

LiveMap Resource

Jordan Dalton

04 September, 2019

Contents

LiveMap Resource	2
Installation	2
Configuration	2
Convars	2
Events	3
Client to server	3
Blip structure	4
Server Events	4
Frequently asked questions	5
How do I get blips?	5
I get a "dependency" error	5
I get a YARN error	5
Changelog	6
2.3.2 (2019-06-11)	6
Bug Fixes	6
Changes	6
2.3.1 (2019-06-10)	6
Bug Fixes	6
2.3.0 (2019-06-09)	6
Bug Fixes	6
Changes	6
Features	6
BREAKING CHANGES	6
2.2.12	7
Changes	7
Features	7
2.2.0	7
Bug Fixes	7
2.1.8	7
Bug Fixes	7
Features	7
2.1.7	7
Bug Fixes	7
Changes	7
2.1.6	8
Bug Fixes	8
Changes	8
2.1.4	8
Bug Fixes	8

Changes	8
Features	8
2.1.3	8
Bug Fixes	8
Changes	8
2.1.2	8
Changes	8
2.1.1	8
Changes	8
2.1.0	9
Bug Fixes	9
Changes	9
Features	9
2.0.0	9
Changes	9
Features	9

LiveMap Resource

This is the resource that sits on the FiveM server.

It's purpose is to send data to the interface via websockets to keep player location updated.

Installation

Download the ZIP file. And extract the contents into **resources/live_map/**.

Add the following to your server.cfg file.

```
set socket_port 30121
set livemap_debug "info" # "[all]" 'trace', 'debug', 'info', 'warn', 'error', 'fatal', 'off'
set blip_file "server/blips.json"
set livemap_access_control "*"

start live_map
```

To get the in-game blips to show on the interface, you will need to generate a "blips" file. This can be easily done with the in-game command **blips generate** (must be ran as admin).

Configuration

There isn't much to configure with the resource. It's built to be as plug-n-play friendly as possible.

The important stuff you may want to change can be done so with the convars below.

Convars

The following convars are available for you to change

Default			
Name	Type	Value	Description
-----	----	-----:	----
----	-:		
socket_port	int	30121	Sets the port the socket server should listen on

Name	Type	Default Value	Description
livemap_debug	string	"warn"	Sets how much information gets printed to the console ("[all]" 'trace', 'debug', 'info', 'warn', 'error', 'fatal', 'off')
blip_file	string	"server/blips.json"	Sets the file that will contain the generated blips that is exposed via HTTP
livemap_access_control	string	"*"	Sets the domain that is allowed to access the blips.json file (E.g. "https://example.com" will only allow the UI on http://example.com to get the blips), "*" will allow everyone

Events

In an effort to make the addon useful to other developers, I've created a few events that can be used to make changes to the data being sent to the UI.

Client to server

Below you can find some info on the server events that **must** be triggered by the client.

Note, when using `livemap:AddPlayerData` or `livemap:UpdatePlayerData` if the player has been removed using `livemap:RemovePlayer` they will be tracked again.

Name	Parameters	Description
livemap:AddPlayerData	key (string), data (any)	Adds data to a player that gets sent over Websockets
livemap:UpdatePlayerData	key (string), data (any)	Updates the data that is associated with the player. Uses the same "key" as the above event.
livemap:RemovePlayerData	key (string)	Removed data associated with the player. Uses the same "key" as the above events.
livemap:RemovePlayer		Stops sending a player data over Websockets
livemap:AddBlip	table (blip to add)	Adds a blip to the "blip_file" so that it can be showed on the web interface
livemap:UpdateBlip	table (blip to update)	Updates the blip to the table given

Example usage:

```
-- Set the player's "Name" to "Havoc"
TriggerServerEvent("livemap:AddPlayerData", "Name", "Havoc")

-- Update the player's name to "John Doe"
TriggerServerEvent("livemap:UpdatePlayerData", "Name", "John Doe")

-- Remove "Name" from the player (stops displaying it in the UI)
TriggerServerEvent("livemap:RemovePlayerData", "Name")

-- Removes a player from the websockets (stops tracking them)
TriggerServerEvent("livemap:RemovePlayer")

-- Adds a blip to the interface
TriggerServerEvent("livemap:AddBlip", {
    "sprite" = 16, -- Jet. You can find IDs at https://github.com/TGRHavoc/live_map-interface/blob/master/j
    "pos" = {
        x = 0.0,
        y = 0.0,
```

```

        z = 100.0
    },
    "name" = "This is a blip",
    "description" = "This is to show how one can add blips to the interface"
})

-- Updates the blip at the position.
TriggerServerEvent("livemap:UpdateBlip", {
    "sprite" = 51, -- Drugs
    "pos" = {
        x = 0.0,
        y = 0.0,
        z = 100.0
    },
    "name" = "I'm now a drug store",
    "description" = ""
})

```

Blip structure

The structure of the blip is as follows:

```

{
    sprite = Number (required) -- The sprite used when creating the blip
    pos = Table (requires) -- contains the X Y and Z coordinates of the blip
    {
        x = Float (required) -- X coordinate of the blip
        y = Float (required) -- Y coordinate of the blip
        z = Float (required) -- Z coordinate of the blip
    }
    name = String (optional) -- Name to show on the interface (defaults to the name of the blip e.g. "Drugs")
    description = String (optional) -- Description of the blip shown on the interface
}

```

Server Events

Below you can find information on some server-only events. This can only be called on the server.

Name	Parameters	Description
livemap:internal_AddPlayerData	identifier (string), key (string), value (any)	Adds data with the key that gets sent over Websockets for the player with the specified identifier
livemap:internal_UpdatePlayerData	identifier (string), key (string), value (any)	Updated the data that is associated with the player with the identifier
livemap:internal_RemovePlayerData	identifier (string), key (string)	Removed the data that is associated with the player with the identifier
livemap:internal_RemovePlayer	identifier (string)	Removes a player from the websocket data array (stops tracking the player)

Example usage:

```

-- Get the player's identifier
identifier = GetPlayerIdentifier(source, 0)

-- Set the player's "Name" to "Havoc"
TriggerEvent("livemap:internal_AddPlayerData", identifier, "Name", "Havoc")

```

```
-- Update the player's "Name" to "John Doe"
TriggerEvent("livemap:internal_UpdatePlayerData", identifier, "Name", "John Doe")

-- Removes the player's "Name"
TriggerEvent("livemap:internal_RemovePlayerData", identifier, "Name")

-- Removes the player
TriggerEvent("livemap:internal_RemovePlayer", identifier)
```

Frequently asked questions

Below you can find some frequently asked questions about the interface.

How do I get blips?

Whilst in game, run the command `blips generate`. That should be enough to generate the blips.json file.

Note: You must have the permission to run the command. Something along the lines of

```
add_ace group.YOUR_GROUP command.blips allow ## Allow a specific group to use the blips command
```

```
add_ace identifier.YOUR_IDENTIFIER command.blips allow ## Allow a specific player to use the blips command
```

I get a "dependency" error

This is known. Apparently the Node implmentation isn't up-to-par on the Linux servers for FiveM. The best thing to do is to install the dependencies yourself and try again.

1. Make sure NodeJS and NPM are installed for your distro
 2. Navigate to the resource's location
 3. Run `npm install`
 4. Remove dependency "yarn" from the resource file
 5. Try starting again
-

I get a YARN error

The solution to this was found by two community members, Kevin_Gorman and cnyncrvr.

Kevin_Gorman

We have just been doing some experimenting with it, to fix the issue, we had to download the new version of Yarn from the github and install it into an older version of FiveM to get it too work... If you use a new version of yarn in a new version of FiveM, it seems to just error...

cnyncrvr

Absolutely right, reverting back to the June 1st server build seems to make everything work correctly again

Changelog

2.3.2 (2019-06-11)

Bug Fixes

- **sockets:** removeplayer function `payload -> payload (0a5de15)`

Changes

- **sockets:** conform to own style (`ead5fc2`)

2.3.1 (2019-06-10)

Bug Fixes

- players not being removed after leaving server (`81811fb`)
- webpack and other dependencies not loading (`e1de1d6`)

2.3.0 (2019-06-09)

Bug Fixes

- events wrapper not being used (`c3f31ae`)
- **blips:** addblip incorrectly reporting duplicate blips (`b70b345`)

Changes

- change how blip controller is created (`2f4997f`)
- remove `C#` files (`ba6ef5a`)
- update server to use new controllers (`793512d`)
- update socket controller (`57a6094`)

Features

- add basic blip controller (`64ff008`)
- add basic server for node (`7892791`)
- add basic websocket server (`5bbe05`)
- add webpack (`6b0e711`)
- fully implement socketcontroller (`9a206ef`)
- move `blip_server` to node (`744ac41`)
- move wrapper to node (`e67776f`)

BREAKING CHANGES

- remove `C#` files

2.2.12

Changes

- add license to reverse_location_hashes (6625ce2)
- add license to update_check (30c84fa)
- remove console spam (701ca08)
- remove location data (67ad3b6)
- update client.lua to use new reverse function (81c6465)
- update the timer in update_check (c9fae47)

Features

- add location display and more (cb5be6f)
- add location to player data (db610a7)
- add plate check to client.lua (4b783ff)
- add reverse_car_hashes (4774170)

2.2.0

Bug Fixes

- add null checks to various API functions (bbfc23b), closes #24
- player location not being sent to sockets (0974a65), closes #21
- too many files open error (345e266)
- typo in client.lua and SocketHandler (461e92c)

2.1.8

Bug Fixes

- async issues (f51e974)
- async write errors (64583fe)
- Server_OnError error (265b3b6)

Features

- add event handlers for blips (acc71d4)

2.1.7

Bug Fixes

- race condition and null data (5a37bfc)

Changes

- add error handling to update_check (9f22b21)

2.1.6

Bug Fixes

- clients crashing when player leaves (ba7aa6f)

Changes

- create test.lua (46748a3)
- delete test.lua (bfc3a76)
- remove spammy prints (603e370)

2.1.4

Bug Fixes

- sending client data when websocket disconnects (f12837f)

Changes

- remove spammy traces (f5e856e)

Features

- add update_check (c592948)

2.1.3

Bug Fixes

- listener only listening on loopback address (2cb24eb)

Changes

- update server comments (d687c5f)

2.1.2

Changes

- update newtonsoft package (8fc3cf4)
- update socketHandler (3287ebe), closes #6

2.1.1

Changes

- update how players are handled (5fbd071)

2.1.0

Bug Fixes

- events not being registered (963711b)

Changes

- remove old files (5a99b20)
- slightly better logging (5940163)
- update gitignore (054eeee)

Features

- add ability to remove players and data (37c03b3)
- add allow-origin header (b1bb783)
- add blip generation with command (afb9996), closes #3
- add blip helper (dde519d), closes #2 #2
- add blips.json file (d250dda), closes #5
- add default client file (8dca6b3)
- add reverse weapon hash file (58d9bb9)
- add vehicle icons (69208fb)
- update various blip mechanics (5b5a193)

2.0.0

Changes

- add resource_manifest_version (912f3a1)
- remove file writer and console.writelines (1ee0619)
- remove license (4cc1b00)
- update comments (95cae9b)
- update O'Neil Ranch icon (d4613e6)
- update resource_manifest_version to the latest(?) one (9de1ccd)
- update websocket handler (9cda66b)

Features

- ability to add custom data to players (60af4ea)
- add blip helper (0e22f14)
- add gas station blips (ce6f4eb)
- add lua files (11a5e5f)
- add utility events (cfe1843)
- add vehicle data to player (e262ac8)
- fx server compatability (fee9958), closes #4 #1
- ssl support (4358ec9)