**HighLow OpenAPI**

default : [115.68.110.176:8090](http://115.68.110.176:8090/betting_time/1000)

[[목차]]

[**응답 규약**](#_e5gxfpi10jbe) **2**

[**엑세스 토큰**](#_jda1diq4tstq) **2**

[**전체 공지**](#_qucmidoxph3m) **2**

[**특정 유저(id)에 메시지 공지**](#_m04t5ygqgtd3) **3**

[**특정 유저(id)의 차단**](#_17vq3pubply) **3**

[**특정 유저(id)의 차단 해제**](#_lsl9z9setfzs) **4**

[**특정 유저(id)의 차단 여부 검사**](#_wtce49sdt6se) **4**

[**배팅시간 변경**](#_w9vtgjm9x8dy) **5**

[**배팅단위 금액 확인**](#_kfbj8it8r8jt) **5**

[**배팅단위 금액 설정**](#_25upuw6yuaqa) **5**

[**최대 배팅 가능 금액 반환 [Disabled]**](#_v4fnyhi10uay) **6**

[**최대 배팅 가능 금액 설정 [Disabled]**](#_g6g958y3n2cj) **6**

[**환경설정 값 반환**](#_pq9wxu8z885q) **7**

[**종료 게임 목록**](#_7wmn05ndzpqa) **7**

[**게임 배팅 목록**](#_7wod717oe4t8) **10**

[**포인트 차감/지급 로그 파라미터**](#_l58cofqpgad0) **13**

[차감](#_i7jfhxtlym5s) 13

[지급](#_tx7jsikjsm2o) 13

[**환불목록 카운트**](#_vgd99e6cm3e2) **14**

[**환불목록**](#_n5xg02qfhymn) **14**

[**환불**](#_gf5eopaj7wiu) **16**

[**서비스 정지**](#_tse8o9b4bod) **17**

[**서비스 시작**](#_fsugj3cidwf) **17**

[**서비스 정지 확인**](#_e7vn27e918zx) **18**

[**욕설 광고 필터 등록**](#_yxviy8pm6smh) **18**

[**욕설 광고 필터 삭제**](#_um9ni79etr9t) **19**

# 응답 규약

성공의 경우 {“result”:true}

실패의 경우 {“result”:false} 또한 HTTP-CODE가 500번으로 전송됨.

모든 응답은 Content-Type이 application/json으로 응답함.

# 엑세스 토큰

모든 요청에는 access\_token이 반드시 필요함.

access\_token은 게임 서버의 config.json에서 설정함. “admin\_web\_server\_access\_token”에 설정해야함.

만약 엑세스 토큰이 없거나 올바르지 않다면 다음의 에러가 반환됨.

{"result":false,"error":"invalid access\_token"}

# 전체 공지

전체 공지 인스턴스 메시지를 전송함. 주로 정기점검 이전에 메시지를 전송함. message query가 존재하지 않는 경우 응답은 실패함.

|  |
| --- |
| [http://주소:포트/announce?message=helo](http://115.68.110.176:8090/announce?message=helo)&access\_token=엑세스토큰 |

요청 예)

[http://115.68.110.176:8090/announce?message=hello](http://115.68.110.176:8090/announce?message=helo)&access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"message":"hello"}

# 특정 유저(id)에 메시지 공지

특정 유저에게만 인스턴스 메시지를 전송함. 주로 해당 유저에게 경고를 하거나 특정 메시지를 전당할 때 사용함. 해당 유저가 로그인 된 상태가 아니면 에러가 반환됨.

|  |
| --- |
| http://주소:포트/announce/아이디?message=메시지&access\_token=엑세스토큰 |

요청 예)

[http://115.68.110.176:8090/announce/jpwdnv?message=](http://115.68.110.176:8090/announce/jpwdnv?message=%EA%B4%91%EA%B3%A0%EC%84%B1)해킹하지 마세요&access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"id":"jpwdnv","message":"해킹하지 마세요"}

{"result":false,"error":"not exist id or not logined"}

# 특정 유저(id)의 차단

특정 유저의 로그인을 차단하고자 할 경우에 다음 명령을 사용함. 현재 접속중이라면 게임 턴의 종료와 상관없이 차단함. 차단 유저는 DB에 기록하여 지속적으로 차단됨. 본 기능은 본 게임에만 적용되는 특성이므로 Portal시스템에서의 차단과는 별도로 운영되는 특성을 가짐. Portal에서 차단을 해제해도 본 기능에서 특정 유저의 차단이 적용되었다면 차단으로 간주됨.

차단만료날짜시간을 지정하고 싶다면 아래 예시를 참조. 단 차단만료날짜 시간의 포맷은 YYYY-mm-dd HH:ii:ss형태만 가능함. 가령 2019-1-3 오후 3시 30분인 경우 ‘2019-01-03 15:30:00’형태로 써야함. 해당 시간이 지나면 차단이 자동으로 해제됨.

|  |
| --- |
| http://주소:포트/block\_user/아이디?access\_token=엑세스토큰  http://주소:포트/block\_user/아이디/차단만료날짜시간?access\_token=엑세스토큰 |

요청 예)

http://115.68.110.176:8090/block\_user/xxx?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

http://115.68.110.176:8090/block\_user/xxx/2018-11-22%2008:00:00?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"id":"xxx"}

{"result":true,"id":"xxx","expire\_time":"2018-11-22 08:00:00"}

만료시간을 지정한 경우 해당 만료시간을 경과했어도 해당 레코드는 drop\_users테이블에 기록되어 있음.

# 특정 유저(id)의 차단 해제

특정 유저의 차단된 상태를 해제함. Portal의 차단이 아닌 게임 시스템에서의 차단을 해제하는 명령임.

|  |
| --- |
| http://주소:포트/cancel\_block\_user/아이디?access\_token=엑세스토큰 |

요청 예)

http://115.68.110.176:8090/cancel\_block\_user/xxx?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"id":"xxx"}

# 특정 유저(id)의 차단 여부 검사

특정 유저의 차단된 상태를 검사함. Portal의 차단이 아닌 게임 시스템에서의 차단 여부를 검사하는 명령임.

|  |
| --- |
| [http://주소:포트/is\_block\_user/아이디?access\_token=](http://115.68.110.176:8090/is_block_user/xxx?access_token=64756ea95f85f91ab64cc8f4a6275c5f)엑세스토큰 |

요청 예)

<http://115.68.110.176:8090/is_block_user/xxx?access_token=64756ea95f85f91ab64cc8f4a6275c5f>

응답 예)

{"result":true,"id":"xxx","is\_blocked":true}

{"result":true,"id":"xxxx","is\_blocked":false}

# 배팅시간 변경

배팅시간을 변경함. 단, 이미 배팅이 진행중인 경우 다음 배팅 타이밍에 적용됨.

한번 적용한 배팅 시간은 config테이블에 ‘betting\_time’ name으로 기록이 되며, 서버가 재시작 되어도 가장 마지막에 설정한 값을 그대로 적용함.

|  |
| --- |
| [http://주소:포트/betting\_time/](http://115.68.110.176:8090/betting_time/1000)밀리세컨드?access\_token=엑세스토큰 |

요청 예)

<http://115.68.110.176:8090/betting_time/10000>?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"millisecons":10000}

# 배팅단위 금액 확인

1000, 5000, 10000등의 배팅단위 금액의 설정 상태를 반환함.

|  |
| --- |
| [http://주소:포트/betting\_unit\_amounts](http://115.68.110.176:8090/betting_unit_amounts)?access\_token=엑세스토큰 |

요청 예)

<http://115.68.110.176:8090/betting_unit_amounts>?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"betting\_unit\_amounts":[1000,5000,10000,50000]}

# 배팅단위 금액 설정

배팅단위 금액을 설정함. 콤마 구분으로 인자를 제공함. 숫자가 아닌 경우나 비어있는 경우 에러를 반환함. 금액은 순서를 지키지 않아서 내부적으로 소팅하여(asc -- 오름차순) 처리함.

단, 배팅단위 금액은 최대 9개까지만 설정이 가능함. 0은 자동적으로 설정됨. 그러므로 0을 제외하고 9개까지만 세팅해야함. 시스템에서는 9개를 초과하면 에러를 반환함. 9개를 초과하면 모바일에서 하단 중지 버튼이 잘리는 문제가 발생함.

|  |
| --- |
| [http://주소:포트/betting\_unit\_amounts/50000,100000](http://115.68.110.176:8090/betting_unit_amounts/50000,100000)?access\_token=엑세스토큰 |

요청 예)

[http://115.68.110.176:8090/betting\_unit\_amounts/100,1000,1000,50000,100000](http://115.68.110.176:8090/betting_unit_amounts/50000,100000)?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"betting\_unit\_amounts":[100000,50000]}

응답 에러 예)

{"result":false,"error":"invalid betting\_unit\_amounts"}

# ~~최대 배팅 가능 금액 반환 [Disabled]~~

Disabled in Highlow

시스템에 등록된 최대 배팅 가능 금액을 반환함.

|  |
| --- |
| http://주소:포트/max\_betting\_amount?access\_token=엑세스토큰 |

요청 예)

<http://115.68.110.176:8090/max_betting_amount?access_token=64756ea95f85f91ab64cc8f4a6275c5f>

응답 예)

{"result":true,"max\_betting\_amount":300000}

# ~~최대 배팅 가능 금액 설정 [Disabled]~~

Disabled in Highlow

최대 배팅 가능 금액을 설정함. 기본은 300,000임. 1~10,000,000까지 설정 가능함.

|  |
| --- |
| http://주소:포트/max\_betting\_amount/최대배팅금액?access\_token=엑세스토큰 |

요청 예)

[http://115.68.110.176:8090/max\_betting\_amount/300000?access\_token=64756ea95f85f91ab64cc8f4a6275c5f](http://115.68.110.176:8090/max_betting_amount/0?access_token=64756ea95f85f91ab64cc8f4a6275c5f)

응답 예)

{"result":true,"max\_betting\_amount":300000}

# 환경설정 값 반환

설정한 환경설정 정보를 반환함.

|  |
| --- |
| [http://주소:포트/get\_config](http://115.68.110.176:8090/get_config)?access\_token=엑세스토큰 |

요청 예)

<http://115.68.110.176:8090/get_config>?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"rows":[{"name":"betting\_time","val":"10000"}]}

|  |  |  |
| --- | --- | --- |
| **name** | **value 설명** | **비고** |
| betting\_time | 배팅 시간(단위 : milli-seconds) | 해당 값이 없는 경우 시스템은 default 5000(5초)를 사용. |

# 종료 게임 목록

현재 진행중인 게임 또는 비정상 종료된 게임을 제외하고 상태가 END인 완료된 게임 목록을 제공합니다. 정렬은 내림차순으로 제공합니다. 한번에 40개의 데이터를 제공하여 page를 통해서 다음 페이지를 조회할 수 있습니다.

|  |
| --- |
| [http://주소:포트/](http://115.68.110.176:8090/get_config)finished\_game\_history?access\_token=엑세스토큰[&page=?] |

* page : (옵션) 없는 경우 기본값 1

요청 예시)

<http://115.68.110.176:8090/finished_game_history?access_token=64756ea95f85f91ab64cc8f4a6275c5f>

응답 예시)

|  |
| --- |
| {  "result":true,  "game":"highlow",  "channel\_name":"",  "channel\_code":"",  "page":1,  "rows":[  {  "wave":278,  "seed":"val(Heart,7,RED)\_\_68cb7d4103468d901f0ec088d84b1412",  "hash":"8abd855af22c9ed6f000bad7e315b6720b3e3b658830249efc3e8aaee5583059",  "card":"H7",  "card\_number":7,  "prev\_card":"C3",  "highlow\_type":"HI",  "highlow\_rate":1.2,  "highlow\_high\_amount":0,  "highlow\_low\_amount":0,  "highlow\_amount":0,  "highlow\_reward":0,  "color\_type":"RED",  "color\_rate":1.98,  "color\_red\_amount":0,  "color\_black\_amount":0,  "color\_amount":0,  "color\_reward":0,  "number\_type":"NUM\_L",  "number\_type\_label":"3~8",  "number\_rate":1.7,  "number\_left\_amount":0,  "number\_right\_amount":0,  "number\_amount":0,  "number\_reward":0,  "total\_user\_amount":0,  "total\_user\_profit":0,  "total\_user\_reward":0,  "total\_users":0,  "house\_revenue":0  },  {  "wave":276,  "seed":"val(Spade,J,BLACK)\_\_d6bb1e4c7dcb350589761883b7085d8d",  "hash":"49f797f59acf247c67b59f9b37ae794335225eaac856dfea2b8e0d549e3e7931",  "card":"S11",  "card\_number":11,  "prev\_card":"D3",  "highlow\_type":"HI",  "highlow\_rate":1.2,  "highlow\_high\_amount":0,  "highlow\_low\_amount":0,  "highlow\_amount":0,  "highlow\_reward":0,  "color\_type":"BLACK",  "color\_rate":1.98,  "color\_red\_amount":0,  "color\_black\_amount":0,  "color\_amount":0,  "color\_reward":0,  "number\_type":"NUM\_R",  "number\_type\_label":"J,Q,K,A",  "number\_rate":2.6,  "number\_left\_amount":0,  "number\_right\_amount":0,  "number\_amount":0,  "number\_reward":0,  "total\_user\_amount":0,  "total\_user\_profit":0,  "total\_user\_reward":0,  "total\_users":0,  "house\_revenue":0  }  ]  } |

* $.result : true
* $.game : highlow
* $.channel\_name : 채널 이름
* $.channel\_code : 채널 코드
* $.page : 현재 페이지
* $.rows[]
  + $.rows[].wave : 회차
  + $.rows[].seed : 시드 (시드에는 카드 정보가 문자열로 랜덤 문자와 함께 포함되어 있습니다.)
  + $.rows[].hash : 해쉬 (시드에 대한 해쉬코드)
  + $.rows[].card : 카드
  + $.rows[].card\_number : 카드 숫자
  + $.rows[].prev\_card : 이전 카드
  + $.rows[].highlow\_type : 하이로우 타입(HI:하이, LO:로우, NA:해당사항 없음)
  + $.rows[].highlow\_rate : 하이로우 배당률
  + $.rows[].highlow\_high\_amount : 하이 배팅 금액
  + $.rows[].highlow\_low\_amount : 로우 배팅 금액
  + $.rows[].highlow\_amount : 하이로우 배팅 금액
  + $.rows[].highlow\_reward : 하이로우 지급 금액
  + $.rows[].color\_type : 카드 컬러(RED, BLACK, NA:해당 사항 없음 - (예)조커)
  + $.rows[].color\_rate : 컬러에 대한 배당률
  + $.rows[].color\_red\_amount : RED 배팅금
  + $.rows[].color\_black\_amount : BLACK 배팅금
  + $.rows[].color\_amount : 컬러에 대한 총 배팅금
  + $.rows[].color\_reward : 컬러에 대한 지급 금액
  + $.rows[].number\_type : 카드 숫자
    - NUM\_L:3~8,
    - NUM\_R:J,Q,K,A
    - JOKER:조커
    - NA:해당 사항 없음
  + $.rows[].number\_type\_label : 카드 숫자 레이블
    - number\_type이 NUM\_L이면 3~8가 들어감
    - number\_type이 NUM\_R이면 J,Q,K,A가 들어감
  + $.rows[].number\_rate : 카드 숫자 배당률
  + $.rows[].number\_left\_amount : 왼쪽 카드 숫자 배팅금액
  + $.rows[].number\_right\_amount : 오른쪽 카드 숫자 배팅금액
  + $.rows[].number\_amount : 카드 숫자 총 배팅금액
  + $.rows[].number\_reward : 카드 숫자 지급 금액
  + $.rows[].total\_user\_amount : 총 유저 배팅금액
  + $.rows[].total\_user\_profit : 총 유저 수익
  + $.rows[].total\_user\_reward : 총 유저 지급금액(실제 유저에게 지급한 금액)
  + $.rows[].total\_users : 총 유저수
  + $.rows[].house\_revenue : 하우스 수익(양수 : 수익, 음수 : 손해)

# 게임 배팅 목록

유저의 게임 배팅 목록을 제공합니다. 게임 배팅 목록은 한번에 40개를 제공하며, 현재 진행중인 게임 회차의 2회차 이전의 데이터까지만 제공합니다. 그러므로 현재 게임에 배팅한 내역은 미래의 2회차 게임이 진행될 때 확인이 가능합니다. 이렇게 하는 이유는 게임이 완전이 종료가 되어야만 게임 배팅관련 정보가 모두 완벽하게 구성이 되기 때문입니다.

|  |
| --- |
| [http://주소:포트/](http://115.68.110.176:8090/get_config)game\_betting\_history?access\_token=엑세스토큰[&page=?] |

* page : (옵션) 없는 경우 기본값 1

요청 예시)

|  |
| --- |
| <http://115.68.110.176:8090/game_betting_history?access_token=64756ea95f85f91ab64cc8f4a6275c5f&page=1> |

응답 예시)

|  |
| --- |
| {  "result":true,  "game":"highlow",  "channel\_name":"",  "channel\_code":"",  "page":1,  "rows":[  {  "wave":248,  "id":"test1",  "nick":"테스트1",  "betting\_type":"NUM\_L",  "betting\_type\_label":"3~8",  "stat":"END",  "before\_user\_point":21284385,  "after\_user\_point":21283385,  "card":"S3",  "card\_color":"BLACK",  "card\_number":3,  "prev\_card":"D12",  "amount":1000,  "rate":1.7,  "reward":1700,  "profit":700,  "rtime":"2019-04-03 20:51:50",  "refund\_amount":0,  "refunded\_amount":0,  "refunded\_time":"0000-00-00 00:00:00"  },  {  "wave":248,  "id":"test1",  "nick":"테스트1",  "betting\_type":"BLACK",  "betting\_type\_label":"BLACK",  "stat":"END",  "before\_user\_point":21283385,  "after\_user\_point":21282385,  "card":"S3",  "card\_color":"BLACK",  "card\_number":3,  "prev\_card":"D12",  "amount":1000,  "rate":1.98,  "reward":1980,  "profit":980,  "rtime":"2019-04-03 20:51:51",  "refund\_amount":0,  "refunded\_amount":0,  "refunded\_time":"0000-00-00 00:00:00"  }  ]  } |

* $.result : true
* $.game : highlow
* $.channel\_name : 채널 이름
* $.channel\_code : 채널 코드
* $.page : 현재 페이지
* $.rows[]
  + $.rows[].wave : 회차
  + $.rows[].id : 플레이어 아이디
  + $.rows[].nick : 플레이어 닉네임
  + $.rows[].betting\_type : 배팅 구분(HI:하이, LO:로우, BLACK:블랙, RED:레드, JOKER:조커, NUM\_L:3~8, NUM\_R:J,Q,K,A )
  + $.rows[].betting\_type\_label : 배팅 구분 레이블(HI, LO, RED, BLACK, NUM\_L, NUM\_R, JOKER)
    - betting\_type에 따라 레이블이 달라짐
      * HI : 어떤 카드가 선택되어야 HI인지 카드 정보가 들어감.
      * LO : 어떤 카드가 선택되어야 LO인지 카드 정보가 들어감.
      * RED, BLACK : 그대로 들어감.
      * NUM\_L : 3~8이 들어감.
      * NUM-R : J,Q,K,A이 들어감.
      * JOKER : JOKER가 들어감.
  + $.rows[].stat : 플레이 상태 (BETTING, PLAYING, END)
  + $.rows[].before\_user\_point : 배팅 전 포인트
  + $.rows[].after\_user\_point : 배팅 후 포인트
  + $.rows[].card : 카드
  + $.rows[].card\_color : 카드 색상
  + $.rows[].card\_number : 카드 번호
  + $.rows[].prev\_card : 이전 카드
  + $.rows[].amount : 배팅 금액
  + $.rows[].rate : 배팅률
  + $.rows[].reward : 지급 금액 (실제 지급 금액)
  + $.rows[].profit : 수익
  + $.rows[].rtime : 배팅 일시
  + $.rows[].refund\_amount : 환불 금액
  + $.rows[].refunded\_amount : 환불된 금액
  + $.rows[].refunded\_time : 환불 날짜 시간

# 포인트 차감/지급 로그 파라미터

게임에 배팅을 하거나 게임 결과 포인트(balance 또는 point)가 있는 경우 관련된 게임 차감/지급 로그와 함께 전송됩니다. 여기서는 게임의 포인트 차감/지급 항목에 대하여 설명합니다.

기본적으로 이러한 로그 정보는 ‘|’기호로 구분자를 구성하며 lvalue=rvalue형태의 항목을 가집니다.

가령 포인트 차감시 game=roulette|channel\_code=?|wave=?|id=?|code=?|amount=? … {생략} ...과 같은 형태로 로그가 전달됩니다.

## 차감

차감 요청은 유저가 배팅을 한 경우에 요청합니다.

|  |  |
| --- | --- |
| **변수** | **값** |
| game | highlow |
| channel\_code | 채널 코드. 채널 코드가 없는 경우 blank(‘’)가 전달됩니다. |
| wave | 회차 |
| id | 유저 아이디 |
| code | 유저 코드 |
| amount | 차감 금액 |
| betting\_type | 배팅 구분(HI:하이, LO:로우, BLACK:블랙, RED:레드, JOKER:조커, NUM\_L:3~8, NUM\_R:J,Q,K,A ) |
| betting\_type\_label | 배팅 구분 레이블(HI, LO, RED, BLACK, NUM\_L, NUM\_R, JOKER) |

NUM\_L : 3 ~ 8

NUM\_R : J,Q,K,A

입니다.

## 지급

지급은 유저가 배팅에 당첨된 경우에 요청합니다.

|  |  |
| --- | --- |
| **변수** | **값** |
| game | roulette |
| channel\_code | 채널 코드. 채널 코드가 없는 경우 blank(‘’)가 전달됩니다. |
| wave | 회차 |
| id | 유저 아이디 |
| code | 유저 코드 |
| betting\_type | 배팅 구분(HI:하이, LO:로우, BLACK:블랙, RED:레드, JOKER:조커, NUM\_L:3~8, NUM\_R:J,Q,K,A ) |
| betting\_type\_label | 배팅 구분 레이블(HI, LO, RED, BLACK, NUM\_L, NUM\_R, JOKER) |
| rate | 배당률 |
| betting\_amount | 배딩 금액 |
| profit | 수익 (지급 금액 - 배팅 금액) |
| reward | 지급 금액 (실제로 지급되는 금액) |
| loss | 손실 (배팅 금액 - 지급 금액) 단, 음수의 경우 0으로 처리 |

# 환불목록 카운트

|  |
| --- |
| [http://주소:포트/refund\_](http://115.68.110.176:8090/refund_list)count?access\_token=엑세스토큰  [http://주소:포트/refund\_](http://115.68.110.176:8090/refund_list/%EC%95%84%EC%9D%B4%EB%94%94)count[/아이디](http://115.68.110.176:8090/refund_list/%EC%95%84%EC%9D%B4%EB%94%94)?access\_token=엑세스토큰 |

아이디가 없는 경우 응답 예)

{"result":true,"count":1}

아이디가 있는 경우 응답 예)

{"result":true,"count":0,"id":"aloha"}

# 환불목록

게임의 환불은 반드시 목록을 확인한 이후에 반드시 환불 OpenAPI를 호출하여 내부적으로 로그가 남을 수 있도록 해야함. refund\_amount는 게임이 종료(턴 완료)되면 자동으로 0으로 설정됨. 그러므로 배팅과 함께 refund\_amount의 금액이 설정됨에 주의해야함.

정말로 해당 금액을 환불해야 하는지에 대해서는 게임 서버가 죽은 경우 해야함.

또한 이미 환불한 금액이 있는지도 잘 살펴봐야함.

|  |
| --- |
| [http://주소:포트/refund\_list](http://115.68.110.176:8090/refund_list)?page=페이지&access\_token=엑세스토큰  [http://주소:포트/refund\_list/아이디](http://115.68.110.176:8090/refund_list/%EC%95%84%EC%9D%B4%EB%94%94)?page=페이지&access\_token=엑세스토큰 |

아이디가 없는 경우 요청 예)

<http://115.68.110.176:8090/refund_list>?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

아이디가 없는 경우 응답 예)

{"result":true,"rows":[{"wave":1542601875,"id":"xescgr","before\_user\_point":5000,"amount":1000,"after\_user\_point":4000,"is\_auto\_betting":1,"check\_at":0,"auto\_check\_at":0,"profit":0,"check\_at\_time":null,"rtime":"2018-11-19T04:31:25.000Z","refund\_amount":1000,"refunded\_amount":0,"refunded\_time":null}]}

아이디가 있는 경우 요청 예)

http://115.68.110.176:8090/refund\_list/xescgr?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

아이다가 있는 경우 응답 예)

{"result":true,"rows":[{"wave":1542601875,"id":"xescgr","before\_user\_point":5000,"amount":1000,"after\_user\_point":4000,"is\_auto\_betting":1,"check\_at":0,"auto\_check\_at":0,"profit":0,"check\_at\_time":null,"rtime":"2018-11-19T04:31:25.000Z","refund\_amount":1000,"refunded\_amount":500,"refunded\_time":"2018-11-19T05:10:07.000Z"}],"id":"xescgr"}

|  |  |  |
| --- | --- | --- |
| column | 설명 | 비고 |
| wave | 게임 회차(일반적으로 unixtime stamp를 사용함) | wave = 회차 = 웨이브 |
| id | 유저 아이디 |  |
| before\_user\_point | 배팅 차감전 포인트 |  |
| amoount | 배팅 포인트 |  |
| after\_user\_point | 배팅 차감후 포인트 |  |
| is\_auto\_betting | 자동 배팅 여부(0 또는 1) |  |
| check\_at | 중지 위치 |  |
| auto\_check\_at | 자동 중지 위치(0인 경우 자동 중이 위치 사용 안함) |  |
| profit | 수익 |  |
| check\_at\_time | 중지 시간 |  |
| rtime | 레코드 등록 일시 |  |
| refund\_amount | 환불 금액 | 한 게임마다 정상적으로 모두 완료되는 경우 환불 금액을 0으로 설정함. 만약 게임이 비정상 종료된 경우 환불 금액이 그대로 남아 있음. 이러한 경우 해당 금액을 환불해야함. |
| refunded\_amount | 환불한 금액 | 환불 OpenAPI를 사용해서 환불한 금액. |
| refunded\_time | 환불한 시간 | 환불 OpenAPI를 사용해서 환불한 시간 |

# 환불

본 OpenAPI를 사용하면 game\_betting\_history테이블에 해당 회차의 환불한 금액과 환불한 시간(호출 기준)을 설정함.

|  |
| --- |
| [http://주소:포트/refund/회차(웨이브)/아이디/](http://115.68.110.176:8090/refund/1542601875/xescgr/500)환불금액?access\_token=엑세스토큰 |

요청 예)

http://115.68.110.176:8090/refund/1542601875/xescgr/500?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"wave":"1542601875","id":"xescgr","refunded\_amount":"500"}

# 서비스 정지

서비스 정지를 수행하면 마지막 게임까지만 진행하고 더 이상 게임을 진행하지 않음.

또한, 서비스가 재시작 되어도 DB의 service\_stop 테이블의 데이터를 가져와서 만약 데이터가 있는 경우 계속 서비스 정지 상태를 유지함. 그러므로 반드시 서비스 시작을 통해서 서비스를 가동해야함. 서비스가 정지가 되면 모든 유저는 웹 서버 단계에서 지속적으로 referesh하는 모드가 됨. 이후에 서비스가 시작되면 다시 게임모드로 변경됨. 메시지를 넣지 않는 경우 기본 메시지로 설정된 기본 메시지는 “Service is being inspected”이며 다국어 팩에서 번역되어 한국어의 경우 “서비스 점검중입니다…”가 제공됨.

|  |
| --- |
| [http://주소:포트/service\_stop](http://115.68.110.176:8090/service_stop)?message=메시지?access\_token=엑세스토큰 |

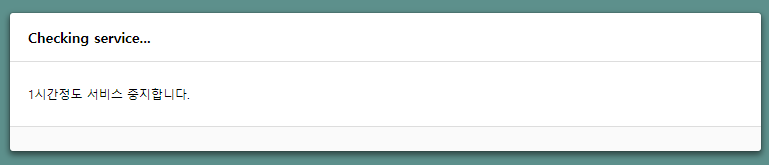
요청 예)

<http://115.68.110.176:8090/service_stop?message=1%EC%8B%9C%EA%B0%84%EC%A0%95%EB%8F%84%20%EC%84%9C%EB%B9%84%EC%8A%A4%20%EC%A4%91%EC%A7%80%ED%95%A9%EB%8B%88%EB%8B%A4>.&access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true}

게임 클라이언트 화면 예)



# 서비스 시작

서비스를 시작함. DB의 service\_stop 테이블의 데이터를 모두 삭제하고 서비스를 가동함.

|  |
| --- |
| [http://주소:포트/service\_start](http://115.68.110.176:8090/service_start)?access\_token=엑세스토큰 |

요청 예)

<http://115.68.110.176:8090/service_start>?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true}

# 서비스 정지 확인

서비스 정지 명령 이후에 정말로 서비스가 정지되어 있는지를 확인.

마지막 게임의 턴이 종료되어야만 서비스를 점검을 시작할 수 있음. 서비스 정지 명령을 수행한 이후에 정말로 점검을 해도 되는지 여부는 서비스 정지 확인 명령을 통해서 확인하거나 또는 웹 게임을 직접 접근해 본 이후에 수행함. 단, 웹 게임을 직접 접근해도 테스팅 모드인 경우에는 접속이 가능할 수 있으므로 주의해야함.

|  |
| --- |
| [http://아이피:포트/is\_service\_stopped](http://115.68.110.176:8090/is_service_stopped)?access\_token=엑세스토큰 |

요청 예)

<http://115.68.110.176:8090/is_service_stopped>?access\_token=64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"service\_stopped":false}

{"result":true,"service\_stopped":false,"service\_stop\_message":"서비스 점검중입니다."}

{"result":true,"service\_stopped":true,"service\_stop\_message":"서비스 점검중입니다."}

message문자열이 empty가 아닌 경우 이번 게임을 마지막으로 게임을 더 이상 운영하지 않음. 만약 message문자열이 없다면, 게임은 계속해서 실행됨.

service\_stopped가 true인 경우 서비스 점검이 시작됨.

# 욕설 광고 필터 등록

abuse\_filter테이블에 욕설 광고 필터를 등록함. 해당 내용은 게임 내에서 채팅시 관련된 단어가 발견되는 경우 전송을 차단하게 됨. db에 기록하고 시스템에 바로 적용됨. abuse\_filter테이블에 직접 등록하는 경우 시스템에서 60초에 한번씩 abuse\_filter내용을 전체적으로 업데이트 함.

|  |
| --- |
| [http://주소:포트/add\_abuse/추가할단어?access\_token=](http://115.68.110.176:8090/add_abuse/%EC%84%B9%EC%8A%A4?access_token=64756ea95f85f91ab64cc8f4a6275c5f)엑세스토큰 |

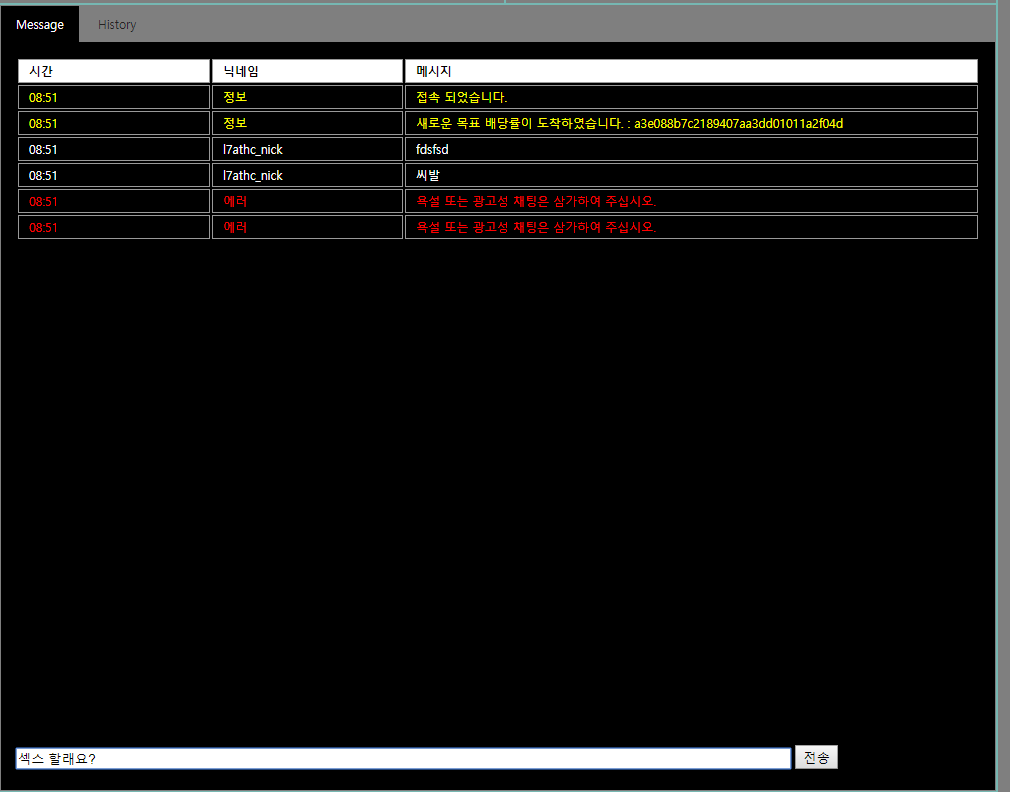
요청 예)

[http://115.68.110.176:8090/add\_abuse/섹스?access\_token=](http://115.68.110.176:8090/add_abuse/%EC%84%B9%EC%8A%A4?access_token=64756ea95f85f91ab64cc8f4a6275c5f)64756ea95f85f91ab64cc8f4a6275c5f

응답 예)

{"result":true,"word":"섹스"}

동작 예)



# 욕설 광고 필터 삭제

abuse\_filter테이블에 등록된 욕설 광고 필터를 삭제함. db에 기록하고 시스템에 바로 적용됨.

|  |
| --- |
| [http://주소:포트/delete\_abuse/삭제할단어?access\_token=](http://115.68.110.176:8090/delete_abuse/%EC%94%A8%EB%B0%9C?access_token=64756ea95f85f91ab64cc8f4a6275c5f)엑세스토큰 |

요청 예)

[http://115.68.110.176:8090/delete\_abuse/씨발?access\_token=64756ea95f85f91ab64cc8f4a6275c5f](http://115.68.110.176:8090/delete_abuse/%EC%94%A8%EB%B0%9C?access_token=64756ea95f85f91ab64cc8f4a6275c5f)

응답 예)

{"result":true,"word":"씨발"}