Hestia Marketplace Architecture and Design

Andrew Lalis

Last Updated: May 16, 2018

Contents

1 Introduction and Background

The Hestia Web App is a method by which users can interact with their local Hestia *controllers*, to manipulate certain *devices* in their home. In order for the controller to be able to send the proper commands to such devices, it is imperative that the user install the device's required *plugin*.

A plugin is a collection of python scripts and configuration files that enables the controller to interact with any IP-enabled device on its local network. To make it easier for users to ensure their devices work without hassle, the *Plugin Marketplace* will provide access to a database of plugins. The marketplace will provide a REST API to let the Web App query plugins and to let plugin developers upload their work.

From the developer's perspective, a plugin is much more than simply a file folder, and as such, information such as the date of creation, author, and a description need to also be stored so that users can easily understand the contents and purpose of a plugin without needing to dissect any source code.

2 Architecture and Database Design

The main focus of this section will be on the tables needed to store all the information relevant to a plugin. This will consist of a list of attributes that describe a plugin, and any needed auxiliary tables for storing other information in a way that wastes as little space as possible.

As mentioned before, the most important table in the database is the *Plugins* table, whose attributes are defined in detail in the following list.

- *id* - A globally unique identification string which can be used to reference this plugin.

- *name* - The name of the plugin, as it will appear in lists when users wish to browse the list of plugins.

- *author_id* - The unique authenticated id of the user who uploaded the plugin.

- *created_date* - The date on which the plugin was first uploaded.

- *last_edited_date* - The date of the most recent update to the plugin.

- *version* - A string of numbers representing the current version of the plugin. This is up to the author's discretion.

- *description_short* - A short and concise description, less than 100 characters, which can be displayed in menus to give users a hint as to what the plugin does.

- *description_long* - A much more detailed description of the plugin, which provides detailed instructions for use of the plugin, or other relevant details that the author thinks the users should know.

- *up_goats* - Up-Goats represent the number of users who cast a positive vote for the plugin, meaning that they believe it is valuable and useful.

- *down_goats* - Down-Goats are the opposite of Up-Goats, and count the number of users who voted negatively for the plugin.

Additionally, some other tables are needed to provide auxiliary information to make the user experience more enjoyable. These are defined below.

The *Tags* table holds many tags, which are can be applied to many plugins to group them into related categories.

- *name* - A unique string which represents a tag, which can be applied to a plugin to give it some extra meaning.

- *description* - A short description of the tag, or an embellishment of its meaning.