

Updated: **Jan. 5th 2025** by glitchdetector

User Applications

This system allows loading any web app into the game, the web app is then fed various information automatically from the game client. The web app is shown directly in the game window, and can be interacted with by the player directly within the game.

See the example app at <https://cdn.tycoon.community/dev/userapp/sample.html>. It contains the basic code to listen to incoming data and send requests back to the game client.

Interface

By default, the F1 key opens the User Applications interface. The user is requested to input a Web URL for the web app they want to load.

The F1 key is used to regain focus to the web app and display the web app if it was hidden previously.

Commands

You can send commands to the game from your web app.

The command is sent as a JSON object with type set to the command name, and each argument as their own properties. Eg
`{type: "setWaypoint", x: 500, y: 250}`

Command	Parameters	Description
setWaypoint	x : number, y : number	Sets the in-game waypoint
sendCommand	command : string	Sends a console command to the game client
notification	text : string	Shows a notification over the map
info	text : string, time : number	Shows a lingering info message on the bottom right, time is in seconds
getData	No arguments	Forces the game client to send the entire data cache

Command	Parameters	Description
getNamedData	keys : array	Requests only the named data keys from the cache
close	No arguments	Gives focus back to the game and hides the web app
pin	No arguments	Gives focus back to the game, but keeps the web app on screen
sfx	sfx : number	Plays a SFX, see list below for sfx indexes
popup	title : string, text : string	Shows a full-screen text message
onelineer	text : string	Shows a black mid-screen box with text
message	text : string	Shows a mid-screen text message

Data

This is a (not complete) list of data that the game may provide. The data is generally provided when updated in-game.

Key	Type	Description
user_id	number	The user ID of the player
source	number	The player index of the player
name	string	The player's name
job	string	The current job the player has
wallet	number	Current wallet balance
bank	number	Current bank balance
vehicle	string	model name for current occupied vehicle (onFoot when on foot)
vehicleClass	number	class id for vehicle
vehicleName	string	display name for vehicle
vehicleMake	string	vehicle brand name
vehicleClassName	string	Class name for vehicle (not always based on vehicleClass)
honk	boolean	Is vehicle horn honking
car	string	owned spawned car model name
cab	string	owned spawned cab model name
trailer	string	owned spawned trailer model name
aircraft	string	owned spawned aircraft model name
helicopter	string	owned spawned helicopter model name
boat	string	owned spawned boat model name

Key	Type	Description
notification	string	Last shown notification
pos_x	number	Player Position X component
pos_y	number	Player Position Y component
pos_z	number	Player Position Z component
pos_h	number	Player rotation (heading)
zone	string	Map area ID
zoneName	string	Name of map area
street	string	Current street name
discord	string	Discord identifier for user, if present (in raw form discord:<id>)
[RUNWAY_ID]	string	Contains state of the runway with specified id, either free (yellow), occupied (red) or reserved (green). reserved being the local player having reserved it.
inventory	string	JSON of the current player inventory
weight	number	Current weight of inventory
max_weight	number	Capacity of inventory
waypoint	boolean	Is a waypoint set?
waypoint_x	number	Waypoint Position X component
waypoint_y	number	Waypointing Position Y component
menu	string	Current open vRP menu title
menu_choice	string	Last vRP menu button choice
chest	string	Internal ID of the current open chest (storage, trunk etc)
[chest_id]	string	JSON of the current inventory in the specified chest, only updates when chest is opened.
faction_id	number	Faction ID
faction_name	string	Name of the current faction
faction_tag	string	Faction chat tag
faction_president	boolean	Is the player the president of the faction?
pkey	string	The public API key for this user, if one is generated.
health	number	Player's current health
armor	number	Player's current armor
landing_gear	string	State of the landing gear. deployed, retracting, deploying, retracted or broken.
hidden	boolean	Is the web app hidden (closed)?
pinned	boolean	Is the web app pinned (shown but not in focus)?
focused	boolean	Is the web app in focus?
tabbed	boolean	Is this web app the current tab?

Player identifiers (like steam, license etc) are also provided when available, with the type as the key and the raw identifier as the value. Eg steam:
steam:12345678

Triggers

Triggers work just like data, and are sent as data. The key is always prefixed with `trigger_`. Triggers are not cached and will not be returned by `getData`. The value provided is the game client timer value, which is in milliseconds.

There are 4 built-in trigger binds that can be configured in Settings > Keybinds > FiveM. **Square**, **Triangle**, **Circle** and **Cross**.

These send `trigger_square`, `trigger_triangle`, `trigger_circle` and `trigger_cross` respectively. The command `userapp_trigger <key>` can send custom triggers, eg `userapp_trigger accept` would send `trigger_accept` to the web app.

Sound Effects

Using the `sfx` command, you can play a curated set of sound effects:

sfx id Name

- | | |
|----|-----------------------------|
| 1 | CHECKPOINT_MISSED |
| 2 | FLIGHT_SCHOOL_LESSON_PASSED |
| 3 | TIMER_STOP |
| 4 | Bed |
| 5 | MEDAL_UP |
| 6 | CHALLENGE_UNLOCKED |
| 7 | ScreenFlash |
| 8 | On_Call_Player_Join |
| 9 | Out_Of_Bounds_Timer |
| 10 | ROUND_ENDING_STINGER_CUSTOM |
| 11 | DELETE |
| 12 | OTHER_TEXT |
| 13 | GOLF_NEW_RECORD |
| 14 | GOLF_BIRDIE |
| 15 | GOLF_EAGLE |
| 16 | MP_RANK_UP |
| 17 | MP_WAVE_COMPLETE |