Endpoint profiling

The endpoint profile is a structured data set of custom-defined complexity that describes specific characteristics of the endpoint. The endpoint profile is used to classify endpoints into endpoint groups and is comprised of the service and custom parts. The service part is identical across the applications and is used by Kaa internally for its functions. The custom part is defined by an application developer and is unique to the application.

The endpoint profile is initially generated at the stage of a new endpoint registration. Prior to that, the profile values should be specified by the client developer using the Kaa endpoint SDK. The endpoint profile can be also updated at run time either by the user or as a result of a change in the client's operating conditions. In both cases, the endpoint membership in the endpoint groups is re-evaluated and updated to match the new profile.

The endpoint profile is unidirectionally synchronized, and thus should **not** be considered as a means to temporarily store client data in the server. There is no way for the endpoint to retrieve the profile information back from the server. At its start, the client must fill in the current endpoint profile with the up-to-date data. It is the responsibility of the client code to generate or read the profile data. The endpoint SDK does not persist the profile information over the client reboots. However, it detects profile data changes and submits the new data to the server as a profile update.

Profile custom part

The custom part of the endpoint profile is defined by the developer and comprises two elements: the profile schema and the profile data.

> **NOTE:** for the purpose of simplicity, *profile data* is sometimes referred to as *profile* in this documentation.

Profile schema and profile data

The custom part of the endpoint profile is based on a specific Avro-based profile schema created by the Kaa developer. The profile schema defines the data structure which can be used across multiple endpoints, while the profile data represents the value of each field in this structure for the individual endpoint - the endpoint's unique characteristics.

The profile schema is defined by the Apache Avro schema format and supports all of Avro features: primitive types, complex types, arrays, maps, etc.

The profile schema is maintained within the corresponding application, with its own version that distinguishes it from the previous schemas. Multiple schema versions and corresponding endpoint profiles created upon those schemas can coexist within a single application.

Once created, the profile schema must be activated on the Kaa server (which does not require any system downtime). To update a client to a newer schema version, it is necessary to re-generate the endpoint SDK and recompile the client application.

The following code block provides a simple profile schema example.

```
{
    "namespace": "org.myproject",
    "type": "record",
    "name": "MyProfile",
    "fields": [
        { "name": "country", "type": "string" },
        { "name": "city", "type": "string" },
        { "name": "age", "type": "int" },
        {
            "name": "hobbies",
            "type": { "type": "array", "items": "string" }
        }
    ]
}
```

The following profile would be compatible with our schema example.

```
{
    "country": "US",
    "city": "San Francisco",
    "age": 32,
    "hobbies": [
        "skydiving",
        "hiking"
    ]
}
```

The schema structure from our example allows filtering the endpoints by the owner's country (for example, to show only US news), city (for example, to push weather notifications only for the specified cities), age (for example, to apply age restrictions), and hobbies (for example, to push football scores for those whose hobby list contains "football"). It is allowed to create complex filtering conditions by combining as many one-operand filtering conditions as needed.

Endpoint profile update

When the profile information changes as the result of the client operation or user's actions, it is the client implementation responsibility to update the profile via the endpoint SDK API calls. The endpoint SDK detects profile changes by comparing the new profile hash against the previously persisted one. Should there be a change, the endpoint profile management module passes it to the Operations server, which in turn updates the endpoint profile information in the database and revises the endpoint groups membership.

Profile service part

The service part of the endpoint profile is used by Kaa internally for its functions and contain the following information.

Property	Description
Endpoint Public Key	The public key used for security purposes to validate endpoint requests
Profile Hash	The profile custom part's hash used to validate all endpoint requests
Configuration Hash	The endpoint configuration's hash used to validate endpoint configuration requests
Version Info	The object which includes versions of the profile, configuration, and notification schemas, as well as versions of the event class families.
Server Key Hash	The hash of the server that was used during the last endpoint request