Introduction

Thank you for your continued interest in Bluehost / Endurance! I really enjoyed speaking with

you and would like to continue the interview process with a technical exercise.

For the coding portion, you may work in any language you would like.

Work on the exercise is time-boxed to three hours.

The Exercise

You have been contracted by a new, shoe-string start-up for up to three hours of work in which

you are expected to provide:

1. An architecture design of service(s). Documentation with visual examples encouraged.

2. A mock API service prototype. Include all source code, and it should be something that could

be run/interacted with locally by start-up engineering reps.

Your work should be delivered as a shared GitHub repo.

The service(s) will support a mobile app, which the founders assure you will go viral within a

month of its initial launch.

The service(s) will provide the following functionality:

● Storing location data from cellular transmitters attached to pet collars and collected by

units installed on cell towers.

● Returning historical location data for a given pet (basic users can see the last 24 hours

while premium users can see the last 30 days of data).

● Searching for and returning contact information for opted-in owners of other pets within a

5-mile radius. [Note: This is premium functionality.]

You will provide an architecture diagram for the service(s) and related infrastructure. The

founders are not available for consultation; list significant assumptions you make in your design

(including work to be done in different phases of development).

The service prototype will use canned location data and will be used for initial integration as the

mobile app is developed. Again, list significant assumptions made in creating the prototype.

Introduction

Thank you for your continued interest in Bluehost / Endurance! I really enjoyed speaking with

you and would like to continue the interview process with a technical exercise.

For the coding portion, you may work in any language you would like.

Work on the exercise is time-boxed to three hours.

The Exercise

You have been contracted by a new, shoe-string start-up for up to three hours of work in which

you are expected to provide:

1. An architecture design of service(s). Documentation with visual examples encouraged.

2. A mock API service prototype. Include all source code, and it should be something that could

be run/interacted with locally by start-up engineering reps.

Your work should be delivered as a shared GitHub repo.

The service(s) will support a mobile app, which the founders assure you will go viral within a

month of its initial launch.

The service(s) will provide the following functionality:

● Storing location data from cellular transmitters attached to pet collars and collected by

units installed on cell towers.

● Returning historical location data for a given pet (basic users can see the last 24 hours

while premium users can see the last 30 days of data).

● Searching for and returning contact information for opted-in owners of other pets within a

5-mile radius. [Note: This is premium functionality.]

You will provide an architecture diagram for the service(s) and related infrastructure. The

founders are not available for consultation; list significant assumptions you make in your design

(including work to be done in different phases of development).

The service prototype will use canned location data and will be used for initial integration as the

mobile app is developed. Again, list significant assumptions made in creating the prototype.

Introduction

Thank you for your continued interest in Bluehost / Endurance! I really enjoyed speaking with

you and would like to continue the interview process with a technical exercise.

For the coding portion, you may work in any language you would like.

Work on the exercise is time-boxed to three hours.

The Exercise

You have been contracted by a new, shoe-string start-up for up to three hours of work in which

you are expected to provide:

1. An architecture design of service(s). Documentation with visual examples encouraged.

2. A mock API service prototype. Include all source code, and it should be something that could

be run/interacted with locally by start-up engineering reps.

Your work should be delivered as a shared GitHub repo.

The service(s) will support a mobile app, which the founders assure you will go viral within a

month of its initial launch.

The service(s) will provide the following functionality:

● Storing location data from cellular transmitters attached to pet collars and collected by

units installed on cell towers.

● Returning historical location data for a given pet (basic users can see the last 24 hours

while premium users can see the last 30 days of data).

● Searching for and returning contact information for opted-in owners of other pets within a

5-mile radius. [Note: This is premium functionality.]

You will provide an architecture diagram for the service(s) and related infrastructure. The

founders are not available for consultation; list significant assumptions you make in your design

(including work to be done in different phases of development).

The service prototype will use canned location data and will be used for initial integration as the

mobile app is developed. Again, list significant assumptions made in creating the prototype.

Introduction

Thank you for your continued interest in Bluehost / Endurance! I really enjoyed speaking with

you and would like to continue the interview process with a technical exercise.

For the coding portion, you may work in any language you would like.

Work on the exercise is time-boxed to three hours.

The Exercise

You have been contracted by a new, shoe-string start-up for up to three hours of work in which

you are expected to provide:

1. An architecture design of service(s). Documentation with visual examples encouraged.

2. A mock API service prototype. Include all source code, and it should be something that could

be run/interacted with locally by start-up engineering reps.

Your work should be delivered as a shared GitHub repo.

The service(s) will support a mobile app, which the founders assure you will go viral within a

month of its initial launch.

The service(s) will provide the following functionality:

● Storing location data from cellular transmitters attached to pet collars and collected by

units installed on cell towers.

● Returning historical location data for a given pet (basic users can see the last 24 hours

while premium users can see the last 30 days of data).

● Searching for and returning contact information for opted-in owners of other pets within a

5-mile radius. [Note: This is premium functionality.]

You will provide an architecture diagram for the service(s) and related infrastructure. The

founders are not available for consultation; list significant assumptions you make in your design

(including work to be done in different phases of development).

The service prototype will use canned location data and will be used for initial integration as the

mobile app is developed. Again, list significant assumptions made in creating the prototype.

**Solution Approach**

**Over All Infrastructure**:

Collecting towers enriches incoming data with own id and sends to Pet Manager Service.

{towerId}

Persistence data provider

Pet Management System

User Service

/PetManagement/UserService

Pet Location service

/PetManagement/PetService

Pet Location service

/PetManagement/PetService

User Service

/PetManagement/UserService

Location data along with pet id sent to collecting towers.

{petId, latitude, longitude, time stamp}

Pet Location service

/PetManagement/PetService

**Assumptions:**

1. Data persistence can be in relational or no sql db based on volume/ throughput expected. For now its stored in memory using simple data structure.
2. jdk1.8.0\_271 , Jersey jaxrs-ri-3.0.2 for rest service, tomcat apache-tomcat-10.0.6-windows-x64 to host, and eclipse eclipse-inst-jre-win64 for development is used.
3. Validations/ exception handling/ authentication/ authorization/ data transformations etc. is skipped in the prototype.
4. Not all the crud methods are implemented. Only the ones described in the problem are elaborated to some extent.

**Data Collection:**

HTML5 provides location api which can be used to capture geolocation of pets. Refer <https://www.w3schools.com/html/html5_geolocation.asp>

**Service Details:**

1. PetLocationService.java provides following endpoints:
2. Used for collecting the location data for pets.

POST /PetService/petLocations

{

        "petId": "1",

        "latitude": 17.386,

        "longitude": 78.4887,

        "towerId": "Tower1",

        "time": "30/05/2021 14:30"

}

b) Used for getting the location log details. Based on the user making the request, date range is validated for 10 or 30 days limit.

GET /PetManagement/PetService/petLocations?petId=1&startTime=30/05/2021 01:30&endTime=30/05/2021 06:30

Header :

UserId : 1 // id of caller making the request. Used in the absence of login frame and OAuth token.

1. Other crud endpoints are skipped for now.
2. UserService.java provides following endpoints:
3. For now its assumed that premium service can be enrolled into at the time of creation itself. Put and other crud can be provided later..

POST /UserService/users

{

        "id": "3",

"address": "Address 3",

        "latitude": 17.385,

        "longitude": 78.4867,

        "name": "User3",

        "premiumUser": **false**,

        "zip": "500051",

        "pet": [{

            "id":"31",

            "name":"Dog",

            "ownerId":"3"

        },{

            "id":"32",

            "name":"Cat",

            "ownerId":"3"

        }]

  }

1. For adding Pets to existing users following is provided.

POST /PetManagement/UserService/users/3/pets

{

            "id":"11",

            "name":"Dog"

     }

Header :

UserId : 3 // id of caller making the request. Used in the absence of login frame and OAuth token.

// this is to ensure only owners can add their pets.

1. For getting user data by location following endpoint is provided:

GET /PetManagement/UserService/users?latitude=1.323&longitude=2.12131&radious=5

Header :

UserId : 2 // id of caller making the request. Used in the absence of login frame and OAuth token to validate premium user can make the call.

For now all the users are returned. Later one of following location service can be used:

Using spatial queries in Oracle DB:

<https://asktom.oracle.com/pls/apex/f?p=100:11:::::P11_QUESTION_ID:9539156000346599825>

Use 3rd party search service like:

<https://www.algolia.com/doc/guides/managing-results/refine-results/geolocation/>

Google Place Search

<https://developers.google.com/maps/documentation/places/web-service/search>