

Take-Home Assignment (Web Client)

Thanks for taking some time from your busy schedule to work through this assignment; we certainly appreciate it. Our goal here is to learn about your coding style, use of frameworks/libraries, and approach to development in general.

We value testability at Trōv. Many of us come from teams with few (if any) dedicated testers so we expect developers to ensure their code executes as flawlessly as possible. We still have project owners, stakeholders, and others to help with acceptance and usability testing - but the main responsibility for quality remains with the developers. We ensure that any estimate and deliverable for a feature include unit tests or some form of automated testing.

The exercise is on the following page along with the list of deliverables. Please be sure to include answers and any other notes/comments/design decisions in a `README.md`. One of our tenets is “it doesn’t exist if it’s not written down”, so feel free to over-communicate in that document. This is your chance to show us what you consider to be great code (and tests)! We really enjoy seeing a passion for development surface early in this process. When you are finished, please reply to this email with a link to your repository.

Looking forward to seeing your code and answers!

The Gilded Rose Expands

As you may know, the Gilded Rose* is a small inn in a prominent city that buys and sells only the finest items. The shopkeeper, Allison, is looking to expand by providing merchants in other cities with access to the shop's inventory via a public HTTP-accessible API.

API requirements

- Retrieve the current inventory (i.e. list of items)
- Buy an item (user must be authenticated)

Here is the definition for an item:

```
class Item
{
    public string Name { get; set; }
    public string Description { get; set; }
    public int Price { get; set; }
}
```

A couple questions to consider:

- How do we know if a user is authenticated?
- Is it always possible to buy an item?

Please model (and test!) accordingly. Using a real database is not required.

Deliverables

1. A system that can process the two API requests via HTTP
2. Appropriate tests (unit, integration, etc)
3. A quick explanation of:
 - a. choice of data format. Include one example of a request and response.
 - b. what authentication mechanism was chosen, and why

One Small Reminder

The purpose of this exercise is for you to demonstrate your current skills, not pick up a new library/framework/workflow/IDE. We know good developers can adapt, so for this exercise please use your preferred stack to deliver working and properly-tested code.

* The Gilded Rose is a [refactoring/TDD exercise](#) created by Terry Hughes and Bobby Johnson. You can safely ignore it during the coding exercise, but it might be fun to try out later.