# Interfaces

## Client-Side Interfaces

/\*Interprets the generic-form message the Client passes to it into something game-specific that can be used by the game controllers/views.

\*Translates game data back to a generic format for use by the client elsewhere

\*/

public Interface GameInterface

{

//TODO: Define Generic methods here.

//make sure to follow the description of what the interface is supposed to do

}

/\*Interprets raw Socket.IO message data, sending the data to the Client in a standard generic game format (which can then be processed by the game block/GameInterface).

\*APIs to communicate with the server, by passing generic game data

\*/

public Interface GameCommInterface

{

//TODO: Define Generic methods here.

//make sure to follow the description of what the interface is supposed to do

}

/\*Interprets and sends raw Socket.IO data for client side chat; sending and receiving chat APIs.

\*/

public Interface ChatInterface

{

//TODO: Define Generic methods here.

//make sure to follow the description of what the interface is supposed to do

}

/\*Interface to easily store user/game data formats into a cache for easily retrieval at a later date

\*/

public Interface CacheInterface

{

//TODO: Define Generic methods here.

//make sure to follow the description of what the interface is supposed to do

}

## Server-Side Interfaces

/\* Interprets the generic-form message the Server passes to it into something game-specific that can be used by the particular game logic.

\*Translates game data back to a generic format for use by the server elsewhere.

\*/

public Interface GameLogicInterface

{

//TODO: Define Generic methods here.

//make sure to follow the description of what the interface is supposed to do

}

/\* Interprets raw Socket.IO message data, managing connected client sockets, and passing along data to the Server in a standard generic game format (which can then be processed by the game logic block/GameLogicInterface).

\*APIs to communicate with the connected clients, by passing generic game data

\*/

public Interface GameCommInterface

{

//TODO: Define Generic methods here.

//make sure to follow the description of what the interface is supposed to do

}

/\* Interprets and sends raw Socket.IO data for client side chat, managing each individual chatroom instance; directing chat messages to proper chatrooms, and allowing for server intervention.

\*/

public Interface ChatInterface

{

//TODO: Define Generic methods here.

//make sure to follow the description of what the interface is supposed to do

}

/\* Interface to easily store user/game/friend/chat data formats into a database, with support for the relational database and the flat file data format.

\*/

public Interface DatabaseInterface

{

//TODO: Define Generic methods here.

//make sure to follow the description of what the interface is supposed to do

}