



Seamless Api Specification  
ver 1.03

Last update 2024.05.06

## 1. 에이전트 준비 사항

### A. Request API 제공

B. api 통신은 HTTP Post 방식으로 연결하며, JSON 으로 요청 및 응답한다.

C. SmartApi backoffice 를 통해서 Seamless api url 를 등록해주세요.

i. /getcredit - 유저 밸런스 조회

ii. /placebet - 유저 배팅

iii. /returnresult - 유저 게임 결과

iv. /cancelbet - 유저 게임 취소

v. 예시) seamless api url : <https://api.smartapi.com> 일경우

1. <https://api.smartapi.com/{api 키}/{api 토큰}/getcredit> - 밸런스 api

2. <https://api.smartapi.com/{api 키}/{api 토큰}/placebet> - 배팅 api

3. <https://api.smartapi.com/{api 키}/{api 토큰}/returnresult> - 게임결과 api

4. <https://api.smartapi.com/{api 키}/{api 토큰}/cancelbet> - 유저 게임 취소 api

\* {api 키}와 {api 토큰}는 에이전트 백오피스에서 제공이 됩니다.

### D. Seamless Api - 유저 밸런스 조회

i. Request : <https://seamless-url/{api 키}/{api 토큰}/getcredit>

ii. Method : post

iii. Request Json format

1. {"user\_id":"aaa","username":"kim"}

변수명	Type	설명	필수유무
user_id	string	user account	필수
username	string	user name	필수

iv. Response - success

1. {"status":0, "balance":680699}

변수명	Type	설명
status	int	0:성공, 1:에러
balance	int	유저 밸런스 금액

v. Response - error

1. {"status":1, "error\_code":"오류에 대한 설명"}

### E. Seamless Api - 유저 배팅

i. Request : <https://seamless-url/{api 키}/{api 토큰}/placeBet>

ii. Method : post

iii. Request Json format

```
1. {"user_id":"aaa", "username":"kim", "game_id":"evo", "amount":1000,
    "betdata":{"uid":"b7ac7a1e-ab16-4c73-8ac1-
    82402124e904","transaction_id":"C714556123781692899","transaction_ref_id":
    "714556123781692899","session_id":"09830614-656b-416b-95b0-
    55dbc5a49226","user_id":"58","transaction_type":"credit","amount":0,"currency
    ":"KRW","game_id":"17cbfc4183ef369ffc86e9f6-
    r4w2ijpj3tob72mp","game_type":"baccarat","game_table_id":"p63cmvmwagte
    emoy","game_table_vid":"rpg6td5znd57gxvr","cash_before":2572}},
    "session":{"id":"111ssss3333","ip":"192.168.0.1"},"transaction_id":"aaa-bbb-ccc-
    ddd"}
```

변수명	Type	설명	필수유무
user_id	string	user account	필수
username	string	user name	필수
game_id	string	게임 ID (ex:evo, pp, bti etc)	필수
amount	int	베팅한 금액	필수
betdate	string	상세 베팅정보	필수
transaction_id	string	베팅에 대한 transaction_id	필수
session	object	Session 정보	필수
session.id	string	유니크 id 값 [최초 gameopen 시 전달된 값]	필수
session.ip	String	접속한 유저 ip 값	필수

iv. Response Json Format

```
1. {"status":0,
    "balance":680699,
    "transaction_id":"4c91f593-3aa9-42ac-af45-ccc8d6b2683a"}
```

변수명	Type	설명	필수유무
status	int	0:성공, 1:Error	필수
balance	String	유저 밸런스 금액	필수
transaction_id	String	유니크 transaction id	필수
error_code	String	Invalid Id Invalid Username Invalid Game Not enough balance	옵션 [오류에 대한 설명]

F. Seamless Api – 유저 베팅 결과

- i. Request : <https://seamless-url/{api 키}/{api 토큰}/returnresult>
- ii. Method : post
- iii. Request Json Format

```
1. {"user_id":"aaa","username":"kim","game_id":"evo","bet_amount":"1000",
  "bet_result":"1000", "betdate":"
  {"cust_id":"58","reserve_id":"574305538562101248","amount":"5000","cash_bef
  ore":102572,"xmldata":{"Bets":{"$":{"cust_id":"136859455","reserve_id":"574305
  538562101248","amount":"5000.00","currency_code":"KRW","platform":"W"},"B
  et":{"$":{"BetID":"0","BetTypeID":"1","BetTypeName":"Single
  bets","Stake":"5000.0000","OrgStake":"5000.0000","Gain":"8900.0000","IsLive":"
  0","NumberOfBets":"1","Status":"Opened","IsFreeBet":"0","FreebetAmount":"0",
  "RealAmount":"5000.0000","CreationDate":"2024-05-
  03T18:45:10","PurchaseBetID":"0","CommomStatusID":"0","Odds":"-
  128","OddsDec":"1.7800","BonusID":"0","ComboBetNumersLines":"0","Referen
  celD":"0","ReserveAmountType":"Real","ReserveAmountTypeID":"1","LineID":"2
  084310360","LiveScore1":"0","LiveScore2":"0","EventState":"","LineTypeID":"2","
  LineTypeName":"Asian
  Handicap","RowTypeID":"1","BranchName":"Soccer","LeagueID":"79878371","
  MasterLeagueID":"5647","EventName":"Verspah Oita vs Kochi United
  SC","LeagueName":"Japan - Football League","YourBet":"Verspah Oita
  0","HomeTeam":"Verspah Oita","AwayTeam":"Kochi United
  SC","TransEventName":"베르스파 오이타 vs 코치 유나이티드
  SC","TransEventTypeName":"아시아 핸디캡","TransLeagueName":"일본 -
  풋볼 리그","TransYourBet":"베르스파 오이타 0","TransHomeTeam":"베르스파
  오이타","TransAwayTeam":"코치 유나이티드
  SC","TransBranchName":"축구","EventTypeID":"0","EventDate":"2024-05-
  04T04:00:00","MasterEventID":"1179058786","EventID":"1179058786","NewMa
  sterEventID":"573913158289469440","NewEventID":"573913158289469440","
  NewLeagueID":"554194462147186688","NewLineID":"0HC5739131592457871
  68HMM","TeamMappingID":"0","Score":"0:0","EventTypeName":"FT Asian
  Handicap","Points":"0","BranchId":"1","EachWaySetting":"","}}}","trx_id":"171476
  190063267"},"","transaction_id":"aaa-bbb-ccc-ddd"}
```

변수명	Type	설명	필수유무
user_id	string	user account	필수
username	string	user name	필수
game_id	string	게임 ID (ex:evo, pp, bti etc)	필수
bet_amount	string	베팅한 금액	필수

bet_result	string	승리 금액	필수
betdate	string	상세베팅정보	필수
transaction_id	string	배팅에 대한 transcasction_id	필수

iv. Response – json

1. {"status":0, "balance":680699}

변수명	Type	설명	필수유무
status	int	0:성공, 1:Error	필수
balance	string	유저 밸런스 금액	필수
transaction_id	string	유니크 transaction id	필수
error_code	string	Invalid Id Invalid Username Invalid Game Not enough balance	옵션

G. Seamless Api – 배팅 취소

i. Request : <https://seamless-url/{api 키}/{api 토큰}/cancelbet>

ii. Method : post

iii. Request Json format

```
1. {"user_id":"aaa",
    "username":"kim",
    "game_id":"evo",
    "amount":1000,
    "transaction_id":"123122313",
    "session": {"id":"1231111", "ip":"192.168.0.1"}
}
```

변수명	Type	설명	필수유무
user_id	string	user account	필수
username	string	user name	필수
game_ID	string	게임 ID (ex:evo, pp, bti etc)	필수
amount	string	베팅한 금액	필수
transaction_id	string	유니크 트랜잭션 ID	필수
session	object	Session 정보	필수
session.id	string	유니크 id 값 [최초 gameopen 시 전달된 값]	필수
session.ip	string	접속한 유저 ip 값	필수

iv. Response – json

1. {"status":0, "balance":1200, "transaction\_id":"1302875839457"}

변수명	Type	설명	필수유무
status	int	0:성공, 1:Error	필수
balance	int	취소후 유저의 보유금	필수
transaction_id	string	유니크 transaction id	필수
error_code	string	Too late to cancel Invalid ID Invalid Username Invalid Game	옵션

## 2. KOSCA API 서버에 요청

A. Method : post

B. [HTTPS://{hostname}/{api 키}/{api 토큰}/{API 명}](https://{hostname}/{api 키}/{api 토큰}/{API 명})

C. Hostname : https://gateway.smartapi24.com

D. {API 키}, {API 토큰} 정보는 가입 후에 backoffice 에서 제공됩니다.

E. 제공되는 {api}는 다음과 같습니다.

- i. opengame : 게임실행 api
- ii. getgamelist : 슬롯 게임리스트 api
- iii. getbethistory : 유저의 배팅 정보에 대한 api

## F. 게임실행 Api 설명

i. API 명 : opengame

ii. Method : post

iii. Request Json Format :

1. {"user\_id":"aabbcc",  
"username":"kimkimLee",  
"game\_id":"pp",  
"room\_id":"101",  
"session":{

```

    "id": "4c91f593-3aa9-42ac-af45-ccc8d6b2683a",
    "ip": "123.123.123.123"
  }
}

```

## 2. API 변수 설명:

변수명	Type	설명	필수유무	비고
user_id	string	Site 에서 사용중인 유저 ID	필수	
username	string	Site 에서 사용중인 유저이름	필수	
game_id	string	게임사 GameID	필수	GameID 리스트 참고
room_id	string	슬롯사용시 슬롯 id	옵션	
session	object	Session 정보	필수	
session.id	string	유니크 id 값	필수	
session.ip	string	접속한 유저 ip 값	필수	

## 3. Response Json Format

```

{
  "status": "0",
  "message": "Success.",
  "entry":
    "https://evo.luckylove.asia/entry?params=Y2FzaW5vX2lkPWdhdW1peHY4OHdsMDAwMDEKc2lnbmF0dXJlPUVmMzIEKcGxheV9tb2RlPXJlYWxfbW9uZXkk&JSESSIONID=r42mp2rt3tociqnr42mq2wrtifb26ghb521885b",
  "entryEmbedded":
    "https://evo.luckylove.asia/entry?params=Y2FzaW5vX2lkPWdhdW1peHY4OHdsMDAwMDEKc2lnbmF0dXJlPUVmMVlFlpZlEKcGxheV9tb2RlPXJlYWxfbW9uZXkk&JSESSIONID=r42mp2rt3tociqnr42mq2wrtifb26ghb521885b&embedded"
}

```

변수명	Type	설명	필수유무
status	string	0:성공, 1:Error	필수
message	string	status 가 1 일때 오류내용 표시	옵션
entry	string	게임실행 url (새창, 새탭)	필수
entryEmbedded	string	게임실행 url (iframe 용)	옵션

## G. 게임리스트 Api 설명

### i. API 명 : getgamelist



- ii. Method : post
- iii. Request Json Format :
  - 1. { "game\_provider": "pp", "game\_type": "slot" }
  - 2. API 변수 설명:

변수명	Type	설명	필수유무	변수명
game_provider	String	게임사코드명 (ex: EVO, PP, etc)	옵션	문서 마지막에 Game_id 참조
game_type	String	게임타입 (ex: slot, roulette, black jack...)	옵션	

3. Response Json Format:

```
{
  "status": 0,
  "error_code": "Success.",
  "data": {
    "all": [
      {
        "gameID": "vs15seoultrain",
        "gameIdNumeric": 1705056856,
        "gameName": "Train to Seoul",
        "gameTypeID": "vs",
        "jurisdictions": [
          "99"
        ],
        "lines": 10,
        "frbAvailable": true,
        "variableFrbAvailable": true,
        "features": [
          "BUY"
        ],
        "demoGameAvailable": true,
        "sortingOrder": 1
      }
    ],
  },
}
```

변수명	Type	설명	필수유무
status	int	0:성공, 1:Error	필수
error_code	string	status 가 1 일때 설명표시	필수
data	object	게임리스트 정보	필수

data.all.gameID	string	해당 게임 id 를 게임실행을 할 경우에 room_id 에 사용해서 각각의 게임실행	필수
-----------------	--------	--	----

## H. Bet History

i. API명 : getbethistory

ii. Method : post

iii. Request Json Format:

1. {"user\_id":"","startdate":"","enddate":"","game\_id":"","game\_type":"","page":"","limit":""}

2. API 변수 설명:

변수명	Type	설명	필수사항	참고
user_id	String	유저아이디	옵션	
startdate	String	2024-01-01 00:00:00	옵션	
enddate	String	2024-01-01 23:59:59	옵션	
game_id	String	게임사코드명 (ex: EVO, PP, etc.)	옵션	
game_type	String	live, spin, sports	옵션	
page	Int	Bet history paging	옵션	Default: 1
limit	int	Bet History limit per page	옵션	Default: 100

3. Response Json Format

```
{
  "status": 0,
  "data": {
    "user_id": "103",
    "username": "test0004",
    "game_id": "pp",
    "transaction_ref_id": "69930730308008",
    "game_type": "live",
    "game_name": "441",
    "betAmount": 1000,
    "winAmount": 2000,
    "betdata": {
      "user_id": 45,
      "game_id": "pp",
      "amount": 1000,
      "session": {
        "id": "c6bc36e88215954cfb5b3dc7c9974226",
        "ip": ""
      }
    }
  }
}
```

W"162.158.119.11W"}}, "resultdata": "{W"user\_idW":45,W"game\_idW":W"ppW",W"bet\_amountW":2000,W"bet\_resultW":2000},"reg\_datetime":"2024-05-0100:17:25.257"}], "totalCount":701}

- 별첨자료 game\_id 및 room\_id

No	종류	게임사	game_id	room_id
1	라이브	에볼루션	evo	
2	슬롯	프로그마틱	pp	
2-1	라이브	프로그마틱	pp	101
3	스포츠	BTI	bti	
4	슬롯	BNG	bng	
5	슬롯	CQ9	cq9	
6	라이브	DCI	dic	
7	라이브	Gamplay	gameplay	
8	슬롯	Habanero	habanero	
9	미니게임	Smart soft	smartsoft	
10	슬롯	Oq	oq	
11	라이브	Wm	wm	
12	라이브	올벳	allbet	
13	슬롯	BG	bg	
14	슬롯	LG	lg	
15	슬롯	PG SOFT	pg	
16	라이브	Dream Gaming	dreamgame	