focus on building a bidder so you do not need to worry about what happens at the ad exchange. If we zoom into what a bidder does then the typical flow is outlined below:

- Bidder receives and parses a bid request from an ad exchange.
- Bidder retrieves all the available campaigns from its campaign pool.
- Bidder filters out campaigns that do not match the targeting criteria.
- If there are matching campaigns then the bidder finds the highest paying campaign and submits a bid for that campaign to the ad exchange.
- Otherwise, the bidder submits an empty response with no bid.

Bid Request/Response Info

The bidder receives bid requests from an ad exchange and it responds back either with a bid or without one.
The bid request from the ad exchange contains info that is needed by the bidder to perform its operation.

Bid Request Info

- bid id: The unique identifier of the bid request
- mobile app info: the id and name of the mobile app that asking for an ad
- mobile device info: the device_id, the mobile operating system and its geo location info (Latitude, Longitude, country)

Bid Response Info

- bid id: The unique identifier of the bid request
- campaign_id: The campaign id of the winning campaigns
- price: The price of the winning campaign
- ad creative: The JavaScript code of the creative that is going to be rendered

Campaign Info

- id: The id of the campaigns
- name: The name of the campaigns
- price: The price of the campaign
- ad_creative: The JavaScript code of the ad that is going to be rendered
- targetedCountries: The countries that are going to be targeted for these campaigns. ALL COUNTRIES must be included as a value.
- targetedLocations: The Locations that are going to be targeted (Location is defined as the combination of the Latitude and Longitude ranges). ALL LOCATIONS must be included as a value.

Task

Objective

The main objective is to design and implement the basic flow of a bidder which was described above. In order to do this, you need to perform the following tasks:

1. Design the Bidder API

2. Design the Campaigns API

3. Design the Bidder Service

4. Implement the Bidder Service in the language of your choice

5. Design the End To End Test Cases for the Bidder Service

6. Implement the End To End Test Cases for the Bidder Service with the framework of your choice

7. Host your code in the version control of your choice

8. Deploy the Bidder Service in the platform of your choice (either locally or remotely)

Additional Info

Targeting

1. For the targeting matching you should use the country and location info contained in the bid request and check it against the list of *targetedCountries* and *targetedLocations* for each campaign.
2. For the campaign ranking you should use the *price* info of the campaign. The campaign with the highest price wins the bid.

Mocking the Campaign API

Your bidder implementation depends on an external service which exposes the **Campaign API**. There is no instance of this service running, therefore in your tests, we expect to see this dependency mocked to simulate the real production functionality.

You are free to use any approach you like to implement mocking.

End-to-end test cases

In order for the task to be completed your solution should implement automated tests. You are free to design yourself the test cases you think that are necessary for the Bidder Service to operate as expected.

Deployment

In order for the task to be completed, your solution should implement a CI/CD process. You are free to design and implement the CI/CD process in the way you want.

Deliverables

In your uploaded solution you should include:

1. The complete working code (for service and tests) in a public repository.
2. Step-by-step instructions on how to deploy and run the service (either remote or locally).

Notes

- Add the appropriate documentation for the design of code, tests and deployment process.
- Please keep your solution as simple as possible.

- Indicative Completion Time: 8 hours.

Assessment criteria

Below you can find the list of assessment criteria we use to judge your code along with their importance.

Dimension	**Description**	**Importance**
Code architecture	Is separation of concerns clear in your code? Is it split into modules/components or is it a big monolith?	**HIGH**
Simplicity	Does your code fulfil the specifications in the simplest way possible? Did you use complicated code structure or complex data structures that are an overkill for our problem?	**HIGH**
Reusability	Do you follow the DRY or the WET principle? Did you break your code down into reusable services/components?	**HIGH**
Extensibility	Is your code structured in a way that future changes would require minimal effort?	**HIGH**
Solution Completness	Did you complete the implementation of the solution according to the specifications?	MEDIUM
Test coverage	Did you cover most of your code's functionality with tests?	MEDIUM
Performance	Did you make an effort to optimize the performance of your code?	LOW