



GAME PROVIDER INTEGRATION API

VERSION 1.0

Table of Contents

1.	Introduction.....	4
2.	Pre-requisite.....	4
3.	Game Login.....	4
3.1	Request.....	4
3.2	Response.....	5
3.3	Failure response	5
3.4	Request Parameter	5
3.5	Response Parameter	5
4.	Get Balance.....	6
4.1	Request.....	6
4.2	Response.....	6
4.3	Failure response	6
4.4	Request Parameter	7
4.5	Response Parameter	7
5.	Fund Transfer.....	7
5.1	Request.....	8
5.2	Response.....	8
5.3	Failure response	8
5.4	Request Parameter	8
5.5	Response Parameter	9
6.	Game Settlement.....	9
6.1	Request.....	9
6.2	Response.....	10
6.3	Failure response	10
6.4	Request Parameter	10
6.5	Response Parameter	10
7.	Rollback	11
7.1	Request.....	11
7.2	Response.....	12

7.3	Failure response	12
7.4	Request Parameter	12
7.5	Response Parameter	13

1. Introduction

This integration guide is designed to help game providers integrate their games with the Marvel Gaming(MG) Aggregator platform. It details the necessary API calls and parameters required for a successful integration, ensuring that providers can seamlessly integrate their game content with Marvel and offer this to Marvel's end customers.

2. Pre-requisite

The game provider needs to:

- Provide their server IP addresses to be authorized to access our aggregator system.
- Ensure that only pre-registered IP addresses will have their requests and responses processed by our system.

Additionally, the game provider needs to:

- Provide their game list.
- Include a thumbnail image for each game.
- Include a unique identifier for each game to distinguish between games in the game provider's catalog.

3. Game Login

3.1 Request

```
{  
    "type": "login",  
    "tableId": Dttabl,  
    "userName": "testuser",  
    "currency": "USD",  
    "lang": "en",  
    "reloadUrl": ""  
}
```

3.2 Response

3.3 Failure response

```
{  
  "type": "login",  
  "status": "failure"  
}
```

3.4 Request Parameter

HTTP Method	POST		
Parameter Name	Description	Type	Length
type	Login	String	50
username	Player's username.	String	50
tableId	Game Id provided by the Game provider	String	14
lang	Player's language	String	5
currency	Standard code for currency e.g. USD (mandatory)	String	3
reloadUrl	Optional parameter. Need to be redirected when session close	String	500

3.5 Response Parameter

Parameter Name	Description	Type	Length
type	Login	String	24
token	Token received in the login validation response.	String	24
url	Game Launch URL	String	500

currency	Player's Currency	String	24
status	Success/failure	String	15

4. Get Balance

Marvel Aggregator exposes an API to get the player's balance for gameplay and to display their balance on the game screen. The balance API should be called before and after each bet transaction to ensure the correct amount is debited from the player's account.

4.1 Request

```
{
  "type": "balance",
  "userName": "testuser",
  "status": "success",
  "currency": "USD",
  "balance": 99798.50
}
```

4.2 Response

```
{
  "type": "balance",
  "token": "o2i240320105906192315185"
}
```

4.3 Failure response

```
{
  "type": "balance",
  "status": "failure"
}
```

4.4 Request Parameter

Parameter Name	Description	Type	Length
type	balance	String	24
token	Token received in the login validation response.	String	24

4.5 Response Parameter

Parameter Name	Description	Type	Length
type	balance	String	24
userName	Player's Username	String	24
status	Success/failure	String	15
currency	Player's Currency	String	8
balance	Player's current balance	Decimal	18,4

5. Fund Transfer

- When a player places a bet in the game:
 - A fund transfer request is submitted to the Marvel Aggregator via the fund transfer request URL.
 - The player's balance is adjusted upon receiving a successful response from the aggregator.
- Following the resolution of the game:
 - Subsequent fund transfer requests are made to the aggregator.
 - These requests specify the amount details for crediting the balance in the partner system.

5.1 Request

```
{
  "type": "bet",
  "transactionId": "8aee81b08df55893018e024176o4439f",
  "token": "o2i240320105906192315185",
  "round": "DT011709434813761",
  "tableId": "Dttable1",
  "currency": "USD",
  "amount": 500.0,
  "userName": "testuser"
}
```

5.2 Response

```
{
  "type": "bet",
  "userName": "testuser",
  "status": "success",
  "currency": "USD",
  "balance": 99793051450,
  "token": "o2i240320105906192315185",
  "transactionId": "8aee81b08df55893018e024176o4439f"
}
```

5.3 Failure response

```
{
  "type": "bet",
  "status": "failure",
  "currency": "USD",
  "balance": 99793051450,
  "token": "o2i240320105906192315185",
  "transactionId": "8aee81b08df55893018e024176o4439f"
}
```

5.4 Request Parameter

HTTP Method	POST		
Parameter Name	Description	Type	Length
type	bet	String	24
token	Token received in the login validation response.	String	24

transactionId	Transaction reference ID (Unique Identifier)	String	24
tableId	Unique Identifier in which the games are played	String	24
round	unique identifier for each game played	String	24
amount	Amount placed as Bet	Decimal	18,4

5.5 Response Parameter

Parameter Name	Description	Type	Length
type	bet	String	24
token	Token received in the login validation response.	String	24
transactionId	Transaction reference ID (unique identifier)	String	24
userName	Player's Username	String	24
currency	Player's Currency	String	8
balance	Player's current balance	Decimal	18,4
status	Success/failure	String	15

6. Game Settlement

Upon completion of the game, the game provider needs to settle the return-to-player amount for the round/hand.

6.1 Request

```
{
  "type": "betResult",
  "userName": "testuser",
  "transactionId": "qgdr73vfgexhoy40rhw5ack",
  "token": "o2i240320105906192315185",
  "round": "DT011709434813761",
  "tableId": "Dttable1",
  "currency": "USD",
```

```

    "amount": 10.0
}
```

6.2 Response

```
{
  "type": "betResult",
  "user": "testuser",
  "status": "success",
  "token": "o2i240320105906192315185",
  "currency": "USD",
  "balance": 99841051450
}
```

6.3 Failure response

```
{
  "type": "betResult",
  "user": "testuser",
  "status": "failure",
  "token": "o2i240320105906192315185",
  "currency": "USD",
  "balance": 0
}
```

6.4 Request Parameter

Parameter Name	Description	Type	Length
type	betResult	String	24
token	Token received in the login validation response.	String	24
transactionId	Transaction reference ID Unique Identifier	String	24
tableId	Unique Identifier in which the games are played	String	24
round	Unique identifier for each game played	String	24
amount	The amount the player received from the round.	Decimal	18,4

6.5 Response Parameter

Parameter Name	Description	Type	Length
type	betResult	String	24
token	Token received in the login validation response.	String	24
transactionid	Transaction reference ID Unique Identifier	String	24
userName	Player's Username	String	24
currency	Player's Currency	String	8
balance	Player's current balance	Decimal	18,4
status	Success/failure	String	15

7. Rollback

Upon game cancel or delay in bet response from the aggregator, the game provider should refund the players bet to the aggregator.

7.1 Request

```
{
  "type": "rollback",
  "transactionId": "8aee81b08df55893018e024176o4439f",
  "token": "o2i240320105906192315185",
  "round": "DT011709434813762",
  "tableId": "Dttable1",
  "amount": 500.0,
  "currency": "USD",
  "reverseTransactionId": "8aee81b08df55893018e024176o4439g"
}
```

7.2 Response

```
{
  "type": "rollback",
  "user": "testuser",
  "status": "success",
  "transactionid": "8aee81b08df55893018e02417604439f",
  "currency": "USD",
  "balance": 10001000000
}
```

7.3 Failure response

```
{
  "type": "rollback",
  "user": "testuser",
  "status": "failure",
  "transactionid": "8aee81b08df55893018e02417604439f", "currency": "USD",
  "balance": 10001000000
}
```

7.4 Request Parameter

Parameter Name	Description	Type	Length
type	rollback	String	24
token	Token received in the login validation response.	String	24
transactionId	Transaction reference ID (Unique Identifier)	String	24
tableId	Unique Identifier in which the games are played	String	24
round	unique identifier for each game played	String	24
amount	Amount to be rolled back	Decimal	18,4
reverseTransactionId	transactionid sent for the bet placement (Unique Identifier)	String	24

7.5 Response Parameter

Parameter Name	Description	Type	Length
type	rollback	String	24
token	token received in the login validation response	String	24
transactionid	Transaction reference ID (Unique Identifier)	String	24
userName	Player's Username	String	24
currency	Player's Currency	String	8
balance	Player's current balance	Decimal	18,4
status	Success/failure	String	15