

Interface Control Document Between Buildbotics CNC Controller and External software

Revision 1

December 2024

Prepared for:

Commercial Civil Space
Lockheed Martin
NASA Tipping Point program

By:

Joseph Coffland
Buildbotics LLC

Change Log

Revision	Date	Description
1	December 2024	Initial document

Contents

Change Log	3
Purpose and Scope	5
References	5
Glossary	5
Interface Overview	5
Typical Usage Scenario	6
Configuring the Controller	6
API Authorization	6
API Specification	7

Purpose and Scope

This document describes how third-party software can connect to the Buildbotics CNC controller (Controller) over a network connection and control and monitor the controller remotely. The document is written for software developers intending to write such third-party software. It has been kept brief as the details are documented thoroughly in the referenced documentation. The following sections provide an overview of the interface and its documentation.

References

- [Buildbotics User Manual](#)
- [Buildbotics Controller API Reference](#)
- [Buildbotics Controller Configuration Variables](#)
- [Buildbotics Controller Internal Variables](#)
- [Buildbotics Controller WebSocket API](#)
- [Hypertext Transfer Protocol -- HTTP/1.1 - RFC 2616](#)
- [The WebSocket Protocol - RFC 6455](#)
- [The JavaScript Object Notation \(JSON\) Data Interchange Format - RFC8259](#)
- [CAMotics GCode Specification](#)
- [JSON Schema](#)
- [OpenAPI Specification v3.1.0](#)

Glossary

- JSON - JavaScript Object Notation
- HTTP - HyperText Transfer Protocol
- API - Application Programming Interface
- RFC - Request For Comment
- CNC - Computer Numerical Control
- GCode - The most widely used CNC programming language

Interface Overview

Third-party clients may connect to the Buildbotics Controller either via the hardwired Ethernet port or an 802.11 Wifi connection. Using this network connection the controller's web interface may be accessed by navigating to the controller's network address. For example, at `http://bbctrl.local/`. Making such a connection is described in the [Buildbotics User Manual](#).

Over this network channel, it is also possible to access the controller programmatically using both the HTTP and WebSocket interfaces. The API for the HTTP interface is described in the document [Buildbotics Controller API Reference](#). The API for the WebSocket interface is

described in [Buildbotics Controller WebSocket API](#). Using these APIs you can upload CNC program files (GCode), start, stop and pause programs and monitor program progress as well as the status of many other parts of the Buildbotics controller.

Typical Usage Scenario

In the typical use case, third-party software would first establish a WebSocket connection as described in [Buildbotics Controller WebSocket API](#). This provides access to a continuously updated JSON state object from the controller. The variables of the state object are described in [Buildbotics Controller API Reference](#) with additional variables in [Buildbotics Controller Configuration Variables](#) and [Buildbotics Controller Internal Variables](#).

Next a GCode file may be uploaded via the filesystem interface with an HTTP PUT request to the `/api/fs/{path}` endpoint where `{path}` is the target file name. Next an HTTP PUT to `/api/start/{path}` the program. The program can be paused via `/api/pause` and stopped with `/api/stop`. All the while, the status can be monitored via the state object.

The API also allows for jogging the machine via `/api/jog` and homing the machine via `/api/home`.

Configuring the Controller

There are many configuration options which control the machine's performance and limits. These may be modified programmatically by downloading a copy of the current configuration via a GET request to `/api/load`, modifying the resulting JSON data and uploading the changes via a PUT request to `/api/save`. Saving the config however, requires API authorization as described below. Configuration variables are described in [Buildbotics Controller Configuration Variables](#).

API Authorization

Authorization is only required for a few administrative operations such as uploading a new configuration file. Authorization is achieved by setting an HTTP cookie with the name `bbctrl-sid` to the value of the state variable `sid` and then making a PUT request to `/api/auth/login` with that cookie and the correct password in the JSON body of the request. Once the `sid` (Session ID) is authorized any of the endpoints which require authorization may be accessed by providing the `bbctrl-sid` cookie. Only those endpoints which specify the possibility of a 401 return code in the API documentation require authorization. See <https://api.buildbotics.com/#tag/authorization/PUT/auth/login>.

API Specification

The Buildbotics Controller API is documented in the following files:

- [Buildbotics Controller API Reference](#)
- [Buildbotics Controller Configuration Variables](#)
- [Buildbotics Controller Internal Variables](#)
- [Buildbotics Controller WebSocket API](#)

In addition to these human friendly documents, more precise formal specifications can be found in YAML format in the following files:

- [Buildbotics Controller API Specification](#) OpenAPI 3.1
- [Buildbotics Controller Configuration Variable Schema](#) JSON Schema 2020-12
- [Buildbotics Controller Internal Variable Schema](#) JSON Schema 2020-12