GHFU API DOCUMENTATION

This document serves as an informal documentation to the GHFU API between Martin's front-end and JERM Technology's Backend.

Server Address:

- IP: xxxx:xxxx:xxxx:xxxx
- Port: YYYYY
- Protocol: http (https in deployment)

Data format

All data communication within the API will be via JSON. The only exception is when the request is made from an unknown IP, in which case a simple message *"You are not authorised to access this server!"* and error code 401 will be returned

API-URLS

/test

Request JSON: None

Reply JSON: {"status": STRING}

- The STRING should be "Server is up!"

/register

Request JSON:

```
{
    "uplink": INT,
    "names": STRING,
    "deposit": FLOAT
}
```

- eg {"uplink": 3, "names": "Glayn Bukman", “deposit": 1350}

Reply JSON: {"id": INT, "log": STRING}

- If "id" is 0, error occurred, refer to "log" about the error details. Otherwise, the new account created has ID "id" and will from this point on be addressed by this returned ID

NB: In request JSON, "uplink" refers to the new account uplink’s account ID

/details

Request JSON: {"id": INT}

Reply JSON: depends;

If reply contains the key "log", error occurred, refer to the log otherwise, the reply will be in the format:

```
{
    "names": STRING,
    "id": INT,
    "uplink": STRING,
    "pv": FLOAT,
    "total_returns": FLOAT,
    "available_balance": FLOAT,
    "total_redeems": FLOAT,
    "commissions": [
        [
            FLOAT amount,
            STRING reason
        ],
            . . .
        ],
    "leg_volumes": [
        FLOAT leg-1-volume,
        FLOAT leg-2-volume,
        FLOAT leg-3-volume
        ],
    "investments": [
        [
        STRING date,
        FLOAT points,
        STRING package,
```

INT package id,

INT months_returned,

FLOAT returns

],

. . .

],

"direct_children": [STRING child1_names, STRING child2_names, ...]

}

/buy_package

Request JSON:

```
{
    "IB_id": INT,
    "amount": FLOAT,
    "buyer_is_member": BOOL,
    "buyer_names": STRING
}
```

Reply JSON: {"status": BOOL, "log": STRING}

- If "status" is “false”, error occurred, refer to "log"

NB: "IB_id" if the Independent Broker ID mediating the package buying

/data_constants

Request JSON: None

Reply JSON: depends;

If empty ({}), then error occurred. Otherwise, reply will be in format;

```
{
    "point-factor": FLOAT,
    "payment-day": FLOAT,
    "account-creation-fee": FLOAT,
    "annual-subscription-fee": FLOAT,
    "operations-fee": FLOAT,
    "minimum-investment": FLOAT,
    "maximum-investment": FLOAT
}
```

/set_data_constants

Request JSON:

```
{
   "point-factor": FLOAT,
   "payment-day": FLOAT,
   "account-creation-fee": FLOAT,
   "annual-subscription-fee": FLOAT,
   "operations-fee": FLOAT,
   "minimum-investment": FLOAT,
   "maximum-investment": FLOAT
}
```

Reply JSON:

```
{
   "point-factor": BOOL,
   "payment-day": BOOL,
   "account-creation-fee": BOOL,
   "annual-subscription-fee": BOOL,
   "operations-fee": BOOL,
   "minimum-investment": BOOL,
   "maximum-investment": BOOL
}
```

NB: Any number of constants may be omitted in the request JSON. That way, you can even set one constant. Also for any constant NOT in the known list of constants, its corresponding value will be “false” in the reply JSON

/invest

Request JSON:

```
{
    "id": INT,
    "amount": FLOAT,
    "package": STRING,
    "package_id": INT
}
```

Reply JSON: {"status": BOOL, "log": STRING}

- If "status" is “false”, error occurred, refer to "log"

/auto-refills

Request JSON:

```
{
	"data":
	[
		[FLOAT points, FLOAT %],
		[FLOAT points, FLOAT %],
		. . .
	]
}
```

eg

```
{
	"data":
	[
		[375, 59],
		[250, 49],
		[125, 68]
	]
}
```

Reply JSON: {"status": BOOL, "log": STRING}

- If "status" is “false”, error occurred, refer to "log"

NB: This URL should be called atleast 2 days before the payment day as the auto-refill percentages for that month are set here!