ambient

To begin, thank you for taking the time to meet with the Ambient engineering team. We strive to deliver a less stressful, more collaborative experience when progressing through the interview process. While whiteboard problem solving does have its place, we find that it doesn't easily enable practical discussion like one would experience on a day to day basis in an engineering environment. By providing this take home assessment, we hope to create a scenario that would more closely represent the interactions between engineers working on a common codebase.

For this assessment, we hope that you will be able to spend no more than a few hours working on the solution, and do so as your time permits. We encourage you to use whatever language and paradigm that you are most comfortable with. While the solution does not need to be fully complete, according to the specification below, we do ask that it at least be able to be compiled/interpreted as defined by the language you select. If you have any questions, don't hesitate to reach out to recruiting (raul.sierra@ambientproptech.com or ashlee@ambientproptech.com), or a member of the engineering team.

## Paired device manager overview

Let's begin with a scenario. You have been asked to prototype a tracking system that allows for devices to be paired, removed and updated for a given dwelling. To give you a little more context, when working with smart IoT devices there is a requirement for these devices to be able to communicate to send details and receive commands. These devices often communicate over different technologies and this often requires another device, we will call it a hub, to help communicate with different devices. This act of adding a device to the known list of devices for a hub is called pairing.

For this project we would like for you to design a solution that tracks and manages devices and hubs. Because these hubs and devices will all be in a dwelling we would like that to be

included in the solution as well. This means that, at a minimum, there should be the following entities represented in your solution:

1. Dwelling - A living space where hubs and devices can be installed
2. Hub - A hardware piece that interacts with devices
3. Devices
    a. Switch - A device that can be turned on and off
    b. Dimmer - A device that provides variable lighting
    c. Lock - A door lock that can be open/shut and can have a pin code to enter
    d. Thermostat - A device for controlling the heat/cool levels of the dwelling

Each of the defined entities have some operations that apply to them, for this solution we would like to see you implement the following operations:

- Device
    - Create - Create a new device
    - Delete - Delete a device that is not currently paired
    - Info - Retrieve the current state of a device
    - Modify - Change the state of the device
    - List - List all devices
- Hub
    - Pair Device - Pair a previously created device
    - Get Device State - Get information about a device's current state
    - List Devices - Get a list of devices paired to a hub
    - Remove Device - Unpair a device from a hub
- Dwelling
    - Occupied - A new resident has moved into the dwelling
    - Vacant - A resident no longer resides in the dwelling
    - Install Hub - Associate a hub with a dwelling
    - List Dwellings - Get a list of all dwellings

Since this is a PoC, there is no requirement that the solution implement a REST interface, UI or GUI of any sort. It will be adequate enough to prove the design by demonstrating operations via a *driver function*. In our use case, a *driver function* is defined as a function that is written to demonstrate operations from a library-style code implementation. For example, if you write function update(), a driver function would be a function that contains code demonstrating how you would call function update().

**NOTE:** This project does not need to use an official database, any form of in-memory solution would be adequate and preferred. Also be sure to include instructions in how to use the provided solution.

## Project Submission

Once you have completed the project, be sure to submit all source code, documentation, tests, etc. that you have created. This submission can be through email, source control or some other means you are comfortable with.