**Function List**

Test Configuration link:

<http://101.200.214.15/setting/jsonServerSetting.php?name=jinli20180518Config&appid=IronMarine>

Note: above link, you can visit and test directly, if you want to modify the json file, please give json file to me, I help to put on our server, jinli20180518 is channel’s name, it is variable string, if another channel named abc123, the link would be:

<http://101.200.214.15/setting/jsonServerSetting.php?name=abc123Config&appid=IronMarine>

Feature: Better Rating

**boolean** IsRatingAvailable();

String GetRatingURL();

**Note: Done**

Feature: Remote Configuration & Offers

List<String> GetIAPs();

List<String> GetRemoteValuesWithPrefix(String prefix);

String GetRemoteValue(String key);

**Note: Done, but GetRemoteValuesWithPrefix(String prefix) will return string, because I don’t know which list to return, I will explain how is GetRemoteValuesWithPrefix works.**

**1:I modified “iaps“ in Json file, please check link, so GetIAPs() method will return all info of each merchandise. 2:GetRemoteValue(String key) will return the value of key from “configuration”.**

**3: GetRemoteValuesWithPrefix(String prefix) will find which identifier is prefix from creditPackOffers,** **heroOffers,** **specialOffers,** **powerOffers,** **contentOffers, for example, if prefix is “offer\_credits\_pack1”, GetRemoteValuesWithPrefix will return:**

**{**

**"identifier": "offer\_credits\_pack1",**

**"iapIdentifier": "com.ironhidegames.ironmarines.credits\_pack1",**

**"offerType": 0,**

**"offerCost": {**

**"currencyType": 0,**

**"cost": 0**

**},**

**"rewards": [{**

**"identifier": "offer\_credits\_pack1",**

**"rewardType": 2,**

**"count": 1200**

**}],**

**"metadata": {**

**}**

**}**

**Please let me know if that you want.**

Feature: Cloud Save

**void** SetCallback(CloudSaveCallback callback);

**void** Logout();

**void** Login();

**void** Load(String name);

**void** Save(String name, String data);

**Note: Done, but there have differences, some channels don’t have their own login systems, so we usually use IMEI or unique random number to be user ID, if I am right, “name” should be the unique string for each user, anyway I will give you “name” string in login success callback, you could use the unique string to be user ID.**

Feature: Rewarded Videos

**boolean** IsVideoAdAvailable();

**void** ShowVideoAd();

**void** SetVideoAdCallback(VideoAdCallback callback);

**Note: done, but you must use new UnityPlugin.jar to have that callback.**

Feature: Custom Unlock screen

**Note: no problem, game will not lock again if user already reached the level, if am right, I can set json like this:**

{

"identifier": "offer\_special\_content\_campaign",

"iapIdentifier": "com.ironhidegames.ironmarines. unlock\_campaign",

"offerType": 1,

"offerCost": {

"currencyType": 0,

"cost": 0

} ,

"rewards": [{

"identifier": "content\_campaign ",

"rewardType": 5,

"count": 1

} ],

"metadata": {

"value": "解锁关卡",

"offerType": "holyday\_halloween",

"duration": "0.12:59:59",

"unique": "true",

"checkPurchasedHeroes":"false"

}

}

**It is possible to give us all default configurations, so we can test and understand more deeply of the json structure.**

Feature: A/B Testing

**Note: for easy testing, we would try to modify channel name to test different configuration and then replace configuration for really channels first, our server will consider more solutions to do that in future by ourselves, but still look forward your suggestions if you have.**

The Explanation of Json

**Because there has a lot configuration value, I am not sure each value really looks like in game, I think the best solution for both of us is modifying json and test game, then we can adjust some works if we need.**

**As we mentioned before:**

**1: The banner and rewards could be modified by us.**

**2: Merchandise could be purchase or get for free with set days**

**3: Customize AD button and AD rewards by ourselves.**

****

**I would like to handle all setting in json, but I have to realize and test each function one by one, it would be best if you can test together.**