

Using endpoint groups

- [Basic architecture](#)
- [Configuring Kaa](#)
 - [Profile filters](#)
 - [Custom endpoint groups](#)

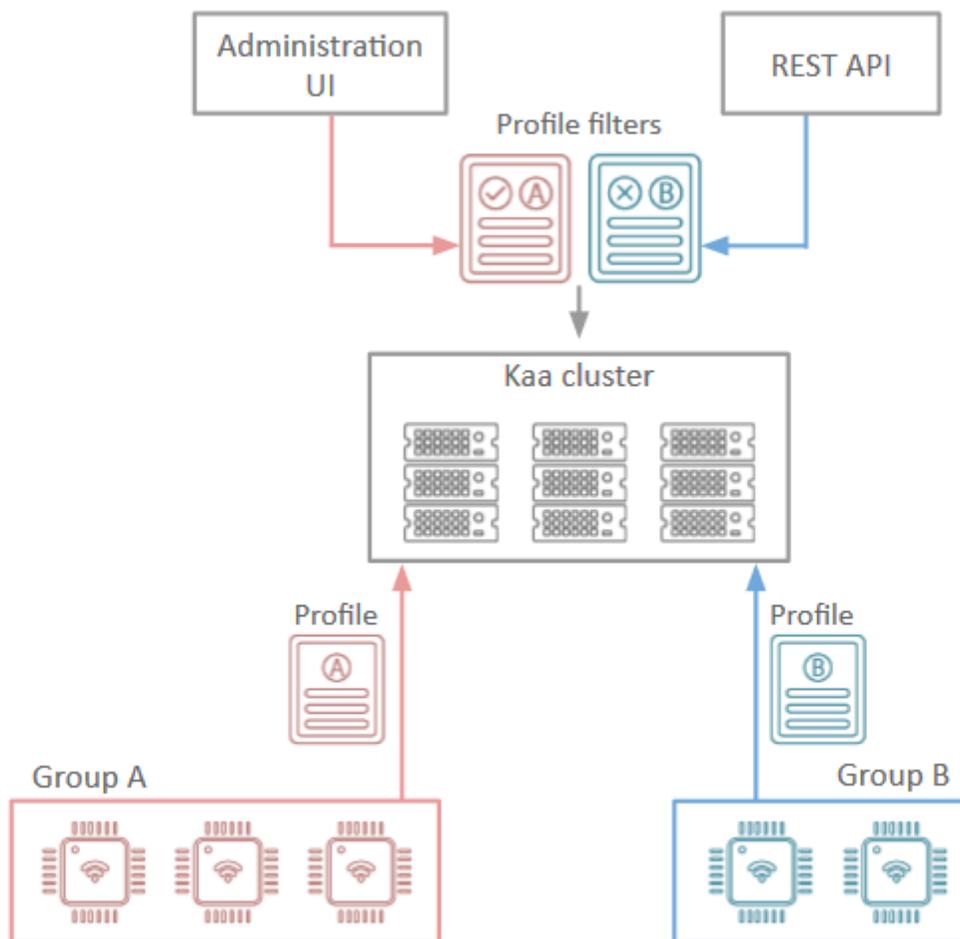
Kaa provides a mechanism for endpoints aggregation within the application that is based on groups. Grouping endpoints enables you to activate specific configuration parameters, control access to notification topics, etc. Please review the [Kaa grouping design reference](#) for more details.

This guide will familiarize you with the basic concepts of designing endpoint groups. It is assumed that you have either set up a [Kaa Sandbox](#), or a fully-blown [Kaa cluster](#) already and that you have created a [tenant](#) and an [application](#) in Kaa. It is also strongly recommended to review [collecting endpoint profiles guide](#) before you proceed with this one.

Basic architecture

The following diagram illustrates basic entities and data flows in scope of the endpoint profile management:

- Endpoint profiles are generated based on the [client-side](#) and [server-side](#) endpoint profile schemas and corresponding profile data.
- Endpoints send their client-side profiles to the Kaa server during the [registration](#).
- Server-side applications and scripts may use Kaa REST API in order to modify server-side profile of endpoints.
- The Kaa tenant developer specifies [profile filters](#) for endpoint groups using either [Admin UI](#) or [REST API](#)
- Kaa Operations service classifies endpoints into groups based on their profiles and the group profile filters



Configuring Kaa

Each Kaa application has a special, built-in, non-user-editable group "all" with weight 0. The associated profile filter is automatically set equal to "true" for each profile schema version in the system. Therefore, group "all" contains every endpoint registered in the application. You can create your custom endpoint groups using the [Admin UI](#) or [REST API](#).

Once created, an endpoint group does not contain any endpoints, so you will need to create and add custom profile filters to the group.

Profile filters

Each group can be associated with multiple profile filters, each specific to a separate client-side and server-side profile schema version combination. Only one profile filter can be defined for a profile schema version combination. However, you may also define profile filters that are agnostic to either client-side or server-side profile part. In this case, either client-side profile or server-side profile will not be accessible in the filter. This is useful in case you want to specify an endpoint group that is based on certain client-side profile property and is not affected by server-side profile updates and vice-versa.

We will use client-side profile schema defined in "[collecting endpoint profiles guide](#)" section and the following sample profiles:

Client-side Endpoint Profile A	Client-side Endpoint Profile B	Client-side Endpoint Profile C
<pre>{ "id": "device1", "os": "Android", "os_version": "2.2", "build": "2.0.1" }</pre>	<pre>{ "id": "device2", "os": "Android", "os_version": "4.0.1", "build": "3.0 RC1" }</pre>	<pre>{ "id": "device3", "os": "iOS", "os_version": "8", "build": "3.0 RC1" }</pre>
Server-side Endpoint Profile A	Server-side Endpoint Profile B	Server-side Endpoint Profile C
<pre>{ "subscriptionPlan": "Regular", "activationFlag": "true" }</pre>	<pre>{ "subscriptionPlan": "Regular", "activationFlag": "false" }</pre>	<pre>{ "subscriptionPlan": "Premium", "activationFlag": "true" }</pre>

Creating profile filters is possible through either [Admin UI](#) or [REST API](#).

Once a profile filter is created, you need to activate it. Filters that are not activated do not impact endpoint groups and do not affect the endpoints.

Custom endpoint groups

The table below demonstrates the use of profile filters and results of filtering for sample profiles.

Group Name	Filter	result for profile A	result for profile B	result for profile C
Android Froyo endpoints	#cp.os.toString().equals("Android") and #cp.os_version.toString().startsWith("2.2")	true	false	false
Android endpoints	#cp.os.toString().equals("Android")	true	true	false
iOS 8 endpoints	#cp.os.toString().equals("iOS") and #cp.os_version.toString().startsWith("8")	false	false	true
3.0 RC1 QA group endpoints	#cp.build.toString().equals("3.0 RC1")	false	true	true
Deactivated devices	#sp.activationFlag == false	false	true	false
iOS devices with premium subscription plan	#cp.os.toString().equals("iOS") and #sp.subscriptionPlan.toString().equals("Premium")	false	false	true

Thus, "Android Froyo endpoints" group will contain only device A; "Android endpoints" group will contain devices A and B; "iOS 8 endpoints" group will contain device C, and "3.0 RC1 QA group endpoints" will contain endpoints B and C.