

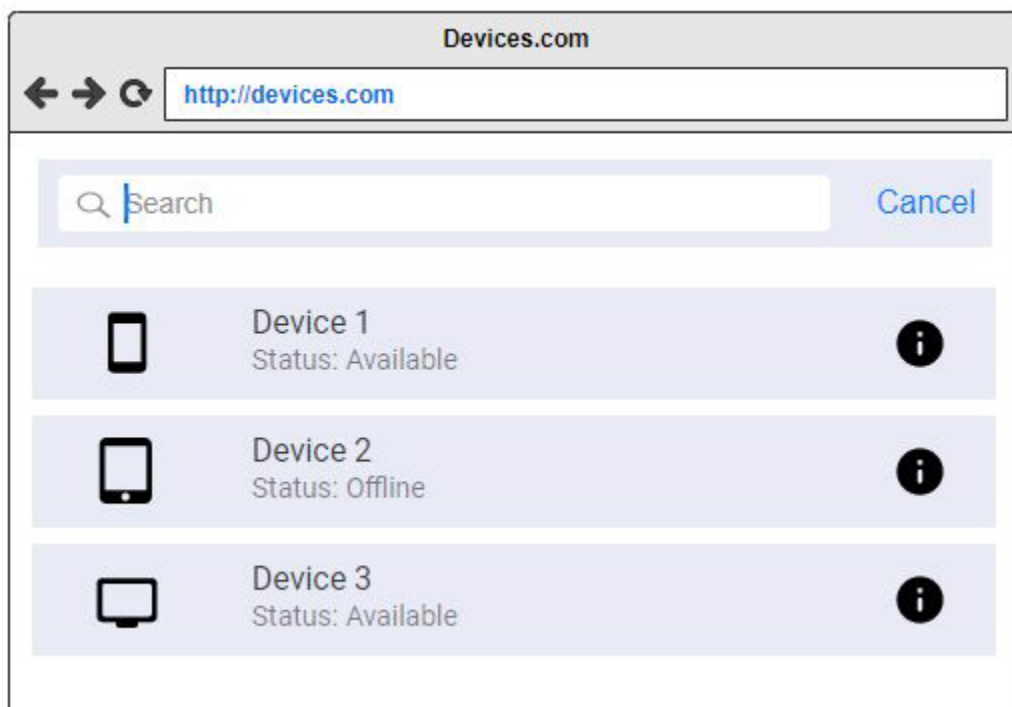
PART I

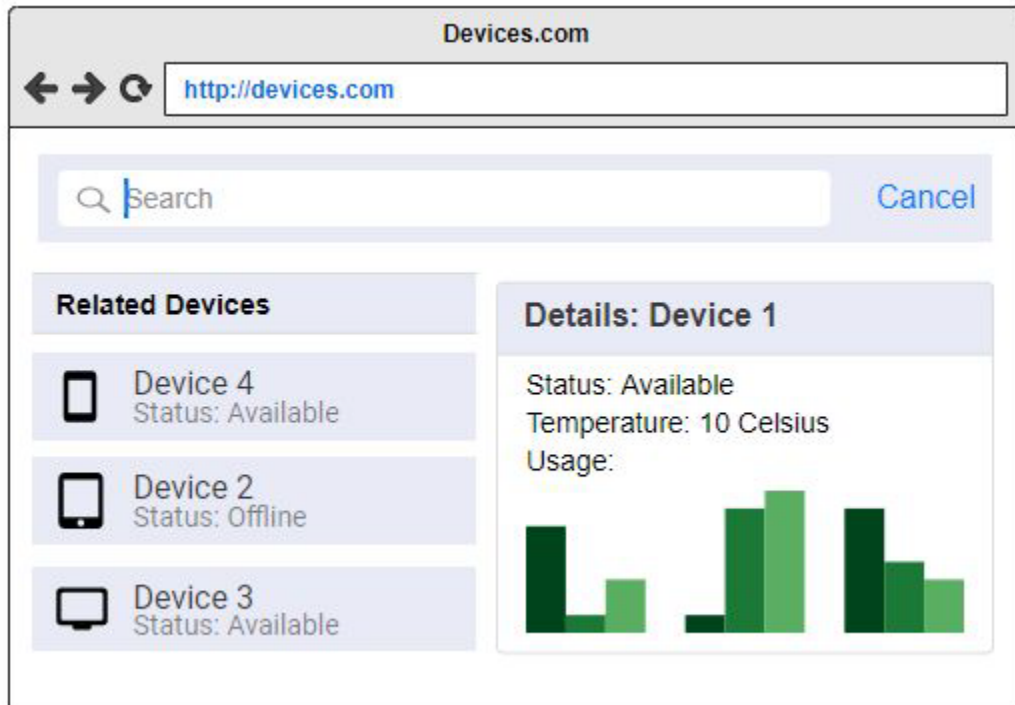
Devices.com - Full-Stack Developer - Exercise

Overview

You are on a team building a fictional web app called devices.com. It shows a complete list of all the devices on the home page. Users can click on a device and are taken to the device details page.

The details page shows more information about the device, as well as a list of related devices. There is also a common search bar on each page, a device icon for each device (devices can be different types), some information about status, temperature and a data representation for the usage of the Device.





Coding Exercise

You need to implement backend and front-end parts for devices.com.

Outcome:

Backend part:

1. .NET Core API application with the list of the API endpoints described in the previous question.
2. (Optional) Implement any type of basic authentication to access API endpoints
3. (Optional) Add Unit Tests to the solution
4. SQL DB with the set of tables necessary for the application
 - a. Pre-populate DB with some fake data
 - b. Data Layer: EntityFramework

Front-end part:

1. Angular 10+ Web (preferably using Node V12+) application with the following views:
 - a. A home page containing a list of devices, and a search bar for filtering devices by name
 - i. **Search bar should be functional**
 - b. A device details view with details displayed for the selected device, a list of related devices, as well as a search bar



- i. **Search bar is non-functional**
 - ii. **Device details usage does not require a functional graph you can use an image as a placeholder**
2. For these views include some demo data (could be simple JSON) and bind it to the elements/components.
 - a. Preferably mocking an HTTP response from an API service is recommended
3. Please follow the styling css/scss as close as possible to the provided wireframes (specific colours and fonts are not important just keep it consistent)
4. (Optional): Add **ONE** unit test which tests that all of the expected devices render on the home page
5. Your application **MUST** compile

PART II

A user story is written in two different styles below. Please read both, and answer the questions that follow. Point-form answers are acceptable. If you are not certain about an answer, feel free to provide your thoughts.

Style #1

As a developer, I want the system to show me information about a device, so I can manage my Hardware

Acceptance Criteria:

- The system should show the device and device status
- The system should show useful details about a device when I click on it
- The system should show the device, using the proper icon
- The system should be fast

Nice to have

- I can view a dashboard of my most used devices

Style #2

As a person planning to manage many devices, I want to see information about my device, so I can manage my hardware

Acceptance Criteria:

- Ensure all devices and device statuses are displayed
- Ensure useful details about a device are displayed when clicked on.
- Ensure devices are shown using the proper icon
- Ensure the system is fast

Nice to have



- I can view a dashboard of my most used devices

Questions

From a testing perspective:

1. Which user style do you prefer, and why?
2. If not, what improvements would you suggest making?
3. What are some tests you would write to ensure the behaviour functions accordingly?