

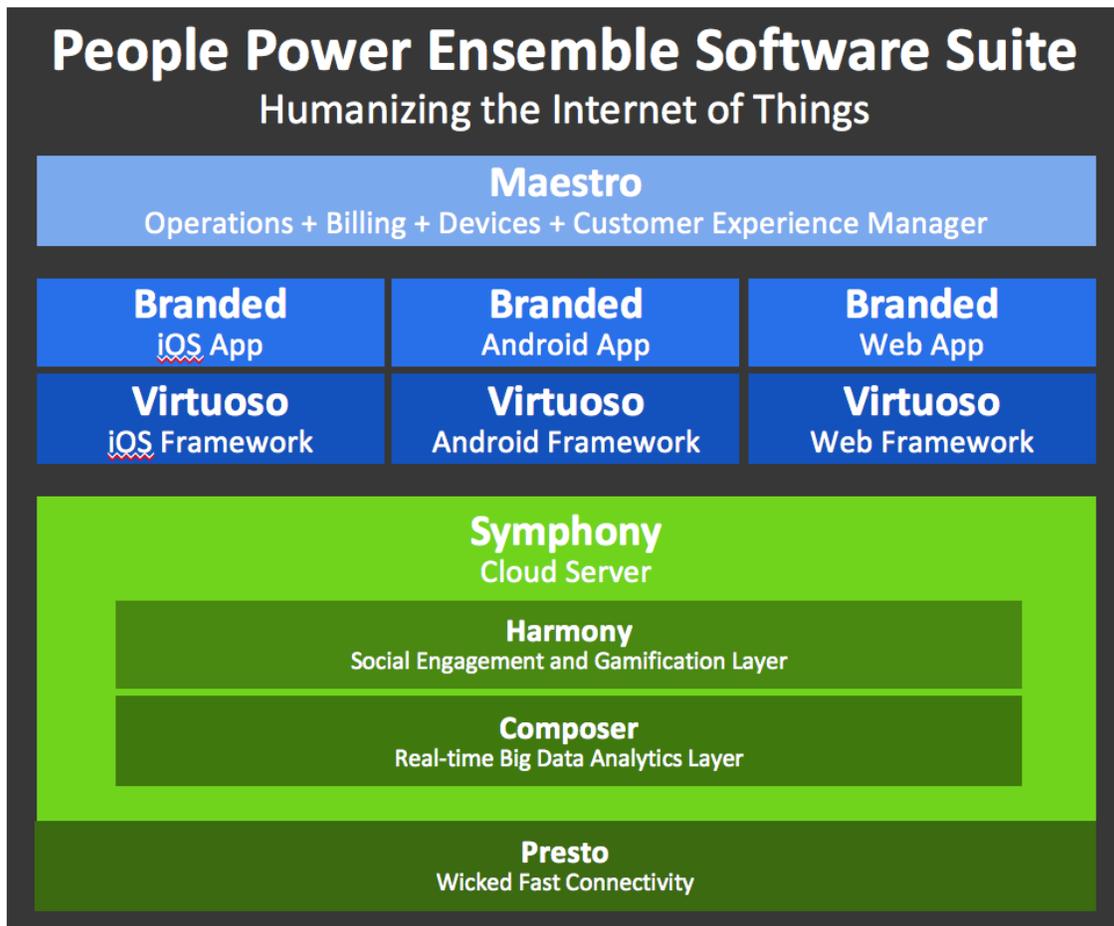


# Presto

## Intro & Getting Started

Presto is one of the fastest and most open connectivity layers available for the Internet of Things. Presto is also free for manufacturers to Internet-enable any device, and with open APIs and an open developer console, it's extremely easy to get connected. Presto also connects to other clouds through cloud-to-cloud integrations and OAuth 2.0, and can even connect with devices that are already shipping in market by teaching Presto how to speak that device's language.

Once connected, manufacturers can take advantage of the upper layers in the Internet of Things stack, including Composer, Harmony, Virtuoso, and Maestro.



**Presto by People Power**

Intro & Getting Started

Question? We're here. [support@peoplepowerco.com](mailto:support@peoplepowerco.com)

2015.12.22



## Presto's Key Features

Presto enables free connectivity with up to 2048-bit SSL connectivity, optional bi-directional authentication, up to 12 measurements and status updates per hour, and near-instantaneous command delivery across unlimited devices. Contact us if you need more.

### **Free Connectivity:**

- Cloud-enable any product or device for remote control anywhere in the world.
- HTTP GET and POST to send measurements and receive commands.
- Both JSON and XML are supported, it's the developer's personal choice.
- Bi-directional data streams with very low bandwidth, usually less than 50 bytes per minute.
- NAT and firewall penetration.
- Synchronization in 0.25 – 0.50 second latency.
- Online and Offline recognition of devices and people.
- Automatic data correction and filters.
- Vector measurements with Hot, Medium, and Cold bucket data, for maximum database efficiency.
- 2048-bit SSL encryption and bi-directional authentication.
- IPv4 and IPv6 support.
- Manage dropped connections and reliably deliver missed messages.
- Marshal message delivery to low capability devices, lowering device cost.
- Broadcast and unicast messaging, sharing a single pipe.

### **Additional supported forms of connectivity and storage:**

- WebSockets support instantaneous delivery of data and control.
- RTMPS and RTSP streaming protocols to connect IP cameras of all types.
- Capture, encrypt, and store large files and media.

## What's exposed?

The Presto developer console exposes a subset of the overall Ensemble software suite. Specifically:

- Developers are fully armed to connect any thing to the cloud services, including access to the easiest out-of-box experience for Internet-connected gadgets. Presto supports both IPv4 and IPv6 connections.

**Presto by People Power**

Intro & Getting Started

Question? We're here. [support@peoplepowerco.com](mailto:support@peoplepowerco.com)

2015.12.22



- The Presto SDK is open source under the Apache 2.0 license, enabling manufacturers to cloud-connect a new product in minutes.
- Developers have access to data filtering and correction analytics, especially for energy measurement related devices.
- Presto features completely open APIs, including open Device APIs and Application APIs.
- You have the power to create your own mobile apps for your devices using the open Application APIs, or contact People Power to rebrand and customize the Virtuoso mobile app framework.
- Rules enable users to create if-then natural language sentences that describe triggers, states, and actions. *Please contact support@peoplepowerco.com if you would like assistance in demonstrating rules for your device.*

Presto API's are fully open, enabling interoperability and preventing vendor lock-in. The main API's are RESTful (HTTPS-based), but also support WebSockets for high-speed app-to-device communications, as well as RTMPS and RTSP for video streaming (which isn't free).

- The Presto Device API at <http://docs.iotdevices.apiary.io> describes how to securely connect devices to the Ensemble software suite, with both XML and JSON support.
- The Ensemble Application APIs at <http://docs.iotapps.apiary.io> describes how to access the rest of the Ensemble software suite at an API level.

Finally, there are Presto SDK's and UI templates available under an Apache 2.0 license on GitHub at <http://github.com/peoplepower/presto-sdk-c>. The SDK's are heavily used on Linux-based platforms to connect many different types of software and hardware protocols to the Presto Device API.

**Presto by People Power**

Intro & Getting Started

Question? We're here. [support@peoplepowerco.com](mailto:support@peoplepowerco.com)

2015.12.22

## 1. Create a Type of Thing – "Device Types"



Each connected device instance has a Device Type. A unique number identifies each Device Type. The Device Type enables Presto to understand and optimize itself for the capabilities of the connected device.

Each Device Type binds a device instance with a unique UI, icon, optional installation guide, optional FAQ, optional help guide, a set of rules, and analytics.

You create your own Device Types in "Types of Things". No other developer can see your Types of Things, until you are ready to go into production.

When you create a new Type of Thing, the developer console asks some questions about the operation of your device. The questions are designed to optimize Presto for maximum scalability for your device. If the thing you're connecting will not use some functionality of Presto, then Presto will turn off that functionality and lower the overall cost for us to host your product.

## 2. Define your Parameters



Parameters are the individual measurements and commands that are passed back and forth between a device and Presto, and apps. Parameters are basically key/value pairs, on steroids.

Parameters are a little more complex than just variable names, and for good reason. Presto is designed to maximize scalability and lower the overall cost of hosting in a production environment, by understanding the data storage and management requirements that are actually needed for your things. We don't mess around.

Because Presto uses a single namespace for parameters, developers are forced to add a prefix to parameter names. We recommend using your initials or your company's initials as a prefix. For example, People Power Company uses the prefix "ppc" in front of parameter names, to create "ppc.parameterName".

When you create a new parameter, the Presto developer console will let you specify its expected behavior. We call anything that your thing sends to the Presto server a "measurement". This could include current state information, or actual sensor measurements. And anything that the Presto server sends to a thing is called a "command".

Measurements can be numeric or string. If it's numeric, then Presto can optimize the



storage even more. Numbers with fewer decimal places are better optimized on Presto.

Presto offers three levels of storage for measurements:

- **Don't store history** – This is used when the parameter represents the current state of the device. This is the most optimized selection, because no values except the most recent value are stored to the database.
- **Update the history each hour** – Presto optimizes database scalability with a "vector measurement" technique. This means Presto avoids logging redundant data points in the database. The history in the database will actually change whenever the parameter's value changes, or at least once per hour, whichever happens first.
- **Update the history always** – Vector measurements are turned off, and Presto will store in the database every single value that is sent from the device for this parameter.

A screenshot of a web browser window showing the "Parameter" configuration page. The browser address bar shows "https://fabrux.peoplepowerco.com/parameters/parameter.html". The page title is "Parameter". The form contains several sections: "Name" with a text input field containing "ppc.parameterName" and a help icon; "Description" with a text input field containing "Showing how to create a parameter" and an "(optional)" label; "System Units" with a dropdown menu set to "Money"; "Used as a command?" with radio buttons for "Yes" (selected) and "No"; "Numeric values only?" with radio buttons for "Yes" (selected) and "No"; "System Multiplier" with a dropdown menu set to "1-Unit Multiplier (implied)/Unit Multiplier"; "Decimal place storage" with a dropdown menu set to "0 digits" and an example "Ex: 3 = 0.000"; "Used as a measurement?" with radio buttons for "Yes" (selected) and "No"; "How to store values?" with radio buttons for "Don't store history" (selected), "Update the history each hour (\$)", and "Save everything, always (\$\$)"; and "Apply automatic filtering and correction?" with radio buttons for "Yes" (selected) and "No".

**Presto by People Power**

Intro & Getting Started

Question? We're here. [support@peoplepowerco.com](mailto:support@peoplepowerco.com)

2015.12.22



Presto can also apply data correction analytics to your parameter. This is normally used in production. Contact People Power at [support@peoplepowerco.com](mailto:support@peoplepowerco.com) if you're interested in using this option, so we can help you connect the right type of data correction analytic.

Parameters can also have index numbers, like an array. This is great for devices that produce multiple measurements or commands for the same parameter. Think about a power strip with multiple sockets, each one accepting a similar on/off command, and each one producing measurements of power and energy. To enable index numbers, your Device Type's attributes will have to be configured to specify the individual index numbers or letters you wish to use.

### 3. Connect Things



Once you've defined a Type of Thing and some Parameters that your device will communicate, it's time to create an instance of that Thing in your account.

Each device has an ID. Device ID's must be globally unique, think of them like a mailing address to communicate with a device. MAC addresses make great device ID's. But if you check out the "**QR-code ready Device ID's**" document, you'll see a format we recommend that guarantees uniqueness and enables your devices to have a fantastic out-of-box experience with a QR-code scan from a mobile app.

**Presto by People Power**

Intro & Getting Started

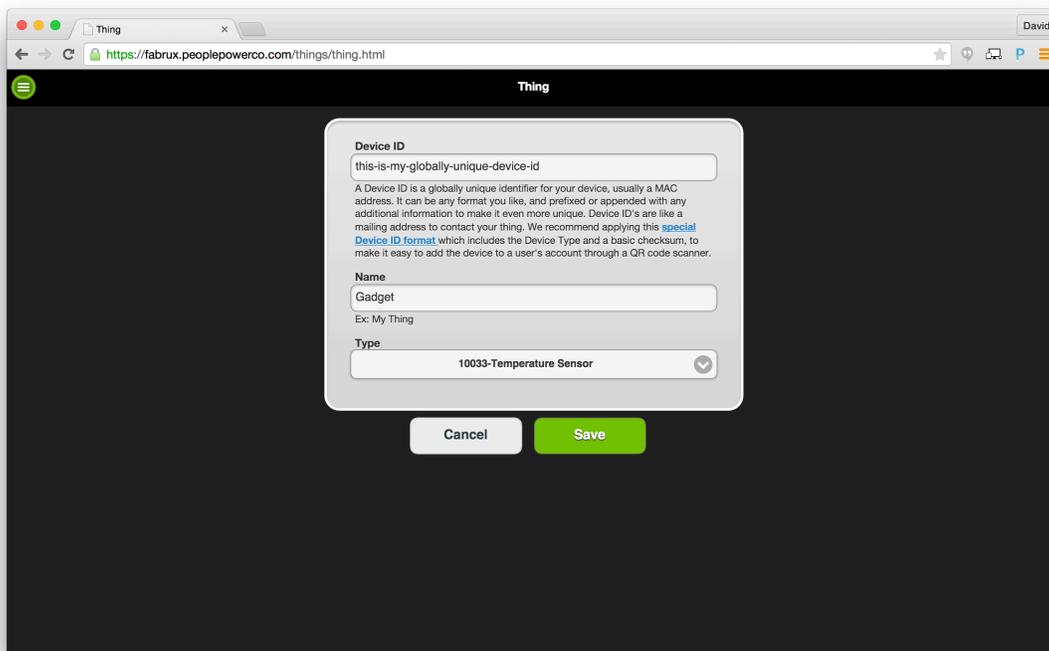
Question? We're here. [support@peoplepowerco.com](mailto:support@peoplepowerco.com)

2015.12.22



The first Thing you add to your account will always be through the Application API (which means you have to log in to use it). If your Thing is actually a gateway or a proxy, it may then use the Presto Device API to register more Things to your account.

If you register a Thing to your account through the developer console, you'll be asked to specify a globally unique Device ID (sometimes we just come up with a unique name when doing testing, like "moss-thingamajig"). You can give your Thing a nickname, and you must specify what Type of Thing it is.



Once your Thing is registered to your users account, you must use the **Device API** to send measurements to Presto, and receive commands from Presto. Check out the Device API documentation at <http://docs.iotdevices.apiary.io> for all the details on the API.

You can use the Presto developer console, or Presence mobile apps in Developer Mode, or even the raw Application API at <http://docs.iotapps.apiary.io> to pull the measurements from your Things back out of the Presto server.

**Presto by People Power**  
Intro & Getting Started

Question? We're here. [support@peoplepowerco.com](mailto:support@peoplepowerco.com)

2015.12.22



## 4. Explore

Check out the other Presto documentation to learn how to do other things with Presto, like create QR-code ready device ID's.

Go into the Presto developer console Settings > FAQ to get answers to common questions.

Ask your own questions and get feedback from Presto developer console Settings > Feedback. Our support team is there to help.

(By the way, the Presto Application API enables your own apps and devices to receive qualified and quantified feedback from end users to help drive your roadmap.)

## 5. Launch!

With Presto, your devices will be ready for production usage. As long as you stay within the bounds of 12 measurements per hour (which is a lot more than most production-grade devices), there are no fees for connectivity! No data usage fees. No per-device fees. No kidding. Of course, please let us know if you're going into production so we can double check your settings and make sure it's ready to scale.

Why did we make connectivity free? Because we believe the value of the Internet of Things is not in connectivity, but higher in the IoT stack. We can help.

Users need apps with your brand. Building quality apps for iOS, Android, and Web is very time-consuming and beyond the timeframe or expertise of most companies. People Power offers the Virtuoso Mobile App Framework to accelerate your time-to-market with a fully branded and customizable app.

People Power has launched full smart home and building services, and found very quickly that connectivity and mobile apps are only half the battle. To deliver a quality service, service providers and manufacturers need to be able to manage customers, perform tier-1 technical support, and engage with their users to create a sticky offering. People Power has created the Maestro Command Center to do just that. It's an Operational Support System (OSS), a Business Support System (BSS), and a Customer Experience Manager (CEM) for the Internet of Things.

If your devices are connected to Presto, they're already compatible with Maestro and the Virtuoso mobile app framework. Contact us at [sales@peoplepowerco.com](mailto:sales@peoplepowerco.com) and we can accelerate you ahead of competitors, and arm you to be successful in the IoT market.

**Presto by People Power**

Intro & Getting Started

Question? We're here. [support@peoplepowerco.com](mailto:support@peoplepowerco.com)

2015.12.22