GoX Bukkit Plugin Short Guide

Permissions

GoX.Go - gives access to all user-level commands
GoX.GoM - gives access to all moderator-level commands

Config

lang.commands	Language preferences, you can edit messages the plugin sends to users. Plugin is shipped with *config.ru.yml* config file with russian messages.
lang.cyrillic_decoding	Sometimes decoder may give unreadable characters for russian users, setting this to true *may* help.
config.node_block	ID of the block that will be treated as a node. List of IDs can be found here. Default is Clay brick. For 1.13.2 (you should use capitalized name) here.
config.station_block	ID of the block that will be treated as a station. List of IDs can be found here. Default is Netherrack. For 1.13.2 (you should use capitalized name) here.
config.cart_velocity_multiplier	Allows to increase/decrease maximum speed of carts. Values higher than 1.25 may make carts laggy and reverse their speed on slopes/corners.
config.prohibited_stations	List of station names prohibited to register. Do not remove the default entries.
config.empty_cart_ticks_live	How long an empty cart can exist before being removed on a node/station. Set to 0 to disable cart removal.
config.chest_place_cart	Enable or disable the ability to place carts using chests.
backup.ticks_between_backups	Ticks between backups of the nodes and station map.

User-level commands

/go <station>

Set player destination to specified station and try to deliver him there when he enters the cart on a station or runs over a node/station block.

/go current

Gives player current destination.

/go cancel

Cancels the destination selection.

/go list [page]

Gives a player a list of registered stations.

/go info <station>

Gives a player information about the station.

/go closest

Gives a player a list of 3 closest stations.

Moderator-level commands

> *If you hear the word “graph” for the first time or don’t know what it is, I highly recommend you to watch some fancy video or check out the Wiki. Though I tried to keep everything as simple as possible.*

/gom add

> *This command can only be executed after placing rails on top of the node block e.g bricks.*

Registers a new node.

/gom addstation <name>

> *This command can only be executed after placing rails on top of the station block e.g netherrack.*

Registers a new station with specified name.

/gom info

Gives information about the node/station you are standing at.

/gom remove

Removes the node you are standing at.

/gom removestation <name>

Removes the station with specified name.

/gom setdrop <name>

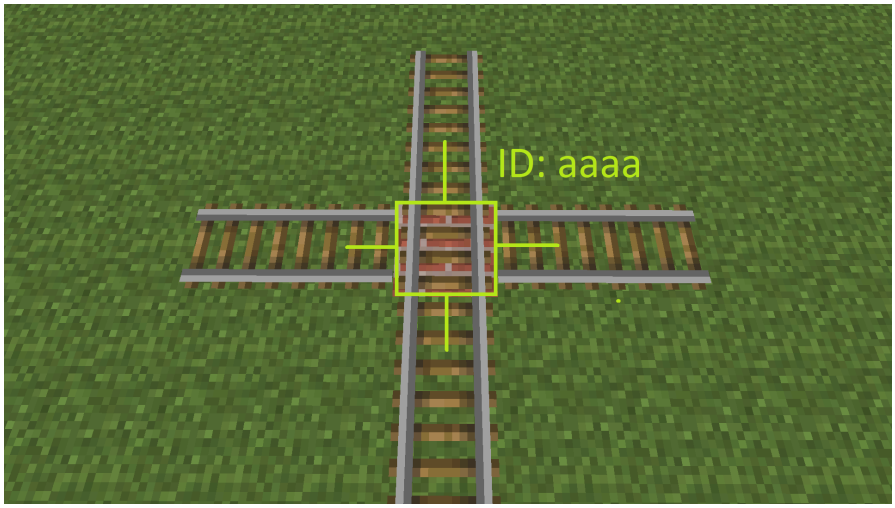
Sets the drop location for the specified station to your current location. Your line of sight is saved as well. Maximum distance is 8 blocks from the station.

/gom rename <old-name> <new-name>

Allows to rename a station.

Linking

Every node and every station has a unique ID that identifies that node/station. Every node or station can **point to** any other 4 nodes. There are only 4 available directions: *north*, *east*, *south*, *west*.



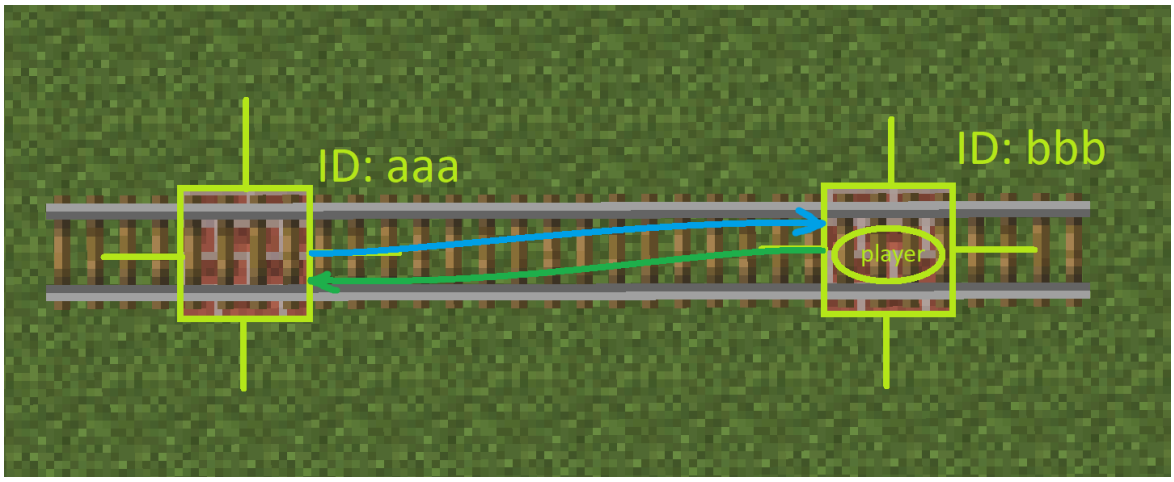
Correct linking is essential for good functioning of the plugin. There are 3 different linking commands.

/gom link <id>

Links the node that the player stands on to the node with specified ID. And links the specified node to the node under the player.

Example:

Player stands on the node *bbb* and executes command **/gom link aaa**. Now eastern side of the node *aaa* is linked to the *bbb*. Western side of the *bbb* is linked to the *aaa*. Note: we only link **one side** of the node to the other **node.**

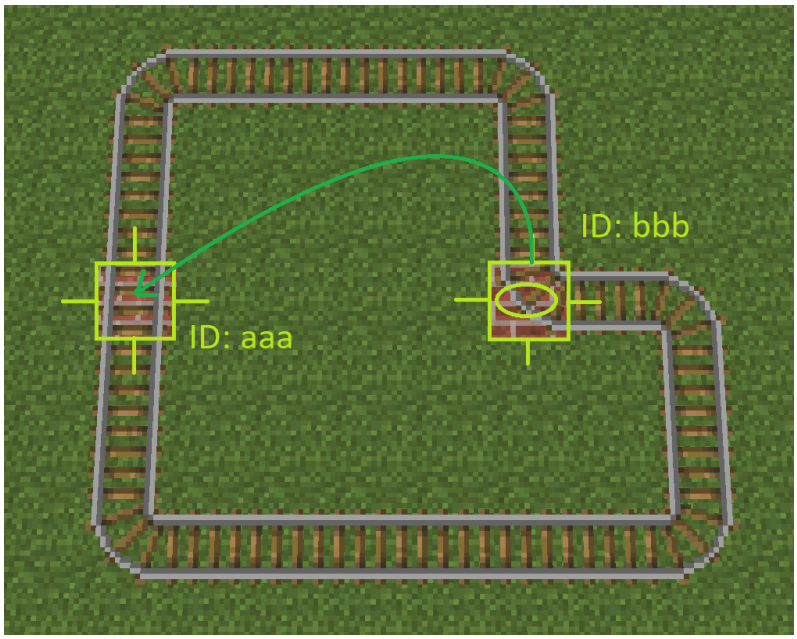


/gom link <direction> <id>

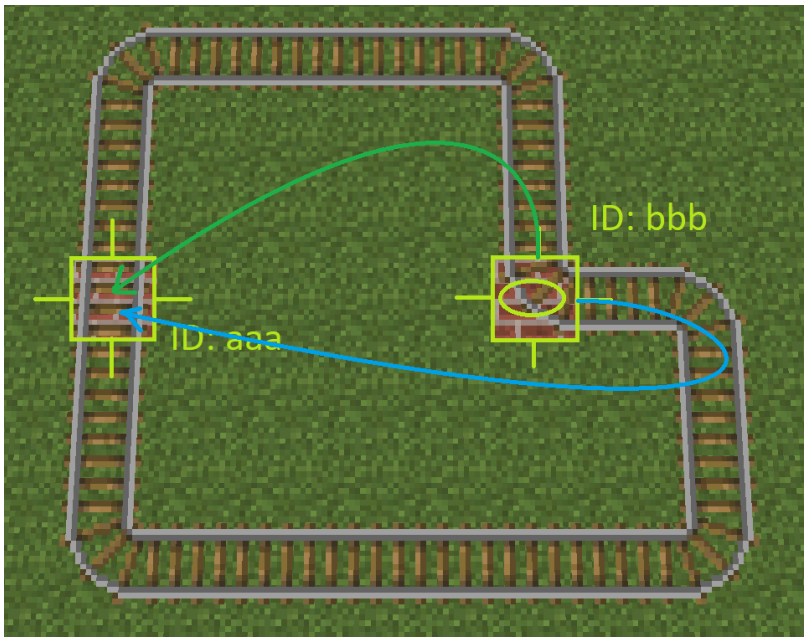
Links one side of the node that the player stands on to the node with specified ID. Does not link the other node back. It's a one-sided operation.

Example:

Here is the result of executing **/gom link north aaa**



Now we execute **/gom link east aaa**



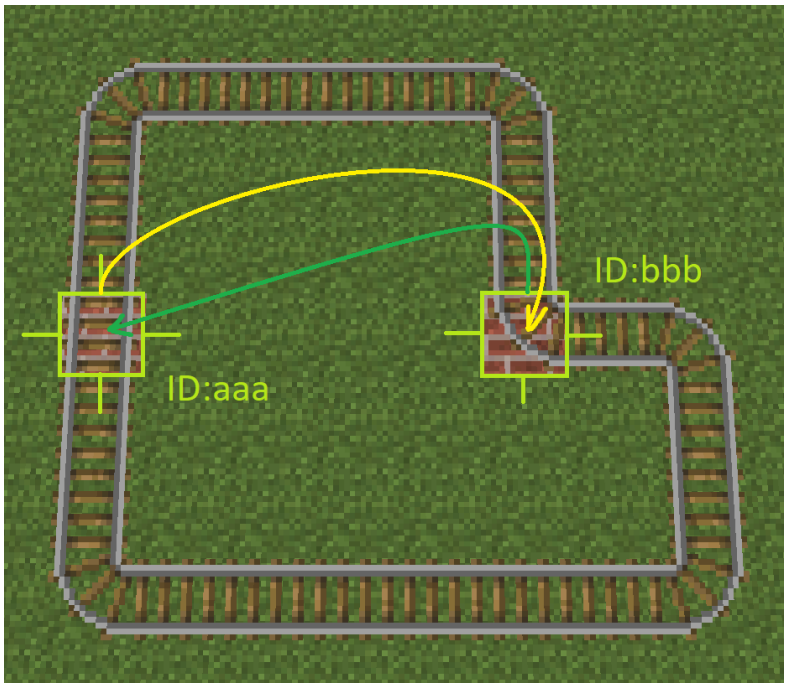
Please note that now there are two ways to get from *bbb* to *aaa*, but there is still no way to get from *aaa* to *bbb*! *bbb* is linked to *aaa* to the north and to the east, but *aaa* is not linked to anything!

/gom link <id> <direction> <id> <direction>

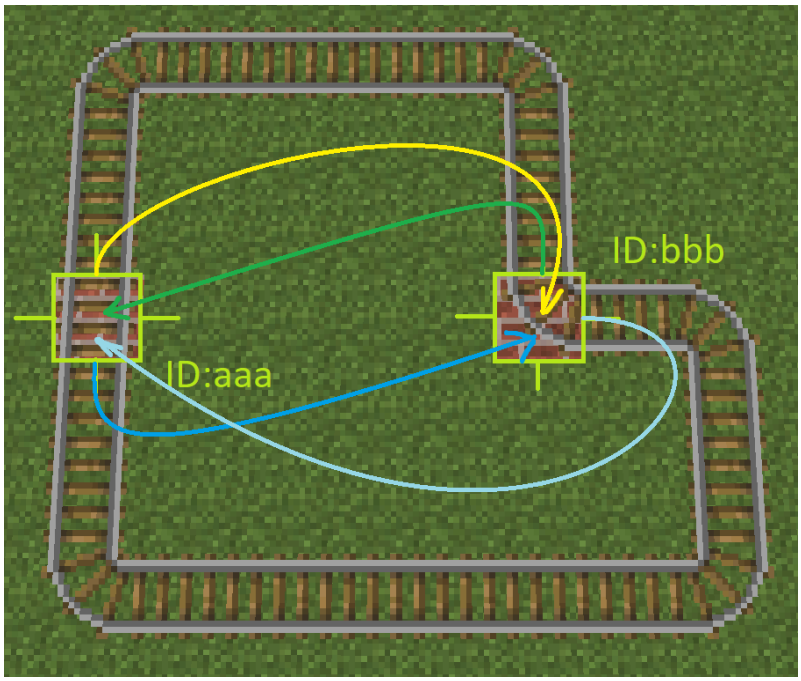
This command allow to link one side of one node to the other and link the side of the other node to the first one. This command affects both nodes and can be executed from anywhere.

Example:

/gom link aaa north bbb north



/gom link aaa south bbb east

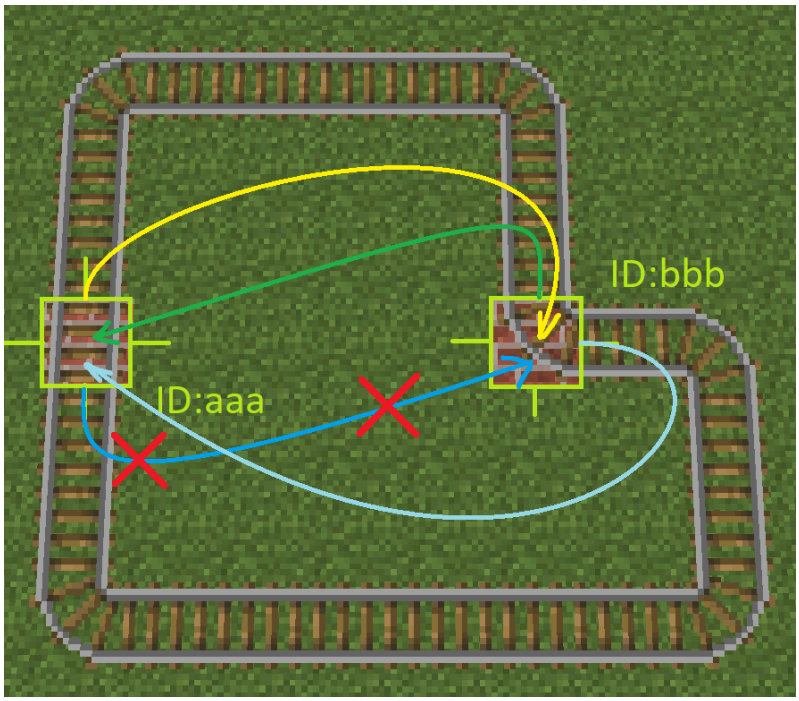


/gom unlink [id] <side>

Allows to remove a link from one side of the node under the player or specified by id.

Example (we use previous picture):

We stand on *aaa* and perform **/gom unlink south**



Now there are two ways from *bbb* to *aaa*, but only one way from *aaa* to *bbb*!

/gom link <id> <direction> <id>

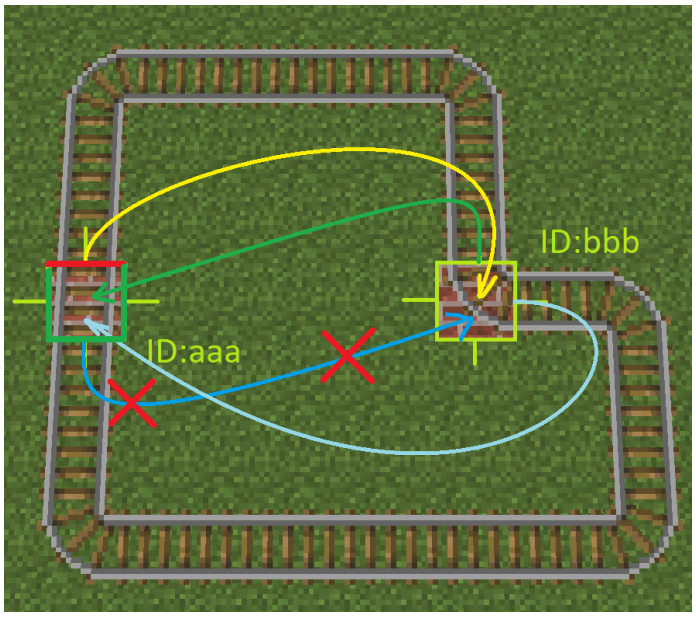
Allows to link one side of the first node to the other node. This command affects only the first node and can be executed from anywhere.

/gom force

Allows to reduce movement and pathfinding on a certain node to only 3 sides. Useful when you have an oncoming lane and want to reduce head-to-head collisions between the carts. Command applies the restriction to the node under the player's feet and takes their current **facing direction**.

Example:

Player stands on *aaa* looking *south* and executes **/gom force:**



Now *aaa* only allows to move west, east and south, which means there are no ways to get from *aaa* to *bbb* (the only northern way was prohibited by the **force** command).

/gom unforce

Removes any side restrictions.

/gom findpath <id1> <id2>

Tries to find a path from *id1* node to *id2*. Useful for testing your links and nodes.

Some examples: