BSER OPEN API Models

Following are the models that can be used in the open API.

Get User Number

User numbers are unique id's used across the game.

Model URL

user/nickname?query={nickname}

Required Field

Nickname

The current nickname of the user.

Data Example

The response will come in as something like this.

```
{
    "code":200,
    "message":"Success",
    "user":{
        "userNum":1234567,
        "nickname":"ANONYMOUS"
    }
}
```

Top Rankers

This model can be used to acquire the top rankers of a season.

Normal matches are not accounted.

You must specify the season and team mode.

Model URL

rank/top/{seasonId}/{matchingTeamMode}

Required Field

Season ID

Please note that the season ID does not match the season numbering known to the public.

- ~~0 : Normal matches~~ (No longer supported.)
- 1 ~ N : Season

Matching Team Mode

- 1 : Solo
- 2 : Duo
- 3 : Squad

Data Example

The response will come in as something like this.

```
{
    "code":200,
    "message":"Success",
    "topRanks":[
      {
        "userNum":1234567,
        "nickname":"ANONYMOUS",
        "rank":1,
        "mmr":5176
      }
    ]
  }
```

V2 Model URL

v2/rank/top/{seasonId}/{server_name}/{matchingTeamMode}

Required Field

Season ID

Please note that the season ID does not match the season numbering known to the public.

- ~~0 : Normal matches~~ (No longer supported.)
- 1 ~ N : Season

Server Code

- 10 : Asia
- 17 : Asia2
- 12 : NorthAmerica
- 13 : Europe
- 14 : SouthAmerica

Matching Team Mode

- 1 : Solo
- 2 : Duo
- 3 : Squad

Data Example

The response will come in as something like this.

```
{
    "code":200,
    "message":"Success",
    "topRanks":[
        {
            "userNum":1234567,
            "nickname":"ANONYMOUS",
            "rank":1,
            "mmr":5176
        }
    ]
}
```

User Rank

This model can be used to acquire the rank of a specific player in a season.

Model URL

rank/{userNum}/{seasonId}/{matchingTeamMode}

Required Field

User Number

This field can be acquired by the Get User Number model.

Season ID

Please note that the season ID does not match the season numbering known to the public.

- ~~0 : Normal matches~~ (No longer supported.)
- 1 ~ N : Season

Matching Team Mode

- 1 : Solo
- 2 : Duo
- 3 : Squad

Data Example

The response will come in as something like this.

The serverCode and serverRank category will be added 21st of Nov. 2024

serverCode

- 10 : Asia
- 17 : Asia2
- 12 : NorthAmerica
- 13 : Europe
- 14 : SouthAmerica

```
{
    "code":200,
    "message":"Success",
    "userRank":{
        "userNum":1234567,
        "mmr":3933,
        "nickname":"ANONYMOUS",
        "rank":11, // total ranking
        “serverCode” : 10, // bound region server code
        “serverRank”:24 // rank within the server
```

```
      }
  }
```

Union Team

This model can be used to acquire union team information.

Model URL

unionTeam/{userNum}/{seasonId}

Required Field

User Number

This field can be acquired by the Get User Number model.

Season ID

- 1 ~ N : Season

Data Example

The response will come in as something like this.

```
  {
"code": 200,
"message": "Success",
"teams": [
  {
     "tnm": "team_name_01",
     "ti": 70,
     "stt": 2,
     "sstt": 0,
     "ssstt": 0,
     "ssti": 90,
     "ssstw": 0,
     "sstw": 1,
     "stw": 3,
     "atw": 0,
     "btw": 1,
     "ctw": 0,
     "dtw": 0,
     "etw": 0,
     "ftw": 0,
     "cdt": 1735034124000,
     "udt": 1737013242000
  }]
  }
```

Data Definitions

Teams

Key	DataType	Description
tnm	str	The name of the team.
ti	int	The current tier of the union team.
stt	int	Number of S tier tickets.
sstt	int	Number of SS tier tickets.
ssstt	int	Number of SSS tier tickets.
ssti	int	Top achieved tier this season.
ssstw	int	Number of wins in SSS tier.
sstw	int	Number of wins in SS tier.
stw	int	Number of wins in S tier.
atw	int	Number of wins in A tier.
btw	int	Number of wins in B tier.
ctw	int	Number of wins in C tier.
dtw	int	Number of wins in D tier.
etw	int	Number of wins in E tier.
ftw	int	Number of wins in F tier.
cdt	int	Dtm of team creation. Epoch time.
udt	int	Dtm of last update. Epoch time.

User Stats

This model can be used to acquire some of the basic statistics of a player in a season.

The V1 model will be deprecated in a few seasons.

The V2 model has been added to support further development.

Model URL

user/stats/{userNum}/{seasonId}

Required Field

User Number

This field can be acquired by the Get User Number model.

Season ID

Please note that the season ID does not match the season numbering known to the public.

- ~~0 : Normal matches~~ (No longer supported.)
- 1 ~ N : Season

Data Example

The response will come in as something like this.

The new rank mode data will be added in 16th of Jan, 2025, when the content becomes live.

```
{
    "code":200,
    "message":"Success",
    "userStats":[
        {
            // Normal/Rank information
            "seasonId":3,
            "userNum":1234567,
            "matchingMode":3,
            "matchingTeamMode":1,
            "mmr":2431,
            "nickname":"ANONYMOUS",
            "rank":311,
            "rankSize":47830,
            "totalGames":146,
            "totalWins":10,
            "totalTeamKills":203,
            "rankPercent":0.01,
            "averageRank":9.36,
            "averageKills":1.4,
            "averageAssistants":0,
            "averageHunts":17.28,
            "top1":0.07,
            "top2":0.12,
            "top3":0.16,
            "top5":0.28,
            "top7":0.39,
            "characterStats":[
                {
                    "characterCode":3,
                    "totalGames":145,
                    "usages":145,
                    "maxKillings":8,
                    "top3":24,
                    "wins":10,
                    "top3Rate":0,
                    "averageRank":9
    },
    {
        "characterCode":22,
        "totalGames":1,
        "usages":1,
        "maxKillings":0,
        "top3":0,
        "wins":0,
        "top3Rate":0,
        "averageRank":17
    },
  {
//   Upcoming new rank mode information
"seasonId":3,
"userNum":1234567,
"matchingMode":3,
"matchingTeamMode":1,
"mmr":2431,
"nickname":"ANONYMOUS",
"rank":311,
"rankSize":47830,
"totalGames":146,
"totalWins":10,
"totalTeamKills":203,
"rankPercent":0.01,
"averageRank":9.36,
"averageKills":1.4,
"averageAssistants":0,
"averageHunts":17.28,
"top1":0.07,
"top2":0.12,
"top3":0.16,
"top5":0.28,
"top7":0.39,
"characterStats":[
                    {
                         "characterCode":3,
                         "totalGames":145,
                         "usages":145,
                         "maxKillings":8,
                         "top3":24,
                         "wins":10,
                         "top3Rate":0,
                         "averageRank":9
                    },
                    {
                         "characterCode":22,
                         "totalGames":1,
                         "usages":1,
                         "maxKillings":0,
                         "top3":0,
                         "wins":0,
                         "top3Rate":0,
                         "averageRank":17
                    }

               ]
          }
     ]
}
```

V2 Model URL

v2/user/stats/{userNum}/{seasonId}/{matchingMode}

Required Field

Season ID

Please note that the season ID does not match the season numbering known to the public.

- 0 : Normal matches
- 1 ~ N : Season

Matching Mode

- 2 : Normal
- 3 : Ranked

Data Definitions

userStats

User statistics for each game mode.

Key	DataType	Description
seasonId	int	Season Id.
userNum	int	Unique number identifier of the user.
matchingMode	int	2 : Normal 3 : Ranked
matchingTeamMode	int	1 : Solo 2 : Duo 3 : Squad
mmr	int	user MMR.
nickname	string	Nickname of the user.
rank	int	User ranking.
rankSize	int	Total pool of users in the current rank.
totalGames	long	Total played games.
toatlWins	long	Total games ended in 1st place.
totalTeamKills	int	Total kills scored by the team. (Regardless of the team mode)
rankPercent	float	Ranking percentile.
averageRank	float	Average rank of all matches in the season.
averageKills	float	Average kills of all matches in the season.
averageAssistants	float	Average assists of all mataches in the season. (Misnamed variable.)
averageHunts	float	Average hunt count of all matches in the season.
top1	float	Percentile for achieving Top 1.
top2	float	Percentile for achieving Top 2. (or above)
top3	float	Percentile for achieving Top 3. (or above)
top5	float	Percentile for achieving Top 5. (or above)
top7	float	Percentile for achieving Top 7. (or above)
characterStats	characterStat	Array of character statistics.
seasonId	int	Season Id.

characterStat

Character statistics.

Key	DataType	Description
characterCode	int	Character code.
usage	long	Number of matches played as the character.
maxKillings	int	Max kill streak in a single match this season.
top3	int	Number of matches that ended in Top 3. (or above)
wins	int	Number of matches that ended in Top 1.
top3Rage	float	Top 3 rate this season.
averageRank	float	Average rank this season.

User Matches

This model can be used to acquire all the matches played within the **last 90 days** by a user.

Model URL

user/games/{userNum}

Required Field

User Number

This field can be acquired by the Get User Number model.

Data Example

```
{
    "code":200,
    "message":"Success",
    "userGames":[
        {BattleUserResult}
    ]
}
```

Data Definitions

See Definitions.

Match Results

This model can used to acquire battle game results of a single match.

Battle game results of all the players will be returned.

Model URL

games/{gameId}

Required Field

Game ID

Game ID can be acquired through other battle game results.

The best way to acquire this value is to use the User Matches model.

Data Example

```
{
    "code":200,
    "message":"Success",
    "userGames":[
        {BattleUserResult}
    ]
}
```

Data Definitions

Battle Game Results

Key	DataType	Description
userNum	int	Unique number identifier of the user.
nickname	string	Nickname of the user.
gameId	int	Uniqure number identifier of the game.
matchingMode	int	2 : Normal 3 : Ranked
matchingTeamMode	Int	1 : Solo 2 : Duo 3 : Squad
seasonId	Int	Season Id.
characterNum	Int	User's character code.
skinCode	int	User's skin code for the character.
characterLevel	int	The level of the character upon death/victory
gameRank	int	Final rank of the user.
playerKill	int	Number of kills during match by the user.
playerAssistant	int	Number of assists during match by the user.
monsterKill	int	Number of wildlife killed during match by the user. (This includes epic monsters like Wickeline.)
bestWeapon	int	ID of the highest leveled weapon mastery at the end of the match.
bestWeaponLevel	int	Level of the highest leveled weapon mastery at the end of the match.
masteryLevel	<int, int>	Dictionary of mastery ID paired with the level of the mastery at the end of the match. <Mastery code, Mastery level> See Details.
equipment	<int, int>	Dictionary of itemID equipped at the end of the match. <Equipment slot, Item code> See Details.
versionMajor	int	Main version of the game.
versionMinor	int	Minor version of the game.
language	string	Language of the current user.
skillLevelInfo	<int, int>	Dictionary of skill ID and level at the end of the match. <Skill code, Skill level>See Details.
skillOrderInfo	<int, int>	Dictionary of skill level up order history during the match. <Order, Skill code>See Details.
serverName	string	Location of the battle server.
maxHp	int	Stats. Maximum health.

maxSp	int	Stats. Maximum stamina.
attackPower	int	Stats. Attack power.
moveSpeed	float	Stats. Movement speed.
defense	int	Stats. Defense.
hpRegen	float	Stats. Health regeneration.
spRegen	float	Stats. Stamina regeneration.
attackSpeed	float	Stats. Attack speed.
outOfCombatMoveSpeed	float	Stats. Movement speed out of combat.
sightRange	float	Stats. Sight range.
attackRange	float	Stats. Basic attack range.
criticalStrikeChance	float	Stats. Additional critical strike chance (Value in 0 ~ 100 ~ and more).
criticalStrikeDamage	float	Stats. Additional critical damage multiplier (Value in 0 ~ 100 ~ and more).
coolDownReduction	float	Stats. Cooldown reduction. (Value in 0 ~ 100 ~ and more)
lifeSteal	float	Stats. Omnisyphon.
normalLifeSteal	float	Stats. Life steal by normal type damage.
skillLifeSteal	float	Stats. Life steal by skill type damage.
amplifierToMonster	float	Stats. Damage amplifier to monster.
trapDamage	float	Stats. Trap damage amplifier.
gainExp	int	Experience points gained after match to the user's account.
startDtm	DateTime	Server time of when the match started.
duration	int	Server frame time at the end of the match.
mmrBefore	int	MMR of the user prior to the match.
mmrGain	int	Delta MMR of the match.
mmrAfter	int	New MMR of the user.
playTime	int	Seconds elapsed until the end of match for the user.
watchTime	int	Seconds elapsed spectating.
totalTime	int	Sum of playtime and watchtime.
botAdded	int	Number of AI bots added in the match.
botRemain	int	Number of AI bots remaining at the end of the match for the user.
restrictedAreaAccelerated	int	Number of time restricted area acceleration took place.
safeAreas	Int	Remaining areas at the end of the match for the user.
teamNumber	int	Team number of the user.
preMade	int	Number of members in the team. This value does not include users matched via the team finder.
eventMissionResult	<int, int>	Dictionary of event mission ID and total count of the objective. <Event code, Count>
~~gainedNormalMmrKFactor~~	~~float~~	~~Certainty of given MMR of the user.~~ (No longer supported.)

victory	int	Boolean for victory.
craftUncommon	int	Number of crafted uncommon grade items.
craftRare	int	Number of crafted rare grade items.
craftEpic	int	Number of crafted epic grade items.
craftLegend	int	Number of crafted legendary grade items.
damageToPlayer	int	Total damage dealt to other player characters by the user.
damageToPlayer_trap	int	Trap damage dealt to other player characters by the user.
damageToPlayer_basic	int	Auto Attack type damage dealt to other player characters by the user.
damageToPlayer_skill	int	Skill type damage dealt to other player characters by the user.
damageToPlayer_itemSkill	int	Item skill type damage dealt to other player characters by the user.
damageToPlayer_direct	int	Direct type damage dealt to other player characters by the user.
damageToPlayer_uniqueSkill		Unique skill type damage dealt to other player characters by the user.
damageFromPlayer	int	Total damage received from other player characters to the user.
damageFromPlayer_trap	int	Trap damage received from other player characters to the user.
damageFromPlayer_basic	int	Auto Attack type damage received from other player characters to the user.
damageFromPlayer_skill	int	Skill type damage received from other player characters to the user.
damageFromPlayer_itemSkill	int	Item skill type damage received from other player characters to the user.
damageFromPlayer_direct	int	Direct type damage received from other player characters to the user.
damageFromPlayer_uniqueSkill	int	Unique skill type damage received from other play characters to the user.
damageToMonster	int	Total damage dealt to monsters by the user.
damageToMonster_trap	int	Trap damage dealt to monsters by the user.
damageToMonster_basic	int	Normal damage dealt to monsters by the user.
damageToMonster_skill	int	Skill damage dealt to monsters by the user.
damageToMonster_itemSkill	int	Unique damage dealt to monsters by the user.
damageToMonster_direct	int	Direct damage dealt to monsters by the user.
damageToMonster_uniqueSkill	int	Unique skill damage dealt to monsters by the user.
damageFromMonster	int	Total damage received from monster to the user.
damageToPlayer_Shield	int	Damage dealt by the user to other player character's shield.

damageOffsetedByShield_Player	int	Damage from other player characters the user absorbed by shield.
damageOffsetedByShield_Monster	int	Damage from other wild animals the user absorbed by shield.
killMonsters	<int, int>	Dictionary of monster ID and number kills by the user. See Details.
healAmount	int	Total amount healed by the user. (Disregard regeneration)
teamRecover	int	Total amount of heal given to other player character.
protectAbsorb	int	Damage protected by shield.
addSurveillanceCamera	int	Number of surveillance camera installed by the user.
addTelephotoCamera	int	Number of telephoto camera installed by the user. (Nathapon's trail camera is considered as a type of telephoto camera.)
removeSurveillanceCamera	int	Number of surveillance camera destroyed by the user.
removeTelephotoCamera	int	Number of telephoto camera destroyed by the user. (Nathapon's trail camera is considered as a type of telephoto camera.)
useHyperLoop	int	Number of times the user used the Hyperloop.
useSecurityConsole	int	Number of times the user used the security console.
giveUp	int	Boolean for giving up.
teamSpectator	int	Boolean value whether the user's spectating the match.
routeIdOfStart	int	Route ID selected at the start of the match by the user.
routeSlotId	int	Route slot ID of the selected route.
placeOfStart	int	Starting area selected by the user. See table.
mmrAvg	int	MMR average of the team.
teamKill	int	Number of kills scored by the team at the end of the match for the user.
accountLevel	int	Current account level of the user.
killerUserNum	int	Unique number identifier of the user's killer. Variations of total 3 killers in accordance to the team mode. (If the killed is a playing character.)
killer	string	Identity of the killer. Variations of total 3 killers in accordance to the team mode. See Details.
killDetail	string	Nickname of the user's killer. If the killer is restriction timer, this will return the current area name. Variations of total 3 killers in accordance to the team mode.
killerCharacter	string	Character name of the user's killer. Variations of total 3 killers in accordance to the team mode.
killerWeapon	string	Weapon of the user's killer. Variations of total 3 killers in accordance to the team mode.
causeOfDeath	string	- Name of the skill or object that caused the

		death of the user. See Details.
placeOfDeath	string	Area ID of the area user died in.
fishingCount	int	Number of fishing done during the match by user.
useEmoticonCount	int	Number of Emotes used during the match by the user.
traitFirstCore	int	Code of the first core augments.
traitFirstSub	int[2]	Code of the first two sub slot augments.
traitSecondSub	int[2]	Code of the second two sub slot augments.
totalTurbineTakeover	Int	[COBALT ONLY] Total times the player took over an amplifier.
StartingItems	Int[7]	[COBALT ONLY] First set of items acquired by the player. Usually resorts to 6 items, with few exceptions leading up to 7 items.
usedNormalHealPack	Int	[COBALT ONLY] Amount of normal heal packs used. (player)
usedReinforcedHealPack	Int	[COBALT ONLY] Amount of reinforced heal packs used. (player)
usedNormalShiedPack	Int	[COBALT ONLY] Amount of normal shield packs used. (player)
usedReinforcedShieldPack	Int	[COBALT ONLY] Amount of reinforced shield packs used. (player)
totalVFCredit	Int[20]	The amount of credit acquired by the player. Amount is separately accumulated each minute into a different index.
creditSource	<str, int>	Accumulated amount of credit sorted by the source. See Details.
usedVFCredit	Int[20]	The amount of credit used by the player. Amount is separately accumulated each minute into a different index.
boughtInfusion	<int, int>	[COBALT ONLY] Dictionary of infusion product and how many times they were bought by the player. <InfusionProductCode , amount>
itemTransferredConsole	Int[]	[LUMIA ONLY] Itemcode of Items requested by the player via transfer console.
itemTransferredDrone	Int[]	[LUMIA ONLY] Itemcode of items requested by the player vis transfer drone.
finalInfusion	Int[3]	[COBALT ONLY] Final 3 trait infusions acquired by the player.
craftMythic	Int	Number of crafted mythic grade items.
playerDeaths	Int	Amount of deaths experienced by the player.
killGamma	Bool	Boolean for killing gamma. (Killer only)
scoredPoint	Int[20]	[COBALT ONLY] Score acquired by the player. Scores are seperatebly accumulated each minute into a

		differend index.
killDetails	<int, int>	Dictionary of characters killed by the player. <CharacterCode, amount>
deathDetails	<int, int>	Dictionary of characters who killed the player. <CharacterCode, amount>
killsPhaseOne	Int	[COBALT ONLY] Amount of kills accumulated by the player in Phase 1.
killsPhaseTwo	Int	[COBALT ONLY] Amount of kills accumulated by the player in Phase 2.
killsPhaseThree	Int	[COBALT ONLY] Amount of kills accumulated by the player in Phase 3.
deathsPhaseOne	Int	[COBALT ONLY] Amount of deaths accumulated by the player in Phase 1.
deathsPhaseTwo	Int	[COBALT ONLY] Amount of deaths accumulated by the player in Phase 2.
deathsPhaseThree	Int	[COBALT ONLY] Amount of deaths accumulated by the player in Phase 2.
ccTimeToPlayer	float	Total amount of CC time achieved by the player. (To other players)
foodCraftCount	<int, int>	Total amount of foods cooked by grade. <Grade, amount>
beverageCraftCount	<int, int>	Total amount of beverage carfted by grade. <Grade, amount>
airSupplyOpenCount	<int, int>	Total amount of air supply opened <SupplyGrade, amount>
~~afk~~	~~Bool~~	Boolean for being afk in Cobalt Protocol Deprecated. **(DATA WILL NOT BE SENT)**
escapeState	int	1: failed to escape due to reason other than 2 2: failed to escape due to enemy 3: escaped success.
collectItemForLog	Int[]	collectItemForLog[CollectibleCode] = count of collectibles collected. See detail.
equipFirstItemForLog	<int, int[]>	Item code of the top tier weapon. <equipment slot, itemCode>. See detail. The value in the int list will be refreshed into the top tier item equipped by the character.
totalDoubleKill	Int	Total amounts of time the user achieved a double kill.
totalTripleKill	Int	Total amounts of time the user achieved a triple kill.
totalQuadraKill	Int	Total amounts of time the user achieved a quadra kill.
totalExtraKill	int	Total amounts of time the user achieved more than 5 consecutive kills.
battleZone1AreaCode	int	The area code for the first battlezone. See details.
battleZone1BattleMark	int	The mark of the battlezone.

battleZone1ItemCode	int[]	The first item from the battlezone.
battleZone2AreaCode	int	Deprecated.
battleZone2BattleMark	int	Deprecated.
battleZone2ItemCode	int[]	Deprecated.
battleZone3AreaCode	int	Deprecated.
battleZone3BattleMark	int	Deprecated.
battleZone3ItemCode	int[]	Deprecated.
battleZonePlayerKillCount	int	Deprecated.
battleZonePlayerDeathCount	int	Deprecated.
battleZone1Winner	int	Victory - 1 Defeat - 0
battleZone2Winner	int	Deprecated.
battleZone3Winner	int	Deprecated.
battleZone1BattleMarkCount	int	Number of competitors that entered the battlezone. (Including the player)
battleZone2BattleMarkCount	int	Deprecated.
battleZone3BattleMarkCount	int	Deprecated.
tacticalSkillGroup	int	The group code for the tactical skill. See details.
tacticalSkillLevel	int	The final level of the tactical skill.
totalGainVFCredit	int	Totally gained credits.
killPlayerGainVFCredit	int	Credits gained by killing another players.
killChickenGainVFCredit	int	Credits gained by killing chickens.
killBoarGainVFCredit	int	Credits gained by killing boars.
killWildDogGainVFCredit	int	Credits gained by killing wild dogs.
killWolfGainVFCredit	int	Credits gained by killing wolves.
killBearGainVFCredit	int	Credits gained by killing bears.
killOmegaGainVFCredit	int	Credits gained by killing Omega.
killBatGainVFCredit	int	Credits gained by killing bats.
killWicklineGainVFCredit	int	Credits gained by killing Wickline.
killAlphaGainVFCredit	int	Credits gained by killing Alphas.
killItemBountyGainVFCredit	int	Credits gained by bounty from killing players.
killDroneGainVFCredit	int	[Cobalt] Credits gained by killing drones.
killGammaGainVFCredit	int	[Cobalt] Credits gained by killing Gamma.
killTurretGainVFCredit	int	[Cobalt] Credits gained by killing the Turret. (This cannot be done)
itemShredderGainVFCredit	int	[Cobalt] Credits gained by selling items.
totalUseVFCredit	int	Credits used by player.
remoteDroneUseVFCreditMySelf	int	Credits used by the user for remote drone transfer. (For self-use)
remoteDroneUseVFCreditAlly	int	Credits used by the user for remote drone transfer. (to ally)
transferConsoleFromMaterialUseVFCredit	int	Credits used in transfer consoles. Excludes rootkey that is used for escape.

transferConsoleFromEscapeKeyUseVFCredit	int	Credits used in transfer consoles for rootkeys.
transferConsoleFromRevivalUseVFCredit	int	Credits used in transfer consoles for revival.
tacticalSkillUpgradeUseVFCredit	int	Credits used to level up tactical skills.
infusionReRollUseVFCredit	int	[Cobalt] Credits used for rerolling infusions.
infusionTraitUseVFCredit	int	[Cobalt] Credits used to buy traits in infusions.
infusionRelicUseVFCredit	int	[Cobalt] Credits used to buy relics in infusions.
infusionStoreUseVFCredit	int	[Cobalt] Credits used to buy items in infusions.
teamElimination	int	Number of times the team eliminated the enemy.
teamDown	int	Number of times the team downed the enemy.
teamBattleZoneDown	int	Number of times the team downed the enemy in battlezone.
teamRepeatDown	int	Number of times the team repeatedly downed the enemy.
adaptiveForce	int	Stats. The final adaptive force of the player.
adaptiveForceAttack	int	The additional attack power stat adapted via the adaptive force.
adaptiveForceAmplify	int	The additional skill amplification stat adapted via the adaptive force.
skillAmp	int	Stats. The final skill amplification of the player.
campFireCraftUncommon	int	Number of uncommon grade food crafted in the bonfire.
campFireCraftRare	int	Number of rare grade food crafted in the bonfire.
campFireCraftEpic	int	Number of epic grade food crafted in the bonfire.
campFireCraftLegendary	int	Number of legendary grade food crafted in the bonfire.
cobaltRandomPickRemoveCharacter	int	[Cobalt] character code of the character removed from the pool by the player.
tacticalSkillUseCount	int	Number of times player used the tactical skill.
creditRevivalCount	int	Number of times the player was revived via the transfer console.
creditRevivedOthersCount	int	Number of times the playe revived allys via the transfer console.
timeSpentInBriefingRoom	int	Time spent in the briefing room.
IsLeavingBeforeCreditRevivalTerminate	bool	Whether the player left the game before credit resurrection was unavailable.
crGetAnimal	int	Total credits gained by killing wild animals.
crGetMutant	int	Total credits gained by killing mutant animals.
crGetPhaseStart	int	Total credits gained by phase change.
crGetKill	int	Total credits gained by killing other players.
crGetAssist	int	Total credits gained by assisting teammates in killing other players.
crGetTimeElapsed	int	Total credits gained by elapsed time.
crGetCreditBonus	int	Total credits that was awarded to the player for having

		the least credits.
crUseRemoteDrone	int	Total credits used by remote drones.
crUseUpgradeTacticalSkill	int	Total credits used to upgrade tactical skills.
crUseTreeOfLife	int	Total credits used to buy Tree of Life.
crUseMythril	int	Total credits used to buy Mythirl.
crUseForceCore	int	Total credits used to buy Force cores.
crUseVFBloodSample	int	Total credits used to buy VF blood samples.
crUseRootkit	int	Total credits used to buy rootkits.
mmrGainInGame	int	Gain of rank point in the game.
mmrLossEntryCost	int	Loss of rank point due to entry of the game.
premadeMatchingType	int	Type of matching used in the game. 0 : all premade teams composed of 1, 2 or 3 members matched. 1 : only premade teams composed of 1 or 2 members are matched. 2 : only premade teams composed of 3 members are matched.
viewContribution	int	Point total of view contribution.
useReconDrone	int	Times the user used the Recon Drones
useEmpDrone	int	Times the user used the EMP Drones
exceptPreMadeTeam	int	User refused to match with duo premade teams.
terminateCount	int	Count of teams terminated by the user's team.
clutchCount	int	Count of clutch scored by the user.
unknownKill	int	Count of downed enemies that has not been finished. (for tournament purpose.)
mainWeather	int	The main weather feature in this game. See details.
subWeather	int	The sub weather feature in this game. See details.
activeInstallation	<int, int>	Count of used installations in the game. Can be empty. See details.
useGuideRobot	Int	Number of times user channeled to the guide robot.
guideRobotRadial	Int	Credits used to buy radial grade items in the guide robot.
guideRobotFlagShip	Int	Credits used to buy flagship grade items in the guide robot.
guideRobotSignature	Int	Credits used to buy signature grade items in the guide robot.
crGetByGuideRobot	Int	Credits acquired by the guide robot in battle mode.
damageToGuideRobot	int	Damage dealt to the guide robot
getBuffCubeRed	Int	Count of sanguine cubes collected.
getBuffCubePurple	int	Count of galaxy cubes collected.
getBuffCubeGreen	int	Count of essence cubes collected.
getBuffCubeGold	int	Count of prosperity cubes collected.
getBuffCubeSkyBlue	int	Count of zephyr cubes collected.

sumGetBuffCube	int	Sum total of all cubes collected.
squadRumbleRank	int	[Squad Rumble Only] Current squad rumble rank.
teamDownCanNotEliminate	int	Number of kills during the ineliminable phase. (Currently day 2)
teamDownCanEliminate	int	Number of kills when not in the ineliminable phase. (Currently day 2)
teamRepeatDownCanNotEliminate	int	Number of repeated kills during the ineliminable phase. (Currently day 2)
teamRepeatDownCanEliminate	int	Number of repeated kills when not in the ineliminable phase. (Currently day 2)
teamDownInAutoResurrection	Int	**(**Deprecation Warning) This value is only available in 1.43 ~ 1.44 versions.** Number of kills during the ineliminable phase.
teamDownDeactiveAutoResurrection	Int	**(**Deprecation Warning) This value is only available in 1.43 ~ 1.44 versions.** Number of kills when not in the ineliminable phase.
teamRepeatDownInAutoResurrection	Int	**(**Deprecation Warning) This value is only available in 1.43 ~ 1.44 versions.** Number of repeated kills during the ineliminable phase.
teamRepeatDownDeactiveAutoResurrection	Int	**(**Deprecation Warning) This value is only available in 1.43 ~ 1.44 versions.** Number of repeated kills when not in the ineliminable phase.

Get Game Data Table

This model is used to acquire in game data tables.

Model URL

/v1/data/{metaType}

Required Field

metaType

In order to get all the available data tables, use 'hash' as the input for the metaType.

Data Example

This model will not feature a data example.

Get Language Data

This model will fetch the text file including all language data of the game.

Model URL

v1/l10n/{langauge}

Required Field

Language

Fully provided languages :

- Korean
- English
- Japanese
- ChineseSimplified
- ChineseTraditional

Pratially provided languages :

- French
- Spanish
- SpanishLatin
- Portuguese
- PortugueseLatin
- Indonesian
- German

- Russian
- Thai
- Vietnamese

Data Example

The model will provide a text file link.

```
{
  "code": 200,
  "message": "Success",
  "data": {
    "l10Path": "https://d1wkxvul68bth9.cloudfront.net/l10n/l10n-Korean-20211117071605.txt"
  }
}
```

Get Route Data

Retrieve the latest saved plans or any specific saved plan.

Retrieving recommended saved plans will return maximum 100 results of the most recently updated saved plans in the last 90 days.

Model URL

v1/weaponRoutes/recommend

v1/weaponRoutes/recommend/{routeId}

Required Field

RouteId

The “id” field in the recommendedWeaponRoute.

Data Example

The model will either provide the list of recommendedWeaponRoute or a single one.

```
{
"code": 200,
"message": "Success",
"result": [
  {
    "recommendWeaponRoute": {
      "id": 93,
      "title": "TestRoute₩t",
      "userNum": 25858,
      "userNickname": "TranslateTestKR",
      "characterCode": 76,
      "slotId": 0,
      "weaponType": 3,
      "weaponCodes": "[104407,202408,201417,205405,204418]",
      "traitCodes": "[7000401,7011001,7010311,7110701,7110601]",
      "lateGameItemCodes": "{₩"1₩":[108502,202509,201415,203502,204508]}",
      "remoteTransferItemCodes": "[]",
      "tacticalSkillGroupCode": 130,
      "paths": "[170,180,30]",
      "count": 0,
      "version": "1.42.0",
      "teamMode": 0,
      "languageCode": "ko",
      "routeVersion": 2,
      "share": true,
      "updateDtm": 1738553481000,
      "v2Like": 0,
      "v2WinRate": 0,
      "v2SeasonId": 0,
      "v2AccumulateLike": 0,
      "v2AccumulateWinRate": 0,
      "v2AccumulateSeasonId": 0
    },
    "recommendWeaponRouteDesc": {
      "recommendWeaponRouteId": 93,
      "skillPath": "q,w,w,q,q,w,q,w,q,w,e,r,r,e,t,r,e,e,t,e",
      "desc": "Sample Test Route"
    }
  }
  ]
}
```

Data Definitions

RecommendWeaponRoute

Key	DataType	Description
id	int	Unique identifier of the WeaponRoute
title	string	Title given by the user.
userNum	int	Unique identifier for the user.
userNickname	Str	Nickname of the user.
characterCode	Int	Code of the character that uses this route.
slotId	Int	Slot Id according to the maker.
weaponType	Int	Weapon mastery. See Details.
weaponCodes	Str	Initial target items designated to the weapon route. Array of int itemCode values stringified. e.g. {“weaponCodes” : “[100242, 100420, 402040, 100302, 104020]”}
traitCodes	Str	Initial traits designated to the weapon route. See Details. Array of int traitCode values stringified. e.g. {“traitCodes” : “[7001020, 7002040]”}
lateGameItemCodes	Str	Late game items designated to the weapon route. A map consisting arrays of itemCode values stringified. The “0” key is the automated late game items. Any other set of late game items will be a stringified integer like “1” or “2”.
remoteTransferItemCodes	Str	Items that must be acquired through the remote drone. Array of int itemCode values stringified.

		{“remoteTransferItemCodes” : “[100202, 100420]”}
tacticalSkillGroupCode	Int	The tactical skill designated to the weapon route. See Details.
paths	Str	Areas the route recommends. See Details.
count	Int	Number of times this weapon route has been shared to others.
version	Str	The client version.
teamMode	Int	The matchingTeamMode. Currently only uses 0.
languageCode	Str	Language code for the route.
routeVersion	Int	The version info of the route.
share	Bool	Whether the route is shared to others.
updateDtm	Int	Last updated time. Epoch Time.
v2Like	Int	Amount of likes received this season.
v2WinRate	int	Win rate of users using this route this season.
v2SeasonId	Int	Current season number.
v2AccumulateLike	Int	Accumulated likes.
v2AccumulateWinRate	Int	Accumulated win rate.
v2AccumulateSeasonId	Int	Not used.

RecommendWeaponRouteDesc

Key	DataType	Description
recommendedWeaponRouteId	int	Identifier for the route.
skillPath	string	Skill order designated to the weapon route in csv. “q, q, w, e, r, … “
desc	string	***This key is only included when the route has a description.** Description of how the weapon route should be handled.

Additional Data

Some data cannot be acquired using the data table.

Following are data used in the game. **(Last updated : 2022/11/07)**

Mastery

Following are the mastery codes.

- 0 : None

- 1 : Glove
- 2 : Tonfa
- 3 : Bat
- 4 : Whip
- 5 : HighAngleFire
- 6 : DirectFire
- 7 : Bow
- 8 : CrossBow
- 9 : Pistol
- 10 : AssaultRifle
- 11 : SniperRifle
- 13 : Hammer
- 14 : Axe
- 15 : OneHandSword
- 16 : TwoHandSword
- 17 : Polearm
- 18 : DualSword
- 19 : Spear
- 20 : Nunchaku
- 21 : Rapier
- 22 : Guitar
- 23 : Camera
- 24 : Arcana
- 25 : VFArm
- 101 : Craft
- 102 : Search
- 103 : Move
- 201 : Defense
- 202 : Hunt

Area

The code of an Area can be matched to the L10N data using this table.

AreaKey	L10N Key
10	Area/Name/Harbor
20	Area/Name/Warehouse
30	Area/Name/Pond
40	Area/Name/Stream
50	Area/Name/SandyBeach
60	Area/Name/Uptown
70	Area/Name/Alley
80	Area/Name/GasStation
90	Area/Name/Hotel
100	Area/Name/PoliceStation
110	Area/Name/FireStation
120	Area/Name/Hospital
130	Area/Name/Temple
140	Area/Name/Archery
150	Area/Name/Cemetery
160	Area/Name/Forest
170	Area/Name/Factory
180	Area/Name/Church
190	Area/Name/School

Skill

Skill names can be found in the language data as

Skill/Group/Name/{SkillGroup}

Equipment

Equipment slots are designed as followed.

- 0 : Weapon

- 1 : Chest
- 2 : Head
- 3 : Arm
- 4 : Leg
- ~~5 : Trinket~~ (No longer used.)

Monster

You can find the data name of each monster using the language data.

Monster/Name/{Code}

Killer

Following are values that you can get from killer.

- player : Other player character
- wildAnimal : Wild life including epic monsters like Wickeline.
- restrictedArea : Death due to restriction timers.

Cause Of Death

Casue of death will return a Korean value.

In order to decode this value into your own language, it must be referenced through the follwing data table.

(Exception : When the cause of death is due to Auto Attacks, it will return 'basicAttack'.)

v1/data/SkillGroup

- Reference the data to the 'Name' column.
- Acquire the 'Code' column.
- Search v1/l10n/{langauge} for SummonData/Name/{Code}

v1/data/SummonObject

- Reference the data to the 'Name' column.
- Acquire the 'SkillGroup' column.
- Search v1/l10n/{langauge} for Skill/Group/Name/{SkillGroup}

Collectible

Each collectible sources will have a unique code.

- 1: Water
- 2: Stick
- 3: Stone
- 4: Fishing Point (Sea)
- 5: Fishing Point (Fresh Water)
- 6: Potato
- 7: Tree of Life
- 8: Meteor
- 9: Flower

Trait

The name of each trait can be decoded using the L10N data.

Trait/Name/{code }

TacticalSkill

Following are the currently used tactical skills.

GroupCode	Search Key
30	Skill/Group/Name/4000000
40	Skill/Group/Name/4001000

50	Skill/Group/Name/4101000
60	Skill/Group/Name/4102000
70	Skill/Group/Name/4103000
80	Skill/Group/Name/4104000
90	Skill/Group/Name/4105000
110	Skill/Group/Name/4107000
120	Skill/Group/Name/4110000
130	Skill/Group/Name/4112000
140	Skill/Group/Name/4113000
150	Skill/Group/Name/4108000
500010	Skill/Group/Name/4501000
500020	Skill/Group/Name/4502000
500030	Skill/Group/Name/4503000
500040	Skill/Group/Name/4504000
500050	Skill/Group/Name/4505000
500060	Skill/Group/Name/4506000
500070	Skill/Group/Name/4507000
500080	Skill/Group/Name/4508000
500090	Skill/Group/Name/4509000
500100	Skill/Group/Name/4510000
500110	Skill/Group/Name/4511000
500120	Skill/Group/Name/4000000
500130	Skill/Group/Name/4001000
500140	Skill/Group/Name/4101000
500150	Skill/Group/Name/4102000
500160	Skill/Group/Name/4103000
500170	Skill/Group/Name/4104000
500180	Skill/Group/Name/4105000
500190	Skill/Group/Name/4107000
500200	Skill/Group/Name/4110000
500210	Skill/Group/Name/4112000
500220	Skill/Group/Name/4113000
500230	Skill/Group/Name/4108000

Weather

The name of the weather can be found using the L10N data.

Weather/Name/{code}

Installation

The name of the installation can be found using the L10N data.

Installation/Name/{code}