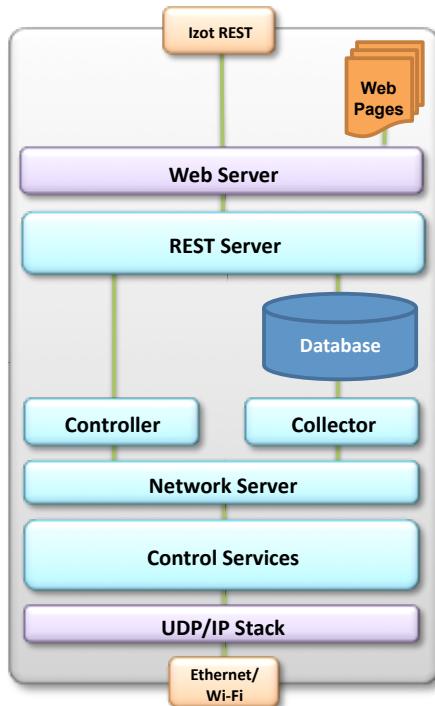




IzoT™ Server Stack

Ideal for developers creating Web applications for the Industrial Internet of Things.



Easy Web Applications

The IzoT Server Stack enables developers to build a Web application server for IzoT networks that runs on a 32-bit processor and operating system. Servers built with the IzoT Server Stack connect IzoT devices to Web clients, providing users with local Web access to monitor and control the operation of communities of devices. The IzoT Server Stack provides an easy-to-use RESTful API with a low-overhead but powerful interface for discovering devices, monitoring their data, and controlling them via network inputs.

A reference implementation of the IzoT Server Stack is available for the popular Raspberry Pi.

Device Discovery

The IzoT Platform allows developers to create communities of widely varying devices. The IzoT Server Stack automatically discovers IzoT devices in the local network using the IzoT Interoperable Self Installation (ISI) services built into the IzoT Device Stack. The IzoT Server Stack provides REST services, allowing any Web page or application to obtain a list of the devices discovered by the server; to determine the data those devices publish, and, finally, to determine acceptable inputs for each device in order to control their operations.

For example, a Web page can issue the following request to gather a list of all the devices in an IzoT network, including properties that identify the device types and interfaces: **GET /api/devices**.

Datapoint Access

The IzoT Platform supports communication among the devices using a publish-subscribe data model. IzoT-enabled devices can have both output and input datapoints. Output datapoints are used to publish data sensed or generated by a device. Input datapoints are used to receive data sensed or generated by other IzoT-enabled devices in a given network. The IzoT Server Stack discovers all the datapoints on the devices in its community. Once discovered, the IzoT Server Stack monitors the datapoints and maintains a database with the current values of all visible datapoints in the network.

Web clients can use the IzoT REST API to get the current value of any datapoints in the IzoT Server Stack database. The IzoT REST API also allows Web clients to update input datapoints within the community. Using the IzoT REST API datapoint services, users can easily create Web pages to display data received from IzoT devices — or even to control the devices.

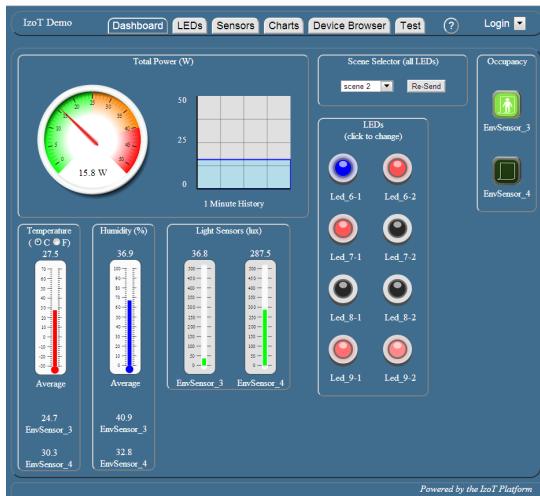
For example, a Web page can issue the following request to read the value of the datapoint with **GET /api/datapoints/5/**. The following request is an example of changing the color of an RGB LED to blue from a Web page:

```
PUT /api/datapoints/5/ {"value": "35"}
```

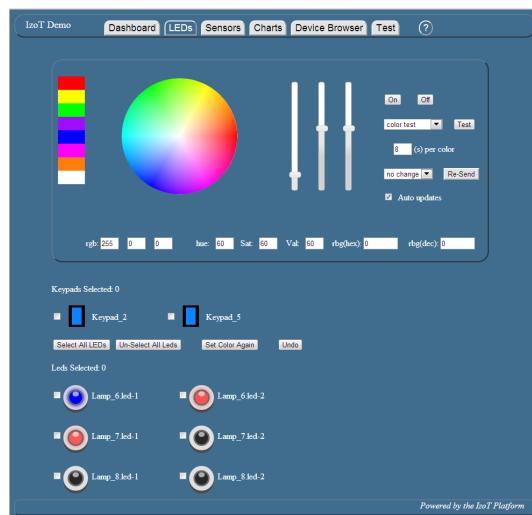
Example IzoT Web Pages

A set of example Web pages are included with the IzoT Server Stack. The pages are organized as tabs to make it easy to navigate between the example pages. You can use these Web pages to try out the IzoT Server, and you can also use these pages as a starting point for designing your own custom Web pages.

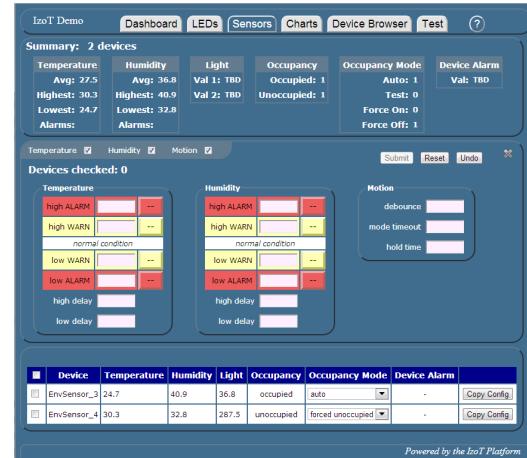
The main example Web page is the example Dashboard page. The Dashboard is a dynamic page that displays summary data and provides high-level control of the three example applications included with the IzoT Device Stack. These example applications are: an LED Controller, a Touch Keypad, and an Environment Sensor. As devices running the example applications are discovered by the IzoT Server Stack, they are automatically added to the Dashboard page.



The LED page demonstrates fine-grained control of IzoT devices. The page can show the status of a selected individual LED lamp, or of multiple selected LED lamps from the LED list on the page and then select a color or scene for all the selected lamps.



The Sensors page demonstrates configuration of IzoT devices. The page can display the configuration of a selected individual Environment Sensor, or multiple selected Environmental Sensors. Users can change the configuration for all the selected sensors simultaneously.



The Charts page demonstrates trending sensor data from IzoT devices. You can select datastreams from temperature, humidity, light-level, and occupancy sensors on Environment Sensor devices. You can also select datastreams from power and energy monitors on LED Controller devices. Using the Charts page, you can identify trends in your monitored devices such as increased power usage at different points in time.



The Device Browser is a general purpose Web page that displays the current data from a selected IzoT device. You can also use the Device Browser to update input datapoints on IzoT devices. The browser provides a valuable tool for testing IzoT devices as you develop them. The Device Browser adapts to discovered devices so you do not have to modify the Device Browser implementation to support a new type of IzoT device.

Type: EnvSensor			
Name	Value	Timestamp (localtime)	URL
humidity-highLimit2Enable	0	2013-10-23T17:34:57.517Z	http://10.2.124.55/api/datapoints/9/
humidity-highLimit1Enable	0	2013-10-23T17:34:57.655Z	http://10.2.124.55/api/datapoints/10/
humidity-lowLimit2Enable	0	2013-10-23T17:34:57.790Z	http://10.2.124.55/api/datapoints/11/
occupancy-mode	auto	2013-10-23T17:34:57.929Z	http://10.2.124.55/api/datapoints/12/
humidity	40.900002	2013-10-23T17:34:58.064Z	http://10.2.124.55/api/datapoints/13/
occupancyIn	0	2013-10-23T17:34:58.205Z	http://10.2.124.55/api/datapoints/14/
humidity-highLimitDelay	0	2013-10-23T17:34:58.341Z	http://10.2.124.55/api/datapoints/15/
humidity-highLimit2	0	2013-10-23T17:34:58.503Z	http://10.2.124.55/api/datapoints/16/
temperature-status	0	2013-10-23T17:34:58.735Z	http://10.2.124.55/api/datapoints/17/
temperature-highLimit2	0	2013-10-23T17:34:59.015Z	http://10.2.124.55/api/datapoints/18/
humidity-lowLimit2	0	2013-10-23T17:34:59.275Z	http://10.2.124.55/api/datapoints/19/
humidity-lowLimit1	0	2013-10-23T17:34:59.530Z	http://10.2.124.55/api/datapoints/20/

Powered by the IzoT Platform

On the Test page, you can enter requests to the IzoT Server and view the server's responses. The Test page is a valuable tool for learning the IzoT REST API, and is also valuable for testing purposes when you are building and testing your custom IzoT applications and networks.

The screenshot shows the IzoT Demo Test page interface. At the top, there are two input fields: 'Get Request' containing '/api/devices/' and 'Put Request' with 'Url: http://api/devices/'. Below these are buttons for 'Clear' and 'Clear Log each request'. A large text area displays the log of requests and responses:

```

Get Request:/api/devices/
Get Response:
id: 1
url: http://10.2.124.55/api/devices/1/
name: cpu
brand: Intel
type: cpu
devId: 1
uid: 1
put:
categories:
notes:
status: pending
timestamp: 2013-10-23T17:33:48.664Z
datapoints: http://10.2.124.55/api/devices/1/datapoints/
id: 2
url: http://10.2.124.55/api/devices/2/
name: Keypad
brand: Echelon
type: Keypad
devId: 2
uid: FEE5A98C5C14
putId: SFFFFF0300700001
categories:
  
```

Powered by the IzoT Platform

Downloading the IzoT Server Stack

A beta release IzoT Server Stack is available as a free download from <http://iiot.echelon.com/get-started>. The download includes ready-to-run software for the Raspberry Pi, as well as full source code for the IzoT Server Stack. You can either modify the current source code for the Raspberry Pi, or port it to other compatible processors and operating systems. The production 1.0 software will be available in Q1 2014, in two versions: free (without support) and premium (with support and expanded rights).

Specifications

Reference Implementation Target Platform

Raspberry Pi Model B with 512 MB RAM and Raspbian Linux

The source code may be ported to other compatible 32-bit processors running Linux, Microsoft Windows, or other operating systems with POSIX services

Reference Implementation Target Flash Memory

8 GB minimum SD flash card

Web Client Interface

REST API supporting XML, JSON, and text data encoding

Compatible Web Browsers

HTML5 compatible Web browsers including Internet Explorer 9 on Windows 7 (64-bit and 32-bit), Internet Explorer 10 on Windows 8 (64-bit), Chrome on Windows 7 (64-bit and 32-bit), Firefox on Windows 7 (64-bit and 32-bit), and Safari 5.1 on iPad Mini with iOS

Reference Implementation Integrated Components

Django Web application framework; Nginx Web server and reverse proxy server; Gunicorn Python Web server gateway interface; SQLite relational database management system