openHAB Addon Suggestion Finder

Architecture Overview

The architecture comprises the following elements.

1. Foundation Work (`org.openhab.core.addon`)

- The AddonInfo DTO has been extended to include fields for 'discovery-methods' which identify the mechanisms by which an addon could be discovered. And sub- DTOs for 'discovery-methods' and 'match-properties' have been added.
- The XML schema for AddonInfo has been extended accordingly.
- The XML de-serializer for AddonInfo has been extended accordingly. With additional sub- DTO de-serializers for 'discovery-methods' and 'match-properties'.
- An additional DTO has been defined to contain a List<AddonInfo>.
- An additional XML de-serializer has been created for loading a List<AddonInfo> DTO.
- Junit tests added for the above.

2. Provide Addons Data (`org.openhab.misc.addonsuggestionfinder`) (may be renamed)

- This new addon has been created in openHAB-addons.
- During the Maven build process this addon is built last of all.
- During the build process Maven reads the addon.xml data from all other previously built addons, and assembles it into an addons.xml file that is built into this addon.
- In other words, this addon contains the addon.xml data from all addons that are included in the openHAB main distribution .kar file.
- The addon provides an OSGI service component that implements the AddonInfoProvider interface.
- When queried, the AddonInfoProvider provides a List<AddonInfo> containing all the respective AddonInfo's of all the addons in openHAB-addons.
- Note: the AddonInfo's include all addon.xml data such as 'connection' and 'countries' as well as 'discovery-methods', so it can be used for other purposes than discovery of suggested addons.
- Junit tests added for the above.
- Note: this provides information about karaf addons that are not yet loaded.
- (A similar addon that provides information about marketplace addons may be considered in future).

3. Scan the LAN and Provide Suggestions (`org.openhab.config.discovery.addon`)

- A new AddonSuggestionFinderService has been defined.
- AddonSuggestionFinderService reads the AddonInfo's from all loaded AddonInfoProviders. And if the above-mentioned 'org.openhab.misc.addonsuggestionfinder' is loaded it will get the data of all the addons included in the openHAB main distribution .kar file.
- AddonSuggestionFinderService scans the user's LAN based on the 'discovery-methods' of each available addon, and if it discovers a matching entity on the LAN it adds the respective addon to an internal list of 'suggested' addons.
- AddonSuggestionFinderService provides an end point method to read the full list of suggested addons.
- AddonSuggestionFinderService depends on OSGI components that implement the AddonSuggestionFinder interface. These components may scan the user's LAN using various technologies in order to discover available devices.

- Two AddonSuggestionFinder components are defined – namely MDNSAddonSuggestionFinder and UpnpAddonSuggestionFinder. Further AddonSuggestionFinder components could be forseen.
- The MDNSAddonSuggestionFinder component uses mDNS (via Jmdns) to discover devices on the LAN.
- The UpnpAddonSuggestionFinder component uses UPnP (via Jupnp) to discover devices on the LAN.
- Junit tests added for the above.
- Whenever a new AddonInfo provider component is loaded, the AddonSuggestionFinderService will start a new scan.
- Note: scan may take (say) 30 seconds depending on the technology of the respective AddonSuggestionFinder components.

4. REST API Endpoint (`org.openhab.core.io.rest.core`)

- The AddonResource REST API component has been extended to import (i.e. instantiate) an AddonSuggestionFinderService.
- The AddonResource REST API has been extended with a 'suggestions' endpoint, to which the UI (or user) can make an HTTP GET request. To which the response is the JSON payload of the AddonSuggestionFinderService's list of suggested addons.

5. User Interface (`org.openhab.ui`)

- On startup the UI shall automatically load the org.openhab.misc.addonsuggestionfinder addon described above. (This causes AddonSuggestionFinderService to start a new scan with the data therein).
- After about (say) 30 seconds the UI shall query the AddonResource REST API 'GET suggestions' endpoint to retrieve the data about all suggested addons.
- The UI shall then display the suggested addons, allow the user to select some or all of them, and then install the respective suggested addons.