i3sdk

Release v1

Dec 18, 2019

Contents:

This documentation is meant for the users of I3 data marketplace. We recommend the users to follow this document to create, manage, and buy data products from the I3 data marketplace.

CHAPTER 1

Overview of I3 Data Marketplace

1.1 Introduction

The I3 data marketplace is a community-driven IoT platform to allow the device owners to sell their application data to help the application developers. The users are required to follow the guidelines provided in this documentation to easily navigate through the frontend.

1.2 Terminologies

Marketplace: The marketplace refers to an I3 instance made available by an organization, including a city or a community. An open-source version of I3 data marketplace software is available here.

Seller: A seller is the owner of a device or the data, and he or she is willing to make the data available via the data marketplace.

Buyer: A buyer is the one who is buying data from the I3 data marketplace to create an application.

Broker: One of the core components of the I3 architecture is a data broker. The current version of I3 uses MQTT as a data broker, which publish-subscribe messaging protocol for the IoT.

Publisher: In the publish-subscribe messaging protocol, data producers are considered as a publisher. Each data producer is required to publish the data to the I3 data marketplace after creating the data product on the marketplace.

Subscriber: A subscriber is a data consumer. In the I3 data marketplace, the data buyers are required to subscribe to the data after purchasing the data product on the data marketplace.

Topic: The topic captures the metadata in a human-readable form. In the publish-subscribe communication model, the publisher and the subscriber exchange information with each other through topics. For example, a seller providing “temperature” data could name the topic as “temperature.” When a buyer is building a temperature monitoring application can buy data from “temperature” to get access to the temperature feed.

CHAPTER 2

Requirements and Installation

To follow the instructions in this documentation, users need access to an I3 instance. And, the data marketplace frontend has been tested on browsers such as Chrome and Firefox.

The sellers and buyers are required to publish and subscribe to data, respectively. For this process, we recommend the users to use an MQTT client. Paho-MQTT-python is a Python implementation of MQTT. We provide sample publish-subscribe scripts for the I3 users: I3-SDK.

First of all, install paho-mqtt-python, using pip as follows:

```
$ pip3 install paho-mqtt==1.3.1
```

Here we specify the version as 1.3.1 while installing. The Eclipse Paho project is an excellent source of MQTT clients, which contains implementations in C, Java, Javascript, Python (contributed from the mosquitto project), Lua, C++, embedded/minimal C, and Go.

CHAPTER 3

I3 Marketplace

Devices in I3 marketplace are registered under hubs, and also require valid input when creating. Invalid input will redirect you to the device create page without further prompt.

Device names are auto tagged as {USER_NAME}${HUB_NAME}${DEVICE_NAME}.

'$' is used as a separator, instead if '/', because '/' is considered dangerous username in mqtt authentication check, and '/' is necessary for mqtt topic authorization wildcard matching.

The other "invisible" user interface is the MQTT pub/sub broker, equipped with an authentication-authorization plugin, plus a Json Web Token signature verification function (built with pyjwt and c-python connector inside the plugin). The **1883** and 8883 (MQTT+TLS, currently not opened yet) ports are exposed for publish and subscribe.

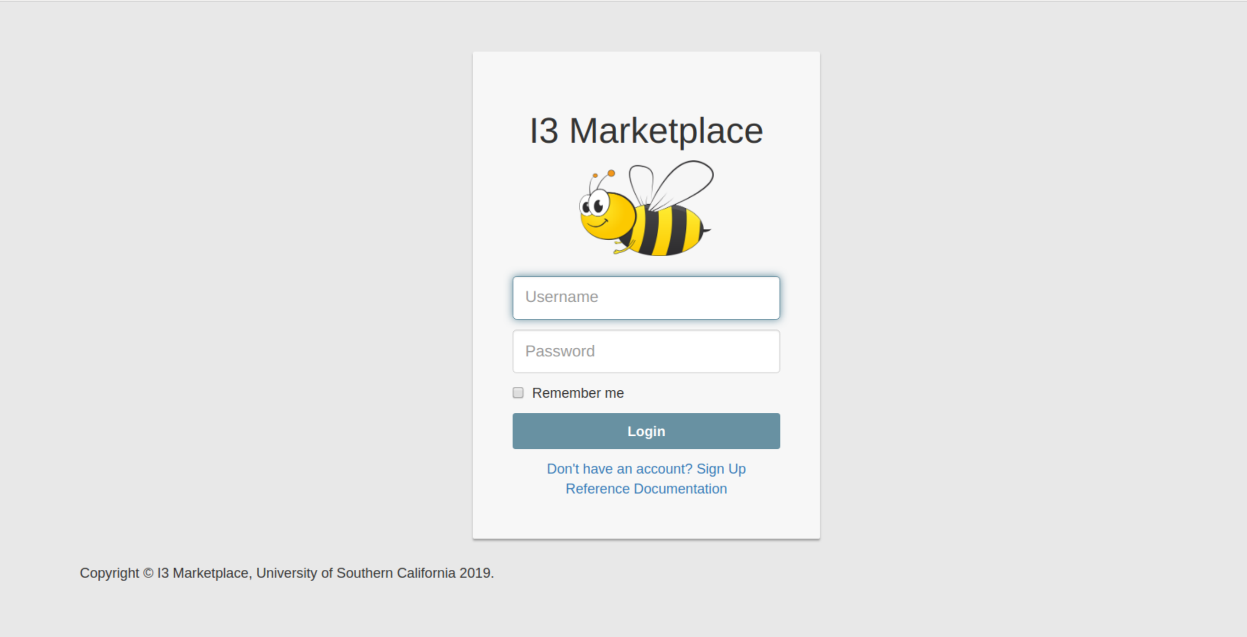
MQTT connect requests mainly consist of five parts: username, password, topic, socket, pub or sub. For sellers: In the pub script, there's nothing related to the seller himself. It was really the seller's device that's publishing, the device has a name, credential and permission. But of course, the devices are not alive, so the seller has to do everything for them.

CHAPTER 4

Account Creation

4.1 Signup

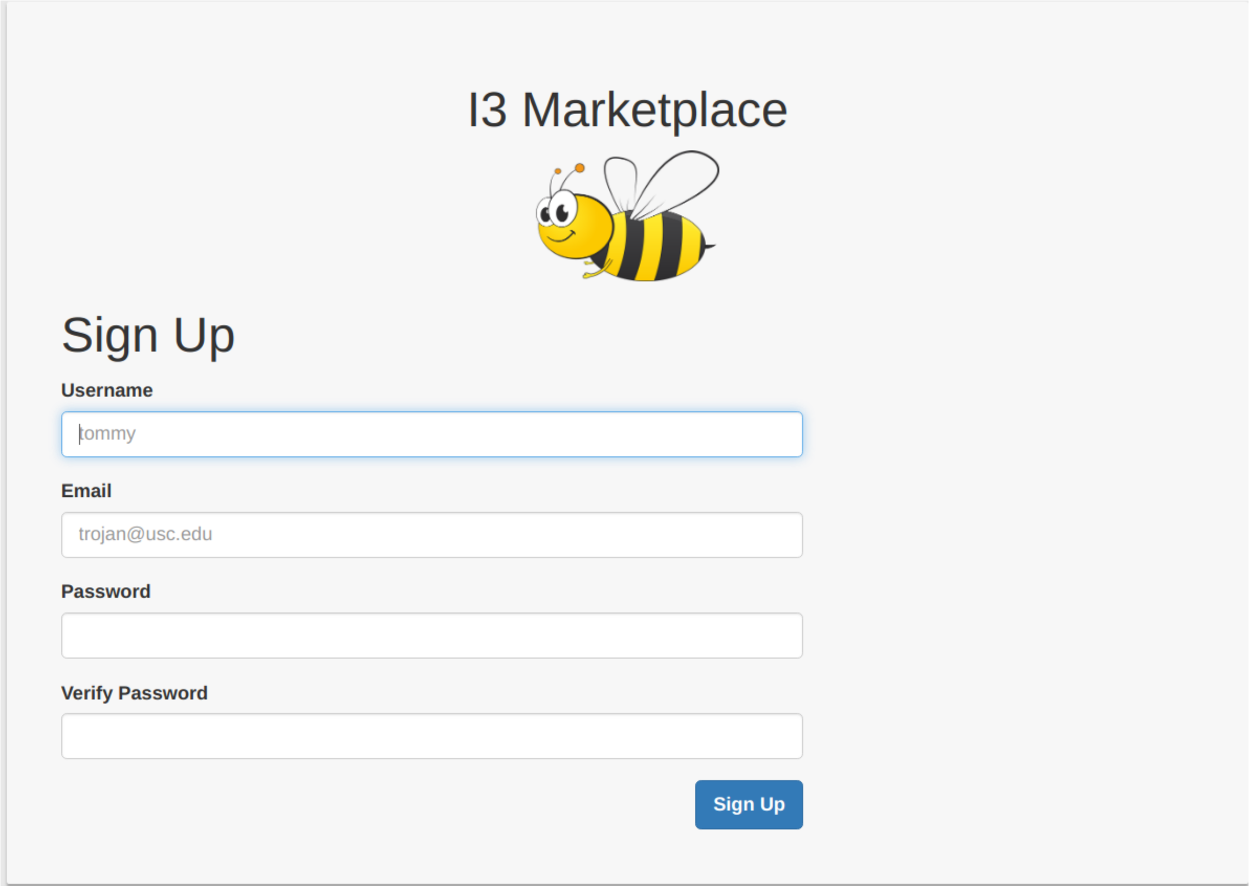
To create, manage, and buy products from the I3 data marketplace, users are required to create an account. When the access the frontend of the I3 marketplace, the platform asks the user to enter their username and password. There is a signup page for the new users.



I3 Marketplace

Copyright © I3 Marketplace, University of Southern California 2019.

The new users have to create an account by clicking "Don't have an account? Sign Up". A new user has to enter an username, email address, and password to set up an account.



I3 Marketplace

Sign Up

Username

tommy

Email

trojan@usc.edu

Password

Verify Password

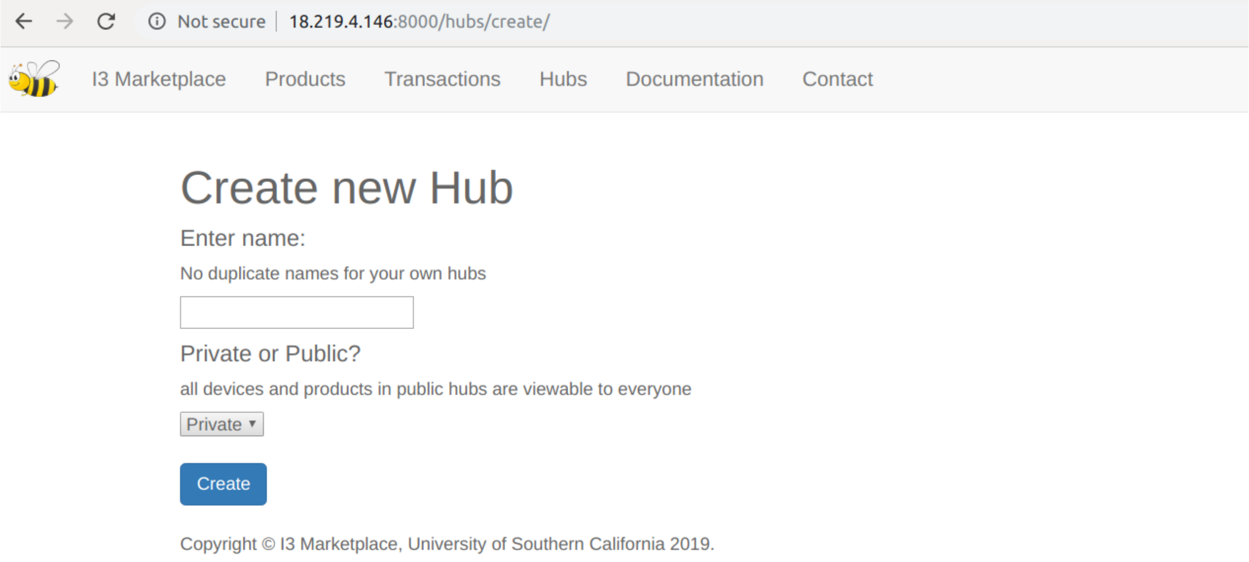
The current prototype of the I3 data marketplace did not undergo a security audit, therefore, we recommend the users not to reuse the username and password from online platforms.

CHAPTER 5

Seller's Guide

5.1 Hub

Products and devices should be registered under a certain hub, so the first thing you should do is to create a hub. Navigate to the hub page on the upper panel, you'll see a list of your current hubs. Click "create hub" to create a hub. The private/public is a hub attribute, if you want others to be able to view the details of your hub (devices and topics under it), choose public. Otherwise choose private. By default, in the same hub, all devices have permission to publish to all topics. No publish permissions across hubs. In the future, you'll be able to assign fine-grained publish permissions for each device (maybe to topics outside its hub), but of course, still restricted to your own products.



Create new Hub

Enter name:

No duplicate names for your own hubs

Private or Public?

all devices and products in public hubs are viewable to everyone

Private ▾

Create

Copyright © I3 Marketplace, University of Southern California 2019.

The detail view of a hub is accessed by /hubs/{HUB_PRIMARY_KEY}, but for hub creators, buttons will be provided for them to see their hub details (see blue hub name, which is a link). When you try accessing the detailed view of a hub (by primary key), your permission will be checked. If you created this hub, you'll see a page like this (left private, right public).

You can use the buttons to create or delete devices, and create or delete or edit products.

When you're not the hub creator, if you try to access a private hub detail view, you see this

If you try to access the detail view of a public hub, you see this

Almost the same as the view page of that of a hub creator, but there's no buttons provided for you because you're viewing someone else's hub, so you can't do things like delete etc. Of course, buttons are just a visible access to URLS, so what if someone tries to delete a device or product that doesn't belong to him by entering URLs?

He'll see this.

Only creator gets to make changes to hubs, products and devices. NOTE: as of today, the product edit view doesn't have permission control yet, so anybody can edit any product by using the edit URL.

5.2 Invalid Form

If there's anything invalid in your create form, you'll be redirected to the same create page without further notice. Please read the instructions carefully, such as "no duplicate product names" and "name must be at least 4 digits".

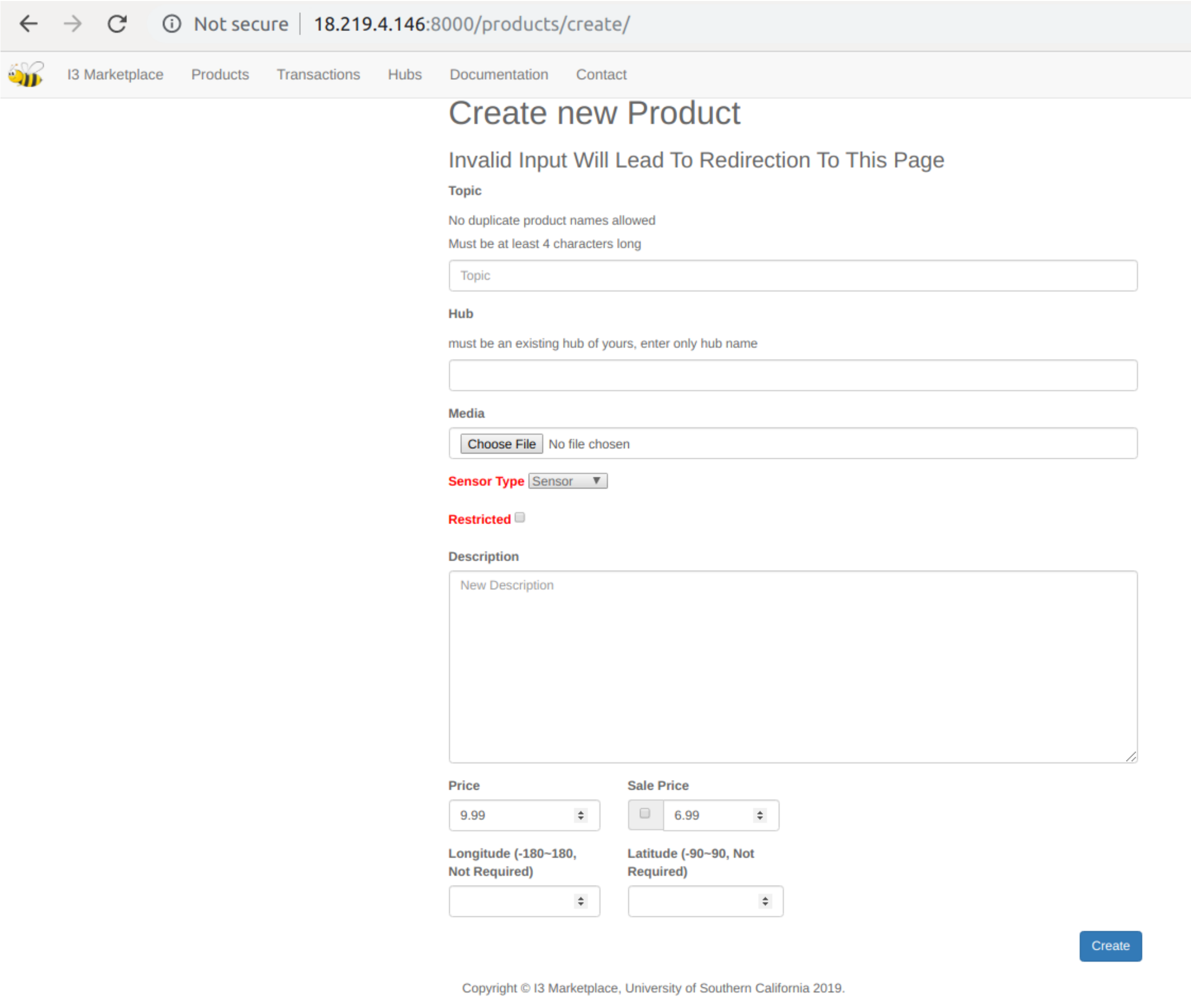
5.3 Naming

When registering, just provide an alpha-numeric name for your hub (also a requirement for products and devices), and the final hub name will be automatically tagged as {USER_NAME}/{HUB_NAME}. NOTE, no duplicate hub names for a single user is allowed. If you registered a hub1, and did it again, you will be automatically redirected to the hub_create view page without extra prompt, and no operation will be performed on the system (as is mentioned above).

5.4 Product Create

Products have to be created inside a hub, and a create button is provided inside hub detail view, and also in the dashboard. They both point to the same URL /products/create.

When creating a product, no duplicate product names inside a hub will be allowed, and the hub name has to be an existing one (eg, for usr1/hub1, don't write usr1/hub1, write only hub1. This is to prevent usr2 from registering a hub that starts with usr1/. Same for devices and products). If these conditions are not met, you will be redirected to the product create page without further prompt. When creating a product, the final product name is automatically tagged as {USER_NAME}/{HUB_NAME}/{PRODUCT_NAME}. This is to make sure that all product names are unique and structured. Detail_view of all products are accessible to all users (login-required), they can be accessed from either marketplace or product link, or /products/{PRIMARY_KEY} (such as /products/1/) or /products/ {PRODUCT_SLUG} (such as /products/usr11hub1topic1/). Also, you can provide an optional location for your product so that it's easier for the buyers to check it out.



Create new Product

Invalid Input Will Lead To Redirection To This Page

Topic

No duplicate product names allowed

Must be at least 4 characters long

Topic

Hub

must be an existing hub of yours, enter only hub name

Media

Choose File No file chosen

Sensor Type Sensor ▼

Restricted □

Description

New Description

Price: 9.99

Sale Price: □ 6.99

Longitude (-180~180, Not Required)

Latitude (-90~90, Not Required)

Copyright © I3 Marketplace, University of Southern California 2019.

5.5 Sensor Type

Sensor' means buyer only gets to subscribe to the topic, 'actuator/both' means buyer can also publish to this topic. Depends on whether you want the buyer to have publishing permission to your topic.

5.6 Restricted Product

If this box is checked, anyone who bought this product will prompt you a notification and a request, and you can choose to approve or decline. Buyer will get the feedback as notification. Approve/decline is using ajax, and when approved, buyer's name will be put into acls table for authorization.

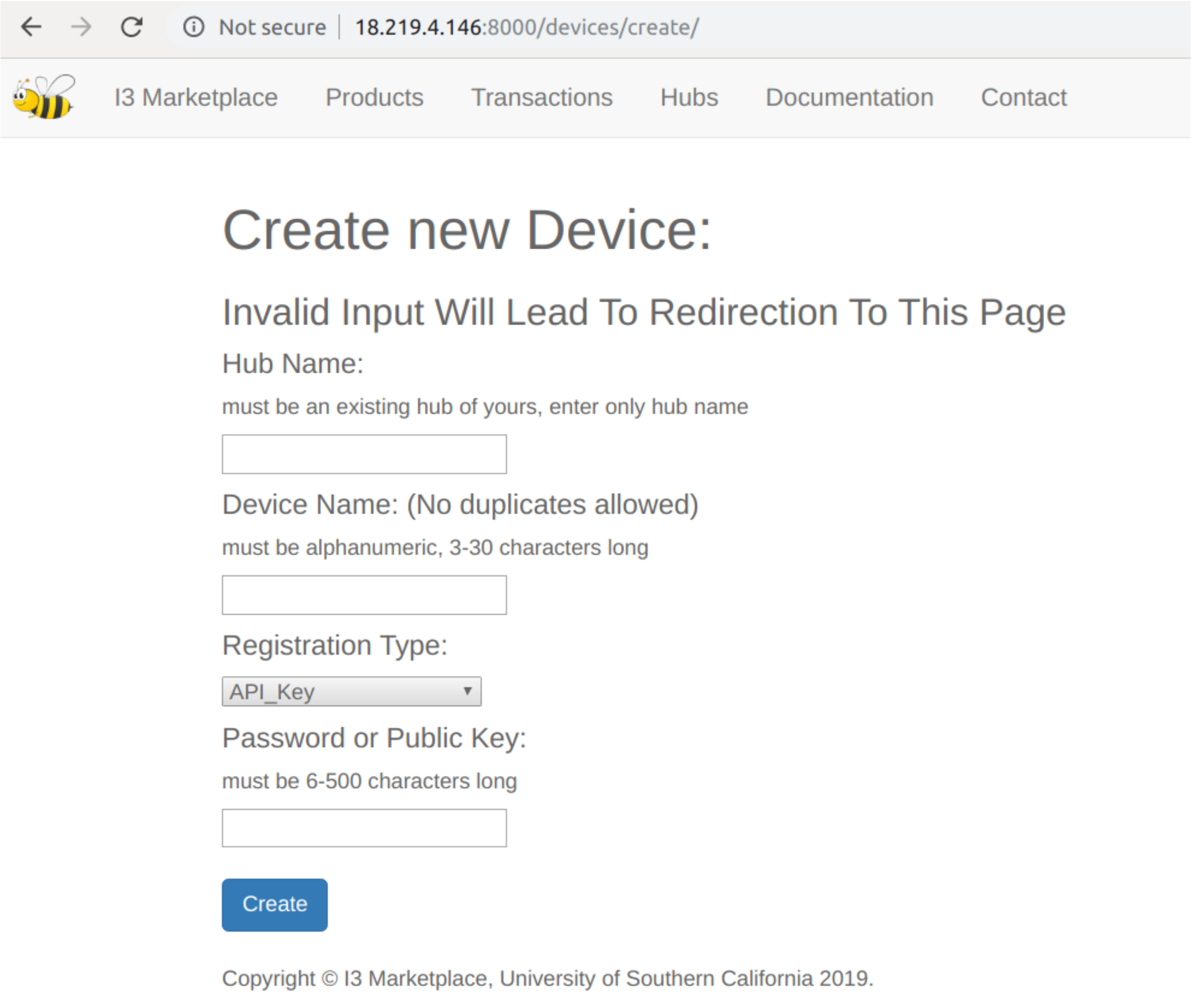
Buyer of a restricted product sees this

5.7 Deleting a Device or Product

When seller deletes a device, only himself is notified; but when he deletes a product (which is NOT supposed to be the usual circumstance), all buyers of this product are notified. Of course, all buyers lose access to this topic, but their personal buying credential remain unchanged. As I’ve mentioned before, it’s specific to buyers, not products.

5.8 Device Registration

Devices are registered under hubs, and also require valid input when creating. Invalid input will redirect you to the device create page without further prompt. Device names are auto tagged as {USER_NAME}${HUB_NAME}${DEVICE_NAME}. ‘$’ is used as a separator, instead if ‘/’, because ‘/’ is considered dangerous username in mqtt authentication check, and ‘/’ is necessary for mqtt topic authorization wildcard matching.



Create new Device:

Invalid Input Will Lead To Redirection To This Page

Hub Name:

must be an existing hub of yours, enter only hub name

Device Name: (No duplicates allowed)

must be alphanumeric, 3-30 characters long

Registration Type:

API_Key

Password or Public Key:

must be 6-500 characters long

Copyright © I3 Marketplace, University of Southern California 2019.

Two methods to register devices: api key and asymmetric key pairs. For the former, enter a password for your device, and it will be np-hashed in database. For the latter, generate a RS key pair using openssl, open the public key file, and get rid of all characters that are not the key body (including ‘n’s and “—BEGIN PUBLIC KEY—” and “—END

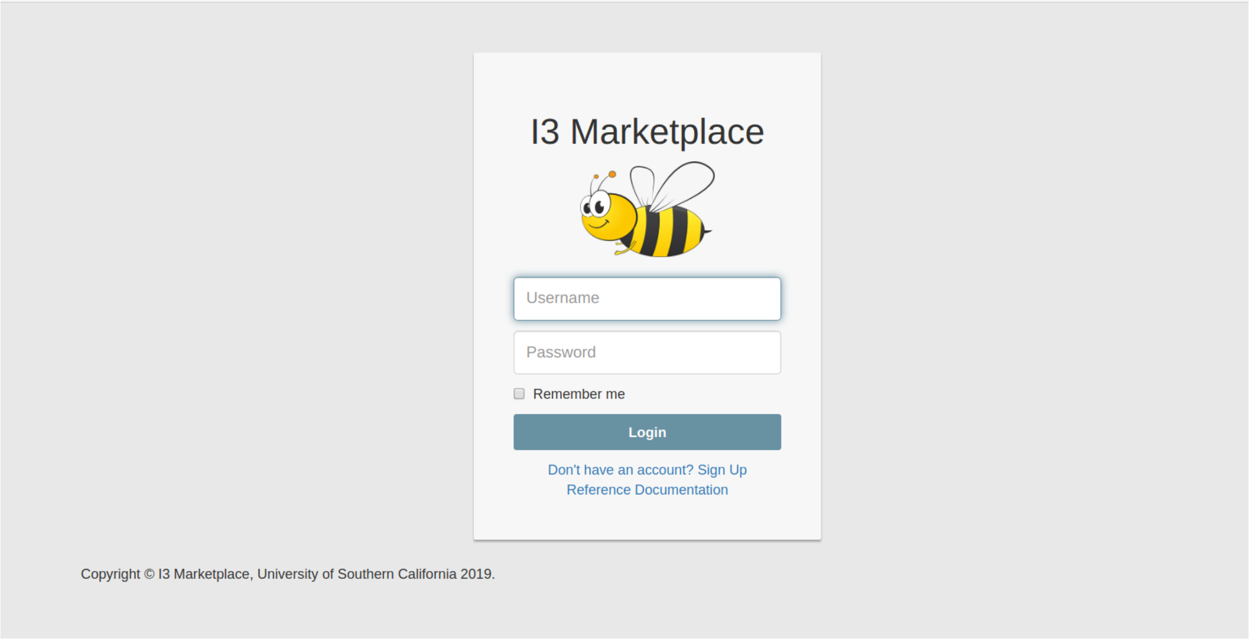
PUBLIC KEY—”), and copy it to i3. Sellers have no credentials for themselves (unless they buy products and act as buyers), but they create credentials for each of their registered devices (which is the password or public key he enters).

CHAPTER 6

Buyer's Guide

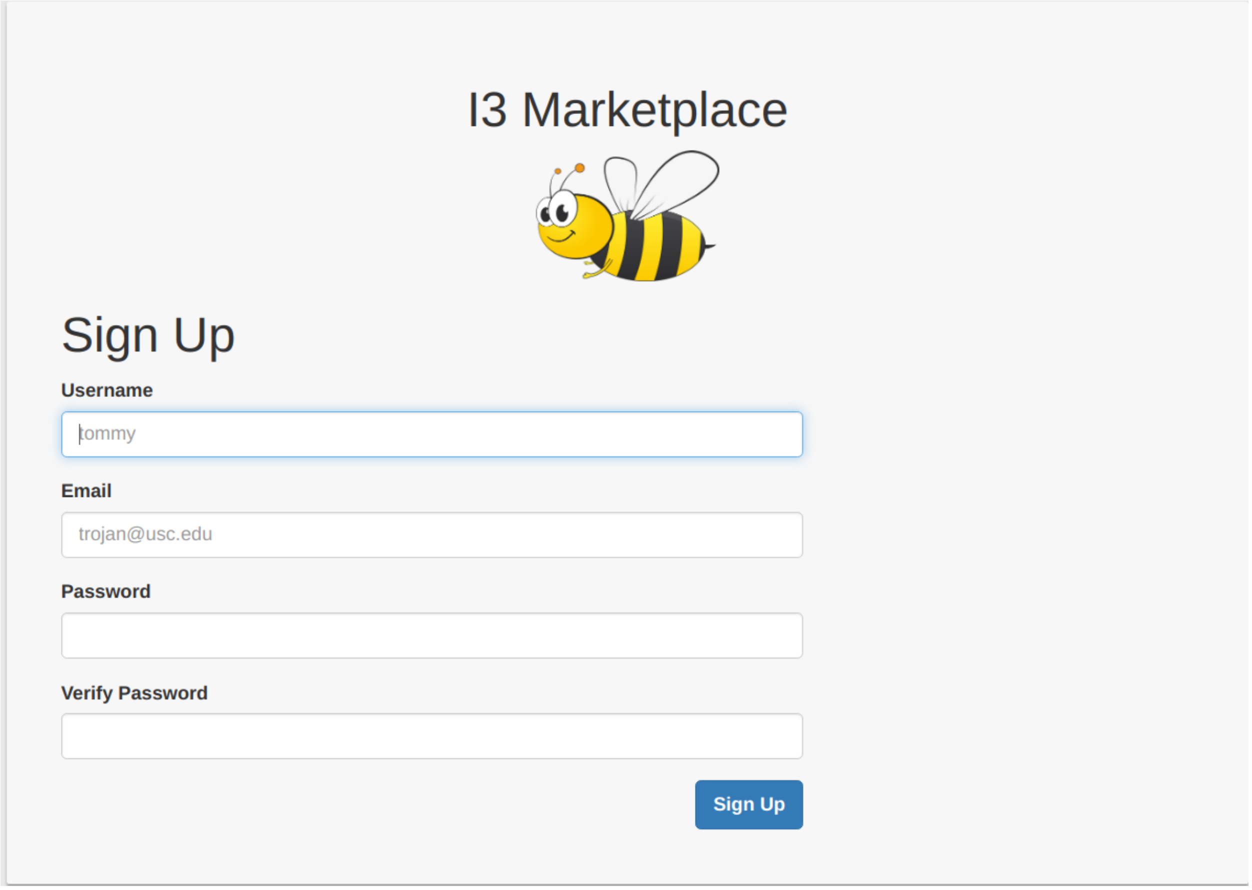
6.1 Signup

All URLs of this platform require login, if you're not logged in, you'll be redirected to the login page. Sign up first before you log in. As indicted below, for now we haven't enabled https, so please don't use a password that's the same as your other passwords.



Copyright © I3 Marketplace, University of Southern California 2019.

The new users have to create an account by clicking "Don't have an account? Sign Up". A new user has to enter an username, email address, and password to set up an account.



I3 Marketplace

Sign Up

Username

tommy

Email

trojan@usc.edu

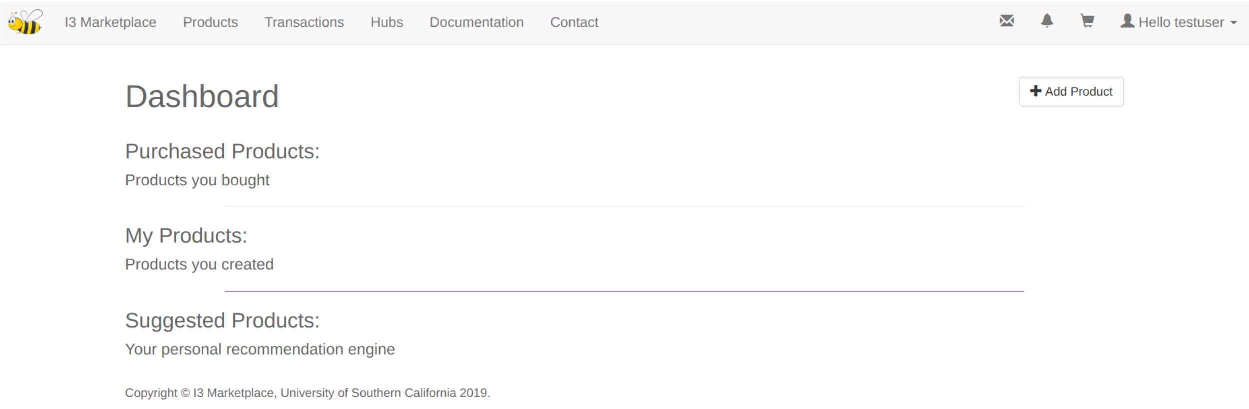
Password

Verify Password

6.2 Log in

After logging in, you'll see a dashboard page, which consists of three parts – purchased products, your products and suggested products. As is suggested by the hints, the "purchased products" are the products that you (as a buyer) bought, the "my products" are products that you (as a seller) created, and the "suggested products" are the products recommended for you according to your preferences (as of today, we don't have it yet, and it just shows all products on the marketplace, except yours).

Apart from the above, the "add product" button on the up-right corner provides you with a quick way to create a product, but note that all products have to be created under existing hubs. So if you don't have a hub yet, please register a hub first.



Dashboard

Purchased Products:

Products you bought

My Products:

Products you created

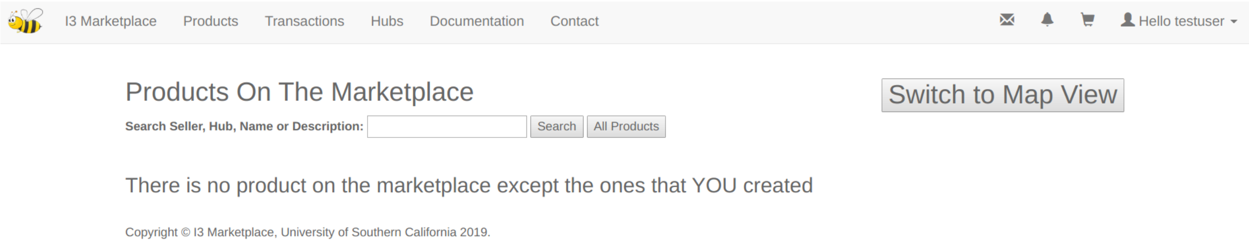
Suggested Products:

Your personal recommendation engine

Copyright © I3 Marketplace, University of Southern California 2019.

6.3 Products

The second field (up on the panel, right next to "I3 marketplace") is called "products". This page shows all the products on the marketplace, except the ones that you created. You can explore the market and buy products from here.



Products On The Marketplace

Search Seller, Hub, Name or Description:

There is no product on the marketplace except the ones that YOU created

Copyright © I3 Marketplace, University of Southern California 2019.

6.4 Product Detail

If you want to view the details of a product, just click the blue link. On this page you'll see the full name, product creator and other information. Products are restricted or unrestricted (as shown in above figures). For restricted products, you have to require access to the product. Only after the product creator approves your purchase will you be able to subscribe to it. Of course he can deny it. You'll get a notification when the seller makes a decision. For unrestricted products, you can just add to cart and check out. Another attribute of products is “sensor type”. Three choices are “sensor”, “actuator” and “both”. As a buyer, if you buy a sensor topic, you get a subscribe permission; if you buy a non-sensor topic, you get a publish-subscribe permission.

6.5 Subscribing Credential and Restricted Products

You’ll get your (personal, not specific to any product) 12-digit subscribing credential once you buy your first product. You will use it for ALL subscription requests (of course you still have to buy a product before you can subscribe to it). Below is what happens when you buy a restricted product (and if it’s the very first product you bought on the marketplace).

Now let’s log in as Matt and approve the purchase.

6.6 Map view

You can view products in both list and map (google map API key).

On any of the marks you can check out the details of a product.

6.7 REST API

All API endpoints require token authentication (auto generate when you click API key on up-right drop-down panel). Currently we only allow get query for products.

```
curl -X GET -H 'Authorization: Token {YOUR_API_KEY}' 'http://{HOST_IP}:{PORT}/api␣
↪product/?{QUERY_NAME}={VALUE}&{QUERY_NAME}={VALUE}&...'
```

The above query parameters are using AND logic.

6.7.1 Allowed Query Params:

- pk : return product with pk
- name : return product with name

1. seller : get a list of products this guy sells
2. sensor_type : sensor, actuator or both
3. restricted_active : 0 for unrestricted products, 1 for restricted
4. hub : filter by hub name
5. description_contains : return products whose description contain your input query (such as air quality). NOTE: use %20 if you want to input space, eg air%20quality

1. search : return any product whose SELLER or HUB or NAME or DESCRIPTION contain your input search query

6.8 Rating

In the product detail page there's a rating function as shown above. GET operation only requires login, while POST operation comes with mutiple permission check: * if request user is not logged in, return fail * if request user is creator of request product, return fail (You can't rate your own product) * if request user already rated the product, return fail (even if he bought it more than once) * everybody (except the seller) can rate the product (once and for all, only ONCE) * product rating have two values: Overall Rating (average by all raters) and Verified Rating (average by buyers). While now at my instance, the overall rating is displayed, we can switch to the mode that only display the average rating of those who actually bought the product. So even though everybody can rate, we can still keep track of the actual score (currently not fully supported though).

CHAPTER 7

Sending and Receiving Data Using MQTT

The other "invisible" user interface is the MQTT publish-subscribe broker, equipped with an authentication-authorization plugin, along with a JSON Web Token signature verification function (built with pyjwt and c-python connector inside the plugin). The 1883 and 8883 (MQTT+TLS, currently not opened yet) ports are exposed for publishers and subscribers. MQTT connect requests mainly consist of five parts: username, password, topic, socket, pub, or sub.

7.1 Data Publishing Guidelines for Sellers

The sellers create data products in the marketplace. After creating the product, they are required to publish the data to the marketplace MQTT broker. Whenever a buyer purchases data on the marketplace, the MQTT broker forwards the data to the buyer's device. Each publisher is required to run a publisher script to post the data to the data marketplace. In the publisher script, the seller needs to include product information such as name, credential, and permission.

MQTT username and topic name: Provide the FULL device name in the connection request, also the FULL topic name.

In MQTT connect request, the sellers are required to provide their device password if this device is registered using the API key. If your device is registered by an asymmetric key pair and you've submitted your public key to I3, generate a JSON Web Token claim and sign it with your private key (paired with the public key you provided to I3). Signature has to be performed using RS256, and right now only supports limited jwt fields. Specify an expiry time for your token, after which it will be invalid for authentication. But no auto disconnection is performed when your token expires (you can't use this token to connect again). Below is a python example to sign a token using a private key file and print it to the command line (valid for 1 minute). For now, please only use 'iss', 'iat', 'exp' fields. Directly use the token as the password for authentication.

NOTE, please don't use your private key string as the password!

7.2 Data Subscribing Guidelines for Buyers

The buyers purchase data products from the marketplace. When a buyer buys a product, he or she will receive credentials to subscribe to the data. Note that the MQTT broker forwards the data to the buyer's device after he has paid for the product, although the current version of the data marketplace does not provide support for real payments.

Each subscriber is required to run a subscriber script to access the data from the data marketplace. In the subscriber script, the buyer needs to include product information such as name and credential.