XCCELERATED

proudly part of Xebia

ACCELERATING GROWTH

» Data Engineering Assessment

Assessment

Inroduction

As part of our assessment and hiring procedure for data engineers, Xccelerated asks all candidates to read and complete the following technical implementation assignment. Take some time to read and consider the problem it describes. We've intentionally left it open so you can determine your own approach, solution strategy, and implementation plan. We will discuss your work during the assessment interview. In evaluating your work we will consider the following:

- Ask questions—if any aspect of the assignment is unclear, get clarity from us.
- Apply and demonstrate your current software development skills. We want to know what you can deliver now. It's not necessarily about the latest technology—you'll learn that in our program later.
- The assignment is a minimum of specifications to jumpstart the assessment. Feel free to discuss any enhancements.
- Include the essential elements of your implementation that show it's a high-quality solution.
- We don't dictate a programming language. Choose what you know best. Please keep in mind that our engineers work primarily with Python and Scala, so we like to ask you to inform us if you want to use something else.

The Assignment: The YakShop

Your new client is a yak shepherd who lives on the tundra where he herds his yaks. He occasionally sells yak wool and milk to a few regular customers and has decided to open a shop on the internet to expand his horizons. He hired you to develop a web shop that will help him sell his products to more people. After several meetings together, you've come up with some user stories to define the focus.

From your internet research, you know that yaks age like humans and produce less milk as they grow older. You've also learned that, unlike human years, a yak year is only 100 days.

All of the shepherd's yaks come from the "LabYaks" tribe— known for its consistency in wool quality, milk taste and production rate of said goods. The shepherd gave you the following facts about LabYaks:

- Each day a LabYak produces 50-D*0.03 liters of milk (D = age in days).
- At most every 8+D*0.01 days you can again shave a LabYak (D = age in days).
- A yak's first shave can occur at the age of 1 year.
- A LabYak dies the day he turns 10.

Assumptions

- The moment you open the YakShop online will be day 0, and all yaks will be eligible to be shaved, as the two of you spent quite a lot of time setting up this shop and the shepherd wasn't able to attend much to his herd.
- Each morning the shepherd milks and shaves the eligible yaks. Yaks that aren't eligible for shaving on the exact day, cannot be shaved. For example, a yak who started out on day 0 as 4 years, can be shaved again on day 13.

User stories

Working with a business analyst, the yak shepherd created the following user stories.

YAK-1

As a yak shepherd, I want to be able to read and query an XML file that contains data about my herd.

Input herd.xml:

```
<herd>
     <labyak name="Betty-1" age="4" sex="f"/>
     <labyak name="Betty-2" age="8" sex="f"/>
     <labyak name="Betty-3" age="9.5" sex="f"/>
</herd>
```

Note: The age is given in standard yak years

Your program should include 2 parameters:

1. The XML file to read
2. An integer T, representing the elapsed time in days.

   Note: T=13 means that day 12 has elapsed, but day 13 has yet to begin

Output for T = 13:

```
In Stock:
   1104.480 liters of milk
   3 skins of wool
Herd:
   Betty-1 4.13 years old
   Betty-2 8.13 years old
   Betty-3 9.63 years old
```

Output for T = 14:

```
In Stock:
   1188.810 liters of milk
   4 skins of wool
Herd:
   Betty-1 4.14 years old
   Betty-2 8.14 years old
   Betty-3 9.64 years old
```

YAK-2

As a yak shepherd, I want to be able to query the file about my herd and current stock using HTTP REST services which output JSON data.

The following are the requests you wish to make.

- `GET /yak-shop/stock/T`
  `Returns a view of your stock after T days`
- `GET /yak-shop/herd/T`
  `Returns a view of your herd after T days`

Please keep in mind that I should be able to make a request for day 13 after a request for day 14 and get the correct result.

Samples

Request 1

```
GET /yak-shop/stock/13
```

Response

```
{ "milk" : 1104.48, "skins" : 3 }
```

Request 2

```
GET /yak-shop/herd/13
```

Response

```
{
  "herd" : [
   { "name" : "Betty-1", "age" : 4.13, "age-last-shaved" : 4.0 }
   { "name" : "Betty-2", "age" : 8.13, "age-last-shaved" : 8.0 }
   { "name" : "Betty-3", "age" : 9.63, "age-last-shaved" : 9.5 }
  ]
 }
```

YAK-3

As a yak shepherd, I want my customers to be able to buy from my stock using my HTTP REST services.

Assume that order requests come in ascending order of time. If you cannot fulfill one of the ordered goods because you're out of stock, you deliver the other goods that are entirely in stock.

For instance, if your stock contains 4000 liters of milk and 10 yak hides, and your customer orders 4500 liters of milk and 4 hides, you only deliver the 4 hides (and omit the milk from the result) and give an HTTP status code 206 (partial content). If the full order is not in stock, you only return a HTTP 404 (not found) status code. If the order was placed successfully, you return HTTP status code 201 (created) with the resulting order.

Keep in mind that orders are cumulative. When checking the stock availability, you must take into consideration previously placed orders.

Request

- `POST /yak-shop/order/T`
  where T is the day the customer orders, this means that day T has not elapsed.

Samples

Request 1

```
POST /yak-shop/order/14
{
  "customer" : "Medvedev",
  "order" : { "milk" : 1100, "skins" : 3 }
}
```

Response

```
Status = 201
{ "milk" : 1100.0, "skins" : 3 }
```

Request 2

```
POST /yak-shop/order/14
{
  "customer" : "Medvedev",
  "order" : { "milk" : 1200, "skins" : 3 }
}
```

Response

```
Status = 206
{ "skins" : 3 }
```

YAK-4

As a yak shepherd, I want to be surprised by your ingenuity so that I can impress my fellow shepherds. You can do anything we haven’t thought of and dazzle us at our monthly shepherd meeting.

Timeframe

If you can make the assessment in approximately 15 hours of work that is a good sign. If you need a little bit more that's ok too. However, do not spend a lot more hours on it. It's better to deliver something simple and working than nothing at all.

Also, let us know if you feel that the time estimate is inaccurate. This feedback is valuable for us.

Could you send us your solution one day in advance of the presentation date? As it is not a public assessment, also the question to keep your repository private.

If you have any questions please feel free to reach out to our Lead Engineer; Emanuel.

Emanuel Lal
Engineering Lead
✉ elal@xccelerated.io

XCCELERATED

proudly part of Xebia

Xccelerated B.V.
Sarphatistraat 640 - 642
1018 AV Amsterdam

www.xccelerated.io
hello@xccelerated.io
+31 (0) 20 76 09 842