

# KittyWar Game Server Protocol

FLAG 1-BYTE

TOKEN 24-BYTES

DATASIZE 3-BYTES

REQUEST HEADER SIZE: 28 BYTES

RESPONSE HEADER SIZE: 4 BYTES

## *General Session Flags*

### LOGIN

---



**# Logs a user into game server**

Request: FLAG = 0

DATASIZE = sizeof username

HEADER > FLAG + TOKEN + DATASIZE

BODY > username

Response: FLAG = 0

DATASIZE = 1 byte

HEADER > FLAG + DATASIZE

BODY > 0 or 1

# 0 for login denied

# 1 for login accepted

### LOGOUT

---



**# Logs a user out of game server**

Request: FLAG = 1

DATASIZE = 0 bytes

HEADER > FLAG + TOKEN + DATASIZE

Response: FLAG = 1

```
DATASIZE = 1 byte
HEADER > FLAG + DATASIZE
BODY    > 0 or 1
# 0 for logout issues - force logout (debugging)
# 1 for logout successful
```

## **FIND MATCH**

```
# Places player into a queue, server will not respond until
match is found
```

Request: FLAG = 2

```
DATASIZE = 0 bytes
HEADER > FLAG + TOKEN + DATASIZE
```

Response: FLAG = 2

```
DATASIZE = 0 bytes
HEADER > FLAG + DATASIZE
# Indicates that a match is found and starting very soon
```

## **USER PROFILE**

```
# Pulls all user information, cat cards, stats, etc...
```

Request: FLAG = 3

```
DATASIZE = 0 bytes
HEADER > FLAG + TOKEN + DATASIZE
```

Response: FLAG = 3

```
DATASIZE = sizeof user_profile
HEADER > FLAG + DATASIZE
BODY    > user_profile
# format: JSON
```

## **ALL CARDS**

**# Pulls all card data and sends back to the user**

Request: FLAG = 4

```
DATASIZE = sizeof card_data
HEADER > FLAG + TOKEN + DATASIZE
```

Response: FLAG = 4

```
DATASIZE = sizeof card_data
HEADER > FLAG + DATASIZE
BODY    > card_data
# format: JSON
```

## CAT CARDS

---

**# Pulls all cat card data and sends back to the user**

Request: FLAG = 5

```
DATASIZE = sizeof card_data
HEADER > FLAG + TOKEN + DATASIZE
```

Response: FLAG = 5

```
DATASIZE = sizeof card_data
HEADER > FLAG + DATASIZE
BODY    > card_data
# format: JSON
```

## BASIC CARDS

---

**# Pulls all basic card data and sends back to the user**

Request: FLAG = 6

```
DATASIZE = 0 bytes
HEADER > FLAG + TOKEN + DATASIZE
```

Response: FLAG = 6

```
DATASIZE = sizeof card_data
HEADER > FLAG + DATASIZE
BODY    > card_data
# format: JSON
```

## **CHANCE CARDS**

---

**# Pulls all chance card data and sends back to the user**

**Request: FLAG = 7**

```
DATASIZE = sizeof card_data
HEADER > FLAG + TOKEN + DATASIZE
```

**Response: FLAG = 7**

```
DATASIZE = sizeof card_data
HEADER > FLAG + DATASIZE
BODY    > card_data
# format: JSON
```

## **ABILITY CARDS**

---

**# Pulls all ability card data and sends back to the user**

**Request: FLAG = 8**

```
DATASIZE = sizeof card_data
HEADER > FLAG + TOKEN + DATASIZE
```

**Response: FLAG = 8**

```
DATASIZE = sizeof card_data
HEADER > FLAG + DATASIZE
BODY    > card_data
# format: JSON
```

## ***MATCH FLAGS***

**END MATCH**

---

**# Notifies the user that the match is over**  
**# This flag is only used by the server to send to the client**

**Response: FLAG = 9**

DATASIZE = 1 byte  
HEADER > FLAG + DATASIZE  
BODY > 0 - 3  
# 0 you lost the match  
# 1 you won the match  
# 2 the match is a draw  
# 3 the match ended because of errors no winner

#### **NEXT PHASE**

---

**# Notifies the user that the next phase is starting**  
**# This flag is only used by the server to send to the client**

**Response: FLAG = 98**

DATASIZE = 0 bytes  
HEADER > FLAG + DATASIZE

#### **READY**

---

**# Client should send this when they are done with the phase**  
**# The server will respond with the NEXT PHASE flag only when both players are ready**

**Request: FLAG = 99**

DATASIZE = 0 bytes  
HEADER > FLAG + TOKEN + DATASIZE

**Response: FLAG = 98**

DATASIZE = 0 bytes  
HEADER > FLAG + DATASIZE

## **SELECT CAT**

**# Used for when the player selects a cat during the setup phase**

**Request: FLAG = 100**

DATASIZE = sizeof cat\_id(should be 1 byte)  
HEADER > FLAG + TOKEN + DATASIZE  
BODY > cat\_id

**Response: FLAG = 100**

DATASIZE = 1 byte  
HEADER > FLAG + DATASIZE  
BODY > 1  
# Response is a 1 if successful  
# Only time it would not be successful is if the player tries to  
# select a cat they do not have in which case the match will end  
# with an error

## **OPPONENT SELECTED CAT**

**# Tell the player the opponent's selected cat**  
**# This flag is only used by the server to send to the client**

**Response: FLAG = 49**

DATASIZE = size cat\_id  
HEADER > FLAG + DATASIZE  
BODY > cat\_id

## **USE ABILITY**

**# Used when the player wants use an active ability**  
**# (PRELUDE & POSTLUDE)**

**Request: FLAG = 101**

DATASIZE = sizeof ability\_id(should be 1 byte)  
HEADER > FLAG + TOKEN + DATASIZE

BODY > ability\_id

Response: FLAG = 101

DATASIZE = 1 byte

HEADER > FLAG + DATASIZE

BODY > 0 or 1

# 0 - The ability is on cooldown you can not use or did not meet  
# the requirements

# 1 - The ability was used successfully

# This response will be accompanied by another packet signifying  
what the ability did EX. HEALTH GAINED packet

#### **GAIN HP**

**█**

**# Notifies the player they have gained health**

**# This flag is only used by the server to send to the client**

Response: FLAG = 50

DATASIZE = sizeof health\_gained

HEADER > FLAG + DATASIZE

BODY > health\_gained

#### **OPPONENT GAIN HP**

**█**

**# Notifies the player the opponent has gained health**

**# This flag is only used by the server to send to the client**

Response: FLAG = 51

DATASIZE = sizeof health\_gained

HEADER > FLAG + DATASIZE

BODY > health\_gained

#### **DAMAGE MODIFIED**

**█**

**# Notifies the player their damage multiplier has changed**

**# This flag is only used by the server to send to the client**

Response: FLAG = 52

```
DATASIZE = sizeof multiplier
HEADER > FLAG + DATASIZE
BODY    > multiplier
# multiplier is how much your current damage is multiplied by
# EX. 2(x2) or 4(x4)
```

#### OPPONENT DAMAGE MODIFIED

# Notifies the player the opponent damage multiplier has changed  
# This flag is only used by the server to send to the client

Response: FLAG = 53

```
DATASIZE = sizeof multiplier
HEADER > FLAG + DATASIZE
BODY    > multiplier
# multiplier is how much your current damage is multiplied by
# EX. 2(x2) or 4(x4)
```

#### GAIN CHANCE

# Notifies the player that they have gained a chance card  
# This flag is only used by the server to send to the client

Response: FLAG = 54

```
DATASIZE = sizeof chance_id
HEADER > FLAG + DATASIZE
BODY    > chance_id
```

#### OPPONENT GAIN CHANCE

# Notifies the player the opponent has gained a chance card  
# This flag is only used by the server to send to the client

Response: FLAG = 55



```
DATASIZE = 0 bytes
HEADER > FLAG + DATASIZE
# Only notifies, does not reveal the chance card
```

#### **GAIN ABILITY**

```
# Tell player their random ability
# This flag is only used by the server to send to the client
```

Response: FLAG = 56

```
DATASIZE = sizeof ability_id
HEADER > FLAG + DATASIZE
BODY    > ability_id
```

#### **GAIN CHANCES**

```
# Tell player their two random chance cards during post setup
# This flag is only used by the server to send to the client
```

Response: FLAG = 57

```
DATASIZE = 2 bytes
HEADER > FLAG + DATASIZE
BODY    > chance_card1 + chance_card2
# Each chance card is a byte
```

#### **SELECT MOVE**

```
# Used for when the player selects a move
```

##### **Available Moves:**

- 0 - PURR
- 1 - GUARD
- 2 - SCRATCH
- 3 - SKIP

Request: FLAG = 102

```
DATASIZE = sizeof move_id
```

HEADER > FLAG + DATASIZE  
BODY > move\_id

Response: FLAG = 102

DATASIZE = 1 byte  
HEADER > FLAG + DATASIZE  
BODY > 0 or 1  
# 0 - move could not be selected  
# 1 - move successfully selected

#### **SELECT CHANCE**

# Used for when the player selects a chance card

Request: FLAG = 103

DATASIZE = sizeof chace\_id  
HEADER > FLAG + DATASIZE  
BODY > chance\_id

Response: FLAG = 103

DATASIZE = 1 byte  
HEADER > FLAG + DATASIZE  
BODY > 0 or 1  
# 0 - chance could not be selected  
# 1 - chance successfully selected



#### **REVEAL MOVE**

# Used during show cards phase  
# Let's player and opponent know what move they selected

Response: FLAG = 58

DATASIZE = 1 bytes  
HEADER > FLAG + DATASIZE  
BODY > move\_id

#### **REVEAL CHANCE**

```
# Used during show cards phase
```

```
# Let's player and opponent know what chance they selected
```

```
Response: FLAG = 59
```

```
DATASIZE = 1 bytes
```

```
HEADER > FLAG + DATASIZE
```

```
BODY    > chance_id
```