| LISTENERS OF SERVER | | Parameters |
| --- | --- | --- |
| mobile-gameStart | Listens to execute the necessary action when the game starts |  |
| web-sendUsers | Listens to send the online user list back |  |
| mobile-sendSocketId | Listens to add the socketId to player on the online user list | email:string |
| mobile-selectAttack | Listens that the player has selected to attack |  |
| mobile-selectHeal | Listens that the player has selected to heal |  |
| mobile-selectCurse | Listens that the player has selected to curse |  |
| mobile-selectUsePotion | Listens that the player has selected to use a potion |  |
| mobile-attack | Listens that the player has execute the action attack | \_id:string(id of the target) |
| web-sendSocketId | Listens to web client socketId emission |  |
| mobile-setSelectPlayer | Listens to mobile selected player | \_id:string(id of the selected target) |
| mobile-deselectPotions | Listens if a potion has been deselected |  |
| web-attackAnimationEnd | Listens when attack animation finishes | {defender: defender.id, attacker: attacer.id} |
| mobile-gameReset | Listens to reset the battle values to start another battle |  |
| mobile-signIn | Listens to the player login | playerEmail, callBack(type \*signIn\*) |
| mobile-isGameCreated | Listens to respond if the game has started |  |
| mobile-getBattles | Listens to respond with the battles | callBack(response) |
| mobile-selectBattle | Listens to send the selected battle to web | battleId |
| mobile-createGame | Listens to the the battle creation | battleId |
| mobile-joinBattle | Listens to the player when joins to the battle | playerId |
| mobile-isGameStarted | Listens to the player when enters the app to check if the game already started to block the JOIN button in the app |  |

| EMITS OF SERVER | | Parameters |
| --- | --- | --- |
| connectedUsers | Emits an online user list (all the players data) to everyone. | ONLINE\_USERS: Player[] |
| web-sendUser | Emit player data to web client | player: Player |
| web-playerDisconnected | Emits the players name that has been disconnected | name:string |
| web-selectAttack | Emit that the player has selected to attack |  |
| web-selectHeal | Emit that the player has selected to heal |  |
| web-selectCurse | Emit that the player has selected to curse |  |
| web-selectUsePotion | Emit that the player has selected to use a potion |  |
| assign-turn | Emit the id of the player to perform the turn | \_id:string |
| web-setSelectedPlayer | Emit the id of the player that has been selected on mobile to web client | \_id:string |
| web-deselectPotions | Emit to web that the potions have been deselected |  |
| updatePlayer | Emit target player with the attributes updated | obj:{\_id:string, attr:Modifier, ,attack: attack[], isBetrayer: boolean |
| gameStart | Emits the start of the game to everyone  Emits when any player sends an emit with the event mobile-isGameStarted | —-------------------------------------- |
| send-timer | Emits the timers actual time left to end the turn | timer: number |
| removePlayer | Emit target player to remove from the state | \_id:string |
| gameEnd | Emits the end of the game if at least there is no player on one side | winner: string (the name of the winner side: kaotika, dravocar, draw) |
| mobile-insufficientPlayers | Emit to Mortimer that there is not the minimum 1 acolyte to start the game | —----------------------------------- |
| web-attackInformation | Emit to web, an object with percentages, die roll, modifiers target, and damage dealed | to decide |
| send-killedPlayer | Emits the killed player to everyone | \_id: string |
| web-attackInformation | Emits the attack data | [link](https://docs.google.com/document/d/1HOlQUxQmFYxPWD8V8hcH-qKyyPMFwuy5ePEqYmMPVUs/edit?usp=sharing) |
| web-currentRound | Emits the next currentRound info | round: number |
| gameReset | Emits the notification of game reset |  |
| web-turnFinished | Emits turn end when turn time becomes 0 or when the attacker disconnects. |  |
| isGameCreated | Emits if Mortimer o Villain has created the game | boolean |
| battles | Emits the list of the battles | Array of battle objects. |
| selectedBattle | Emits the selected battle data | battle data (to be decided) |
| isGameCreated | Emits when the game creates | true |
| web-joinedBattle | Emits to web when a player enters to the battle | playerId |

# 

**INTERFACES**

export type MobileSignInResponse =

| { status: 'OK'; player: Player }

| { status: 'FAILED'; error: string };

export type MobileJoinBattleResponse =

| { status: 'OK', joinBattle: boolean}

| { status: 'FAILED', error: string};