

# Управление узлами

- [Создание узла](#)
- [Изменение узла](#)
- [Удаление узла](#)
- [Получение информации об узле](#)
- [Получение текущего состояния узла](#)

Управление узлами осуществляется при помощи [Websocket API](#) или [REST API](#)

## Создание узла

Новый узел создается автоматически при подключении к бэкенд-серверу нового узла, которого нет в БД.

## Изменение узла

Параметры узла могут быть изменены при помощи запроса `/api/node/update`

API	Request	Response	Response status
WS API	<pre>SEND destination: /app/api/node/update content-length: 121  {   "requestId": "70ed9798-6463-4741-818f-b0b9d7299b6f",   "realm": "/api/node/update",   "payload": {     {       "id": "2",       "profiles": ["1", "2"]     }   } }</pre>	<pre>MESSAGE destination: /user/service content-type: application/json; charset=UTF-8 subscription: sub-1 message-id: 3-25 content-length: 84  {   "requestId": "70ed9798-6463-4741-818f-b0b9d7299b6f",   "status": 200,   "reason": "SUCCESS" }</pre>	200 OK 404 Object not found 500 Persist exception
REST API	<pre>POST: /api/node/update "application/json; charset=utf-8" {   "id": "2",   "profiles": ["1", "2"] }</pre>	<pre>{   "status": 200,   "reason": "SUCCESS" }</pre>	200 OK 404 Object not found 500 Persist exception

Здесь:

- `id` – идентификатор узла
- `profiles` – список идентификаторов профилей, определяющих метрики, собираемые с данного сервера

## Удаление узла

Узел может быть удален при помощи запроса `/api/node/delete`

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

WS API	<pre>SEND destination:/app/api/node/delete content-length:100  {   "requestId":"a2d3fd81-4598-41cf-8840-e40770d597b4",   "realm":"/api/node/delete",   "payload":   {     "id":"2"   } }</pre>	<pre>MESSAGE destination:/user/service content-type:application/json; charset=UTF-8 subscription:sub-1 message-id:3-25 content-length:84  {   "requestId":"a2d3fd81-4598-41cf-8840-e40770d597b4",   "status":200,   "reason":"SUCCESS" }</pre>	<p>200 OK</p> <p>404 Object not found</p> <p>500 Persist exception</p>
REST API	<pre>POST: /api/node/delete "application/json; charset=utf-8" {   "id":"2" }</pre>	<pre>{   "status":200,   "reason":"SUCCESS" }</pre>	<p>200 OK</p> <p>404 Object not found</p> <p>500 Persist exception</p>

Здесь:

- id – идентификатор узла

Мониторинг удаленного узла прекращается.

## Получение информации об узле

Информацию об узле можно получить при помощи запроса /api/node/list

API	Request	Response	Response status
WS API	<pre>SEND destination:/app/api/node/list content-length:98  {   "requestId":"5c809c02-1082-418e-8423-db7dbff9605c",   "realm":"/api/node/list",   "payload":   {     "id":"3"   } }</pre>	<pre>MESSAGE destination:/user/service content-type:application/json; charset=UTF-8 subscription:sub-1 message-id:3-28 content-length:143  {   "requestId":"5c809c02-1082-418e-8423-db7dbff9605c",   "status":200,   "reason":"SUCCESS",   "payload":[     {       "id":3,       "host":"192.168.1.64",       "profiles":[1]     }   ] }</pre>	<p>200 OK</p> <p>404 Object not found</p> <p>500 Persist exception</p>

REST API	<pre>POST: /api/node/list "application/json; charset=utf-8" {   "id": "3" }</pre>	<pre>{   "status": 200,   "reason": "SUCCESS",   "payload": [     {       "id": 3,       "host": "192.168.1.64",       "profiles": [         1       ]     }   ] }</pre>	200 OK 404 Object not found 500 Persist exception
----------	---	--	---

Здесь:

- id – идентификатор узла
- host - адрес WCS-сервера, подключенного к бэкенду
- profiles - список профилей

Если указан идентификатор узла, то ответ будет содержать информацию только об этом узле. Если идентификатор не указан, ответ будет содержать информацию обо всех узлах на бэкенд-сервере.

## Получение текущего состояния узла

Текущее состояние узла можно получить при помощи запроса /api/node\_status/list

API	Request	Response	Response status
WS API	<pre>SEND destination:/app/api/node_status/list content-length:105  {   "requestId": "404ae542-b729-4b6e-bf28-ef85dc1f8630",   "realm": "/api/node_status/list",   "payload":   {     "id": "3"   } }</pre>	<pre>MESSAGE destination:/user/service content-type:application/json; charset=UTF-8 subscription:sub-1 message-id:3-31 content-length:171  {   "requestId": "404ae542-b729-4b6e-bf28-ef85dc1f8630",   "status": 200,   "reason": "SUCCESS",   "payload": [     {       "id": 3,       "rate": 160     }   ] }</pre>	200 OK 404 Object not found 500 Persist exception
REST API	<pre>POST: /api/node_status/list "application/json; charset=utf-8" {   "id": "3" }</pre>	<pre>{   "status": 200,   "reason": "SUCCESS",   "payload": [     {       "id": 3,       "rate": 160     }   ] }</pre>	200 OK 404 Object not found 500 Persist exception

Здесь:

- id – идентификатор узла
- rate – частота, с которой метрики присылаются WCS-сервером на бэкенд-сервер

Если указан идентификатор узла, то ответ будет содержать информацию только об этом узле. Если идентификатор не указан, ответ будет содержать текущее состояние всех узлов, наблюдаемых на бэкенде.

Если узел в данный момент не подключен к бэкенд-серверу, либо такой узел никогда ранее не подключался, запрос вернет ошибку 404OBJECT\_NOT\_FOUND