Backend API(<http://127.0.0.1:8080/>)

Contents

[Backend API(http://127.0.0.1:8080/) 1](#_Toc450737891)

[1. Content List 1](#_Toc450737892)

[2. Connect API 2](#_Toc450737893)

[3. Thumbnail 3](#_Toc450737894)

[4. Play Game 3](#_Toc450737895)

[5. Clean All 3](#_Toc450737896)

[6. Check Device Status 3](#_Toc450737897)

[7. User Login 3](#_Toc450737898)

[8. User Logout 4](#_Toc450737899)

[9. Reserved 4](#_Toc450737900)

[10. Error Codes 4](#_Toc450737901)

# Content List

Provide Game Contents List

* URL: /content\_list
* Paramters
  + “ca”(optional): game category idx
* Description

If “ca” parameter is specified, return games in that category, otherwise return all contents

* Result

{

“code”:”000”,

“msg”:”Succes”,

“count”:”1” // Content count

“datas”:[

{

“ct\_idx”:”1” // Content Idx

“price”:”0” // Content Price in Chinese money. Ex: 100 => 1 yuan

“version”:”0.4.1” // Game Version

“category”:”starship” // Game Category Name

“title”:”Game Title”

“developer”:”Game Developer Name”,

“age” : “Game User Age Limitation”

“upload\_date”: “Game Upload Date”.

“size”: “Total Game File size”

},

]

}

* Error Case:

no special error code. If there is no games, json “datas” should be []

# Connect API

* url: /connect
* Parameters
  + “device\_id” : WinJS Device ID
  + “name”: Device Name . registered in WinJS
  + “is\_test”: “yes”/”no”: only check for test connection
* Description

If “is\_test” == “yes”, just response ok, otherwise clean all (exit process and user logout)

* Result

{

“code”:”000”,

“msg”:”Success”,

“device\_idx”:3 // Server registered Device IDX

}

# Thumbnail

* url: /thumbnail
* Parameters
* Description
* Result

# Play Game

* url: /playgame
* Parameters
  + “idx”: “Content Idx”
  + “pay\_idx”:
  + “pay\_token”
* Description
* Result
  + “000” => “Success”
  + “311” => “Already running”
  + “312” => “Payment Failed”

# Clean All

* url: /cleanProcess
* Parameters
* Description
* Result

# Check Device Status

* url: /checkStatus
* Parameters

No paramters

* Description

Check the Device Status

* Result

{

“code”:”000”,

“msg”:”Success”,

“occupied”:”YES” / “NO” : YES is device is running game, NO=>device is idle

“device\_idx”: “2” // Device IDX that is registered in server

“action\_code”:

“001” => Can Replay? Have to Response in 10 seconds for Game Replay request

}

# User Login

* url: /usr\_login
* Parameters
  + “usr\_idx”: user idx for login
  + “access\_key”: base 64 encoded string from push server
* Description

Support Device QR Code User Login

* Result

{

“code”:”000”,

“msg”:”Success”,

“user\_idx”:”1”,

“user\_name”:”xxxxx”

“thumbnail”:”http://xxxx.com/xxx.png”

}

# User Logout

* url: /usr\_logout
* Parameters
* Description
* Result

# Get User Information

* url: /game/user
* Parameters: No Paramter
* Description

Used in Game. Game need to Get User Information who are playing game now

* Result

{

“code”:”000”,

“msg”:”Success”

“”user\_idx”

“user\_name”

“thumbnail”

}

# Reserved

* url: /???
* Parameters
* Description
* Result

# Error Codes

|  |  |  |
| --- | --- | --- |
| Code | Description |  |
| 000 | Success |  |
| 404 | Invalid Access |  |
| 301 | Invalid Paramters |  |
| 302 | Excution file path is invalid |  |
| 310 | No file Path |  |
| 311 | Process is already running |  |
| 351 | No Contents |  |
| 352 | Now Resolving |  |
| 353 | Content Download Failed |  |
| 361 | Invalid User |  |
| 362 | User Already Logined |  |
| 363 | No logined User |  |