

iOS Tech Test

Candidate Brief

The purpose of this test is for you to demonstrate your knowledge of the platform by building a small app. Along with this brief you should have been provided with an Xcode project that includes some starter code. Please use this project to build your solution on top of.

The project is to build a product listing page that will link through to a product page. We have implemented the product page UI already but you will need to provide it with the required data. All the APIs you will need are documented here. The starter project includes some design files that illustrate how your product listing page should look. Finally, there is a requirement to update the product listing page to include badges.

The test is designed to take around 3 hours but there is no time limit. We suggest completing the tasks in order and submitting what you have done if you choose to end at 3 hours and haven't finished.

Instructions:

1. Read through the provided code, it should all be useful.
2. Build a product listing service that fetches products from `/api/products`.
3. Produce a product listing page to the design document "Product Listing.pdf" included in the project using this api `/api/products`.
4. When a customer selects a product, use the product api `/api/product/{id}` to fetch the product data and present the product view controller supplied with the project.
5. Amend the product listing to include badges using the design document "Product Listing with Badges.pdf".
 - i. Using the `/api/user/{id}/offers` api fetch the offers and "available_badges" string for the given user id. This should happen as

you launch the app and before you show any products.

- ii. Using the "offers_ids" parameter in the product model from the `/api/products` response. Combine the results with the user offers response using the following process:
 - a. Decode the offer string e.g.
"sale:PRIORITY_ACCESS||loyalty:SLOTTED,BONUS" from `/api/user/{id}/offers` as detailed below:
 - a. Split the string on "||" to get available badges in priority order.
 - b. For each badge string separate in to two parts around ":". The first half is the "badge name". The second half is the comma delimited list of "badge types".
 - c. The asset for the badge can be accessed at this path:
`/image/{badge name}_icon.jpg`.
 - b. For each of the "offer_ids" from a product from check to see if the user has any matching offers (where the ids are the same).
 - c. Of the matching offers check if the offer type has a badge associated with it. Show the highest priority badge if more than match.
6. Demonstrate testing.

Sample Data

API is available on `http://interview-tech-testing.herokuapp.com` using the paths provided. The API requires basic auth with the username: admin and password: password. Full documentation is included in the project file. Sample users with ids 1,2,3,4, and 5 (`api/user/{id}/offers`) are available with different combinations of offers and available badges All the products returned from the `/api/products` call are available in the product api `/api/product/{id}`

Other Requirements

- Please do not use any third party code or libraries.

- Structure your code using MVVM.
- The code you submit must build and run.
- Your project need only support iPhone in portrait but should match the designs across different devices.

How to submit

Please provide either a link to a public or private git repo or a zipped folder containing the repo. If you don't complete all the tasks please include a readme describing the steps you would need to take to finish.