QUICK START GUIDE
ctrlX MOTION: REST API

GETTING STARTED

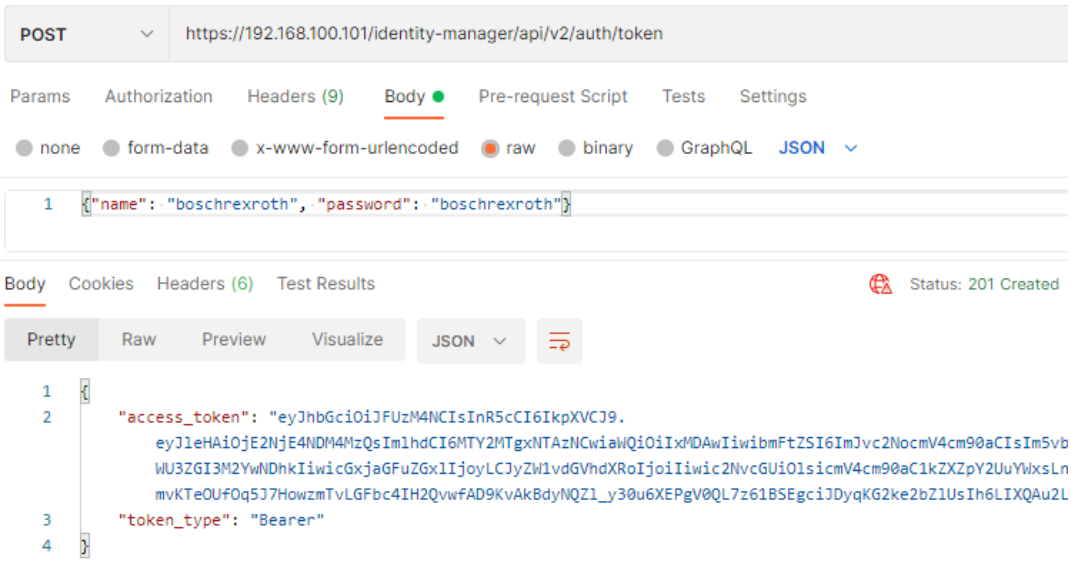
ctrlX CORE exposes its main functionality, and particular its so-called Data Layer, via a REST API. In this guide we provide some examples of what this looks like in the case of ctrlX CORE's MOTION app. For more information, see the article titled "USING REST API OF CTRLX CORE", available in the Bosch Rexroth Developer Community: https://developer.community.boschrexroth.com/.

An HTTP client is required to issue the requests shown in the following sections. We will be using Postman. Note that to simplify the configuration, SSL certificate verification is disabled throughout.

Our CORE has the MOTION app (V1.14.0) installed with licenses for standard (SWL-XCx-MOT-STDMOTIONxxxx-NNNN) and Cartesian (SWL-XCx-MOT-CARTESIANxxxx-NNNN) functionality.

1 Bearer token

To obtain the bearer token required for all additional requests to the API, POST your user credentials to /identity-manager/api/v2/auth/token.



Referring to the figure above, replace the name, password and IP address as required. All examples which follow assume that the request authorization type has been set to "Bearer Token" and that the access token returned from this call has been properly assigned.

2 Reset

To reset an axis error, issue a POST request to motion/axs/*axisName*/cmd/reset. Leave the request body blank.

```
POST
https://192.168.100.101/automation/api/v1.0/motion/axs/Axis_X/cmd/reset
```

3 Axis power

Data layer node motion/axs/*axisName*/cmd/power is boolean-valued. To enable an axis, set this boolean value to true by issuing a POST request to motion/axs/*axisName*/cmd/power with body ”value”: true, ”type”: ”bool8”. Note that the body type is JSON.

```
POST
https://192.168.100.101/automation/api/v1/motion/axs/Axis_X/cmd/power

body
{"value": true, "type": "bool8"}
```

To disable the axis, POST to the same node, setting in the request body value=false.

4 Move absolute

An absolute positioning move may be initiated by posting the required trajectory information to /motion/axs/*axisName*/cmd/pos-abs. Note that axis power (enable) is required.

```
POST
https://192.168.100.101/automation/api/v1/motion/axs/Axis_X/cmd/pos-abs

body
{
 "axsPos":3.35,
 "buffered":false,
 "lim":{"vel":100,"acc":10,"dec":10,"jrkAcc":0,"jrkDec":0}
}
```

5 Move velocity

Likewise, a constant velocity move may be initiated by posting the required trajectory information to /motion/axs/*axisName*/cmd/velocity.

```
POST
https://192.168.100.101/automation/api/v1/motion/axs/Axis_X/cmd/velocity

body
{
  "axsVel": 5,
  "driveVelMode": true,
  "buffered": false,
  "lim": {"vel": 10, "acc": 10, "dec": 10, "jrkAcc": 0, "jrkDec": 0}
}
```

6 Gantry

If Axes Axis_Y1 and Axis_Y2 have been configured are gantry axes, then they may be coupled by issuing a POST request to motion/axs/Axis_Y2/cmd/add-to-gantry with body ”masterName”: ”Axis_Y1”, ”buffered”: true. Here Axis_Y1 acts as the master. Note that both axes must be enabled prior to the request.

```
POST
https://192.168.100.101/automation/api/v1/motion/axs/Axis_Y2/cmd/add-to-gantry

{"masterName": "Axis_Y1", "buffered": true}
```

To de-couple the slave, POST to motion/axs/Axis_Y2/cmd/rem-frm-gantry with an empty body:

```
POST
https://192.168.100.101/automation/api/v1/motion/axs/Axis_Y2/cmd/rem-frm-gantry
```

7 Additional functionality

For additional commands, see motion/axs/*axisName*/cmd.