

Profile management

- [Profile creation](#)
- [Profile changing](#)
- [Profile deletion](#)
- [Obtaining profile information](#)

Profiles can be managed using [Websocket API](#) or [REST API](#).

Profile creation

A new profile can be created with /api/profile/create request

API	Request	Response	Response status
WS API	<pre>SEND destination: /app/api/profile/create content-length:159 { "requestId": "26f4895a-2001-46e0-af56-2a130544f1b7", "realm": "/api/profile/create", "payload": { { "name": "profile1", "rate": "30", "metrics": ["3", "10"], "rules": ["1"] } } }</pre>	<pre>MESSAGE destination: /user/service content-type: application/json; charset=UTF-8 subscription: sub-1 message-id: 3-10 content-length: 84 { "requestId": "26f4895a-2001-46e0-af56-2a130544f1b7", "status": 200, "reason": "SUCCESS" }</pre>	200 OK 400 Object not found 500 Persist exception
REST API	<pre>POST: /api/profile/create "application/json; charset=utf-8" { "name": "profile2", "rate": "30", "metrics": ["1", "2"], "rules": ["1"] }</pre>	<pre>{ "status": 200, "reason": "SUCCESS" }</pre>	200 OK 400 Object not found 500 Persist exception

Where

- name – profile name
- rate – metric acquisition rate
- metrics – metric identifiers list
- rules – rule identifiers list

Profile changing

A profile can be changed with /api/profile/update request

API	Request	Response	Response status
-----	---------	----------	-----------------

WS API	<pre> SEND destination:/app/api/profile/update content-length:166 { "requestId":"76d35097-7948-4e58-a7ae-f4d1led63e02", "realm":"/api/profile/update", "payload": { "id":"2", "name":"static", "rate":"30", "metrics":["3","10"], "rules":["1"] } } </pre>	<pre> MESSAGE destination:/user/service content-type:application/json; charset=UTF-8 subscription:sub-1 message-id:3-11 content-length:84 { "requestId":"76d35097-7948-4e58-a7ae-f4d1led63e02", "status":200, "reason":"SUCCESS" } </pre>	200 OK 400 Object not found 500 Persist exception
REST API	<pre> POST: /api/profile/update "application/json; charset=utf-8" { "id":"3", "name":"profile1", "rate":"30", "metrics":["3","10"], "rules":["1"] } </pre>	<pre> { "status":200, "reason":"SUCCESS" } </pre>	200 OK 400 Object not found 500 Persist exception

Where

- id – profile identifier
- name – profile name
- rate – metric acquisition rate
- metrics – metric identifiers list
- rules – rule identifiers list

Profile deletion

A profile can be deleted with /api/profile/delete request

API	Request	Response	Response status
WS API	<pre> SEND destination:/app/api/profile/delete content-length:103 { "requestId":"e2412d55-4256-4004-bb93-620af1905deb", "realm":"/api/profile/delete", "payload": { "id":"3" } } </pre>	<pre> MESSAGE destination:/user/service content-type:application/json; charset=UTF-8 subscription:sub-1 message-id:3-14 content-length:84 { "requestId":"e2412d55-4256-4004-bb93-620af1905deb", "status":200, "reason":"SUCCESS" } </pre>	200 OK 400 Object not found 500 Persist exception

REST API	<pre>POST: /api/profile/delete "application/json; charset=utf-8" { "id": "4" }</pre>	<pre>{ "status": 200, "reason": "SUCCESS" }</pre>	200 OK 400 Object not found 500 Persist exception
----------	--	---	---

Where

- id –profile identifier

When a profile is deleted, its metrics acquisition is stopped on all nodes, for which that profile was set.

Obtaining profile information

A profile information can be obtained with /api/profile/list request

API	Request	Response	Response status
WS API	<pre>SEND destination: /app/api/profile/list content-length: 101 { "requestId": "2e29bebf-9773-429e-84b6-287dc4d3d479", "realm": "/api/profile/list", "payload": { { "id": "2" } } }</pre>	<pre>MESSAGE destination: /user/service content-type: application/json; charset=UTF-8 subscription: sub-1 message-id: 3-16 content-length: 160 { "requestId": "2e29bebf-9773-429e-84b6-287dc4d3d479", "status": 200, "reason": "SUCCESS", "payload": [{ "id": 2, "name": "static", "rate": 30, "metrics": [3, 10], "rules": [1] }] }</pre>	200 OK 400 Object not found 500 Persist exception
REST API	<pre>POST: /api/profile/list "application/json; charset=utf-8" { "id": "2" }</pre>	<pre>{ "status": 200, "reason": "SUCCESS", "payload": [{ "id": 2, "name": "static", "rate": 30, "metrics": [3, 10], "rules": [1] }] }</pre>	200 OK 400 Object not found 500 Persist exception

Where

- id –profile identifier

If profile Id is set, the response will contain only that profile information. If profile Id is not set, the response will contain information about all the profiles on backend server.

For every profile, the response contains the same fields as /api/profile/update request.