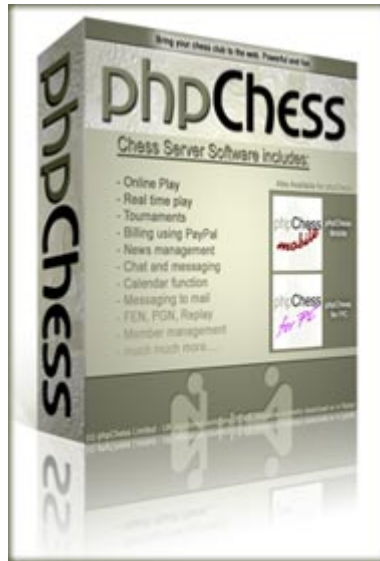


phpChess

platform independent access documentation



This document, all it's content, text, images and other information is the sole property of phpChess Limited. No unauthorised changes or amendments, extracts or publication is permitted without the prior consent of phpChess Limited.

You can contact phpChes Limited via our website (<http://www.phpchess.com>)

Kind regards
Christian Blumhoff

DISCLAIMER.....	3
COPYRIGHT	4
PURPOSE OF THIS DOCUMENT.....	5
TECHNICAL BACKGROUND AND PRE-REQUISITES.....	6
MESSAGE LIST	7
MESSAGES IN DETAIL.....	8
LOGIN.....	8
LOGOUT.....	9
FEN.....	10
MOVE	11
GAMELIST	12
REGISTER	15
GETCLIENTIP	16
POSTCLIENTIP.....	17
GAMELISTALL	18
PLAYERLIST.....	20
GETMSG	21
PLAYERID.....	23
SENDMSG	24
PGN	25
PLAYERSTATS.....	26
NEWS	28
SERVERMSG.....	29
PLAYERSONLINE.....	30
ACCEPTGAME.....	31
REVOKEGAME	32
CREATEGAME.....	33
GAMEDRAWSTATUS	35
DRAWGAME	36
REVOKEDRAWGAME.....	37
RESIGNGAME.....	38
SENDGAMECHAT	39
GETGAMECHAT	40
SWITCHREALTIME	41
EXITREALTIME	42
REALTIMESTATUS	43

Disclaimer

E&OE. Information in this document is subject to change without notice. phpChess Limited reserves the right to revise its products as it sees fit. This document describes the state of this product at the time of its publication, and may not reflect the product at all times in the future.

Use of this Product is subject to the terms of the phpChess professional plus licence and will also be subject to Service Provider terms. By downloading, installing or using this product, you agree to comply with all the relevant terms. Please refer to these terms for all permitted uses and applicable restrictions on the use of the product.

The liability of phpChess Limited with respect to the documentation and the licensed programs referred, are set out in that software licence agreement. phpChess Limited accepts no liability whatsoever for any use of the documentation or the licensed programs by any person other than a permitted user under the software licence agreement.

Copyright

All copyright and other rights in this manual and the licensed programs described in this manual are the property of phpChess Limited, save for copyright in data in respect of which the copyright belongs to the relevant data provider.

Special mention of deferred copyrights

=====

- TinyMCE by Moxicode (<http://tinymce.moxiecode.com/>)
- phpMailer by Brent R. Matzelle (<http://phpmailer.sourceforge.net/>)
- nuSoap by NuSphere and Dietrich Ayala (<http://sourceforge.net/projects/nusoap/>)
- other providers are referenced within the source code.

No part of this manual may be copied, reproduced, translated or reduced to any electronic medium or machine readable form without the written consent of phpChess Limited.

© phpChess Limited 2006

Purpose of this document

This document is designed to allow technical transparency for addressing and interacting with the phpChess Professional Plus Server. The connectivity to the phpChess Server is handled via URLs which allow any application to connect to phpChess and allow for standard chess play to commence.

This document contains technical descriptions of messages that can be sent to and received from the phpChess servers. This should allow most technically minded individuals to create, edit, enhance or change client and server applications and/ or front-ends to phpChess servers.

This document is not designed to explain the inner workings of the actual phpChess server engine, or its configuration. The following bundled components of phpChess utilise this functionality:

- phpChess Vista Gadget
- phpChess Yahoo widget
- phpChess mobile java client
- phpChess PC client
- phpChess Flash Client
- phpChess Java Web client

We plan to enhance the functionality of this area, but there are no guarantees. We hope you will find the document useful. Please share your efforts in this regards with the community where appropriate. You can contact phpChess here:

<http://www.phpchess.com/index-8.html>

Technical Background and Pre-Requisites

The phpChess server is a server side script written entirely in php. It is based on a mysql database, and requires in principle a mailing server (although this is not essential – however correspondence chess without notification of moves made is less effective and not very user centric).

phpChess is somewhat different to other correspondence chess servers since it is capable of also managing real-time games (or more correctly near real-time games).

The phpChess mobile framework is also written in php and encapsulated in the mobile.php script on the server. php coders can extend the mobile framework but should be aware that phpChess is likely to add functionality as well. It is therefore recommended (to ensure upgrade compatibility) to either post requested and working changes to phpChess, so we can integrate these into the standard product. Alternatively it is prudent to divert the includes to a new custom.php file which can then handle access secure from upgrades.

The following technical pre-requisites exist:

- Access to phpChess Professional Plus licensed server
 - o mysql database (on which to run the phpChess server) 4.1 or higher
 - o php 4.2 or higher
 - o outbound message support for web services
 - o mail server (for confirmation mails)
 - o web server running php and mysql and mail server
 - o browser for testing
- Web access

Message List

- [acceptgame](#) - Accept a challenge posted by another player
- [creategame](#) – Allows a player to challenge another player to a game
- [drawgame](#) – Requests or accepts the action to draw a game
- [exitrealtime](#) – Returns real time games to correspondence game mode.
- [fen](#) – returns the current position for a game
- [gamedrawstatus](#) - Identifies if a draw has been requested in a game
- [gamelist](#) – Lists all active games for a specific player.
- [gamelistall](#) – Gets a list from the server for all games for a player (including historic games)
- [getclientip](#) – Retrieves "Tips" from a server
- [getgamechat](#) – Retrieve game related messages from the server
- [getmsg](#) – Retrieve messages from the server
- [login](#) – Login onto the phpChess server to obtain the session ID
- [logout](#) – Clean exit from the phpChess server
- [move](#) – Posts a move in a game to the server. This is validated by the server.
- [news](#) – Reads news postings from the phpChess front screen news
- [pgn](#) – retrieves a standard game notation for a specific game
- [playerid](#) – Retrieves the ID of the current player based on session ID
- [playerlist](#) – Lists all players on the phpChess server
- [playersonline](#) – Lists all players with the current status of being online.
- [playerstats](#) – retrieves the statistics and info for a player
- [postclientip](#) – posts the logged in players IP to the server
- [realtimestatus](#) – Identifies the status of a game in real-time
- [register](#) - Allows for remote registration to phpChess server.
- [resingame](#) – Allows a player to end a game if he/she feels it can no longer be continued
- [revokedrawgame](#) – Allows a player to decline a draw request on a game
- [revokegame](#) – declines the invitation to a game (own or others)
- [sendmsg](#) – sending an asynchronous message to any other player
- [servermsg](#) – retrieving the current server status
- [sendgamechat](#) – Allows a player in a game to send a chat message to his/her opponent.
- [switchrealtime](#) – Allows a properly configured game to move temporarily into real time

Messages in detail

Login

The main function of the login command is to retrieve a valid session ID for a user. The session ID (SID) is used for all restricted phpChess server access to identify the legitimacy of the user. Session ID's timeout after 1 hour of inactivity by default, but this may be set to different value in the bin/config file (the value is set in the unit of seconds $3600 = 60\text{sec} * 60\text{min}$)

Format:

http://www.server.com/mobile.php?action=login&user=USER_NAME&pass=PASSWORD

Outbound Parameters:

action=login
user=USER_NAME
pass=PASSWORD

Return Tags:

Server
Response
SID

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=login&user=demo01&pass=welcome>

Returns:

Failed to log in

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <SID />
  </RESPONSE>
</SERVER>
```

Successful Log in

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <SID>REYyMTgzRTNGNjU1Qjc2RjkxRTAxNzJBNDc2RTdFQjR8MTkx
    </SID>
  </RESPONSE>
</SERVER>
```

Logout

The purpose of the logout function is to end the interaction with the server. This should happen at the end of any interaction stream and on application shut down. The logout function will amend the "WHO IS ONLINE LIST" by removing the user from the list, and also ensure that the session ID is deleted.

Format:

http://www.server.com/mobile.php?action=logout&sid=SESSION_ID

Outbound Parameters:

action=logout
sid=SESSION_ID

Return Tags:

Server
Response
SID

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=logout&sid=REYyMTgzRTNGNjU1Qjc2RjkxRTAxNzJBNDc2RTdFQjR8MTkx>

Returns:

Failed to log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <ERROR>IDS_SESSION_ID_INVALID</ERROR>
  </RESPONSE>
</SERVER>
```

Successful Log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <SID />
  </RESPONSE>
</SERVER>
```

Fen

The Fen command returns the Fen string for any given game. The GAME_ID is normally retrieved from the GameList or GameListAll. The fen string is created based on the usual system (or more details search the world wide web using 'chess "fen explained"' which should return the explanation of the structure. Needless to say it gives a quick understanding of the makeup of the position. It does not hold any dynamic or game status information (you can not deduct from the string if a game is over, has been agreed as a draw. It is simply a chess board representation. (http://en.wikipedia.org/wiki/Forsyth-Edwards_Notation)

Format:

http://www.server.com/mobile.php?action=fen&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=fen
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
FEN

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=fen&sid=REYyMTgzRTNGNjU1Qjc2RjkxRTAxNzJBNDc2RTdFQjR8MTkx&gameid=7A96F0F783FE0299F530DA0F8F07B173>

Returns:

Returns

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <FEN>rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq
    - 0 0</FEN>
  </RESPONSE>
</SERVER>
```

Move

The Move command simply passes a move to the server. It requires a valid Session ID and a Valid game ID. Failure to pass a legal move to the server will receive an "invalid move" response. It is important to note that there is no need to validate the move in advance of sending it to the server. All legality aspects including the 50 move rule and 3 move repetition rules are adhered to.

Format:

http://www.server.com/mobile.php?action=move&sid=SESSION_ID&gameid=GAME_ID&from=FROM_SQUARE&to=TO_SQUARE

Outbound Parameters:

```
action=move
sid=SESSION_ID
gameid=GAME_ID
from=FROM_SQUARE
to=TO_SQUARE
```

Return Tags:

```
Server
Response
Move
```

Example Use:

http://www.phpchess.com/phpchess/mobile.php?action=move&sid=SESSION_ID&gameid=7A96F0F783FE0299F530DA0F8F07B173&from=d2&to=d4

Returns:

Not legal move or not your turn

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <MOVE>false</MOVE>
  </RESPONSE>
</SERVER>
```

Legal move

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <MOVE>>true</MOVE>
  </RESPONSE>
</SERVER>
```

Gamelist

The Gamelist lists all active games for the logged in player where an action is required by the logged in player. The list returns a large number of parameters which outline who is playing, who's move it is, the names of the game and more. The Gamelist is the filtered version of Gamelistall which returns also games in the same format. However Gamelistall returns all historic games as well as current games.

There are two commands you can send to the phpChess server:

[gamelistall](#) – returns all games

[gamelist](#) – Active games only

only 4 main situations are identified by these two commands.

- a) The game is over and then it was either a draw, black won or white won - gamelist does not show these. gamelistall does.
- b) The game is going on as normal (it is either the players turn or not)
- c) Someone has challenged someone else
- d) Someone has requested a draw (i.e. no-one wins and no-one loses)

In the xml you can easily identify which situation is relevant for any specific game. The two tags that are critical are

<STATUS> and <COMPLETIONSTATUS>

<completionstatus> identifies if the game is Incomplete (I) or if it is over, which side has won (B = Black, W = White, D = Draw, and some others)

<status> identifies if the game is active or if it is finished.
Status can have the following values:

<STATUS /> - The game is over and the Completion status shows what the outcome was

<STATUS>**IDS_PLAYER_TURN**</STATUS> - A player needs to action something

<STATUS>**IDS_NOT_PLAYER_TURN**</STATUS> - The game is active, but no action is required from the player

<STATUS>**IDS_GAME_NOT_ACCEPTED**</STATUS> -This is a game challenge

<STATUS>**IDS_DRAW_REQUEST**</STATUS> - This identifies the game is in a draw request state. The GameDrawStatus identifies the situation via DRAWGAMECODE.

<DRAWCODE>**IDS_DRAW_REQUESTED**</DRAWCODE> - a draw has been requested

<DRAWCODE>**IDS_NO_DRAW**</DRAWCODE> - No draw request has been made.

Tips:

B) If the STATUS shows **IDS_NOT_PLAYER_TURN**, then you have a draw

request, but the player still needs to make a move.

C) If the Status says: **IDS_PLAYER_TURN** , then you have a normal game and the player should make a move

D) **IDS_GAME_NOT_ACCEPTED** means you have a challenge. The way to identify if the challenge was made by the logged in player, or the other player is derived from the tag <INITIATOR>

If the Initiator = the players own ID then the logged in player challenged
 If the initiator is NOT the logged in player, then the logged in player has been challenged to a game (use [playerid](#)).

Format:

http://www.server.com/mobile.php?action=gamelist&sid=SESSION_ID

Outbound Paramters:

action=gamelist
 sid=SESSION_ID

Return Tags:

Server
 Response
 Games
 Status
 Completionstatus
 Gametype
 Rated
 Timeout
 Timecontrol1
 Timecontrol2
 Timecreated
 Description
 Initiator
 White
 Black
 Nextmove
 Gameid

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=logout&sid=REYyMTgzRTNGNjU1Qjc2RjkxRTAxNzJBNDc2RTdFQjR8MTkx>

Returns:

Retruns

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <GAMES>
      <STATUS>IDS_GAME_NOT_ACCEPTED</STATUS>
      <COMPLETIONSTATUS>I</COMPLETIONSTATUS>
      <GAMETYPE>GT_NORMAL_GAME</GAMETYPE>
      <RATED>>true</RATED>
      <TIMEOUT>IDS_NORMAL</TIMEOUT>
```

```
<TIMECONTROL1>IDS_NULL</TIMECONTROL1>
<TIMECONTROL2>IDS_NULL</TIMECONTROL2>
<TIMECREATED>1171665045</TIMECREATED>
<DESCRIPTION>demo01 VS demo03</DESCRIPTION>
<INITIATOR>5</INITIATOR>
<WHITE>5</WHITE>
<BLACK>6</BLACK>
<NEXTMOVE />
<GAMEID>7A96F0F783FE0299F530DA0F8F07B173</GAMEID>
</GAMES>
</RESPONSE>
</SERVER>
```

Register

This command allows for anyone to simply register on a phpChess server using simply their e-mail and desired user address. Please note that this isn't perfect since the response depends entirely on the configuration the admin has placed on the phpChess server. This may include manual approval of applicants. Notification of approval is always by e-mail. So the process is not complete by sending this message. The server doesn't simply return a password.

This command is likely to be extended and changed.

Format:

http://www.server.com/mobile.php?action=register&userid=USER_ID&email=EMAIL_ADDRESS

Outbound Parameters:

action=register
userid = USER_ID
email = E-MAIL Address

Return Tags:

Server
Response
Register

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=register&userid=demo04&email=demo@email.com>

Returns:

Successful User creation

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <REGISTER>Your account has been created. An initial password
      will be emailed to the address you specified.</REGISTER>
  </RESPONSE>
</SERVER>
```

Getclientip

This command allows for the retrieval of a players IP address should it have been posted using [PostClientip](#). It allows for players to link directly for a real-time game. While this is not offered directly via phpChess, phpChess does help facilitate this via this command set.

Format:

http://www.server.com/mobile.php?action=getclientip&sid=SESSION_ID&playerid=PLAYER_ID

Outbound Parameters:

action=getclientip
sid=SESSION_ID
playerid = PLAYER_ID

Return Tags:

Server
IP

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=getclientip&sid=QTAzMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&playerid=5>

Returns:

Failed to return an IP

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER />
```

Returned an IP

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <IP>86.138.101.128</IP>
</SERVER>
```

Postclientip

This handy tool lets clients post an IP address to the phpChess server. The ip can then be used to organize direct IP to IP games. While technical in nature it allows also for an additional security aspect. It is the direct partner to [GetClientIP](#)

Format:

http://www.server.com/mobile.php?action=postclientip&sid=SESSION_ID&ip=IP_ADDRESS

Outbound Parameters:

action=postclientip
sid=SESSION_ID
ip=IP_ADDRESS

Return Tags:

Server
Response
FEN

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=postclientip&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&ip=86.138.101.128>

Returns:

Success

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <ACCEPTGAME>true</ACCEPTGAME>
  </RESPONSE>
</SERVER >
```

Gamelistall

The Gamelist lists all active games for the logged in player where an action is required by the logged in player. The list returns a large number of parameters which outline who is playing, who's move it is, the names of the game and more. The Gamelist is the filtered version of Gamelistall which returns also games in the same format. However Gamelistall returns all historic games as well as current games.

There are two commands you can send to the phpChess server:

[gamelistall](#) – returns all games

[gamelist](#) – Active games only

Detailed description of the return given by this function can be found under [gamelist](#).

Format:

http://www.server.com/mobile.php?action=gamelistall&sid=SESSION_ID&playerid=PLAYER_ID

Outbound Parameters:

action=fen
sid=SESSION_ID
playerid=PLAYER_ID

Return Tags:

Server
Response
Games
Status
Completionstatus
Gametype
Rated
Timeout
Timecontrol1
Timecontrol2
Timecreated
Description
Initiator
White
Black
Nextmove
Gameid

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=gamelistall&sid=REYyMTgzRTNGNjU1Qjc2RjkxRTAxNzJBNDc2RTdFQjR8MTkx>

Return:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <GAMES>
```

```
<STATUS>IDS_GAME_NOT_ACCEPTED</STATUS>
<COMPLETIONSTATUS>I</COMPLETIONSTATUS>
<GAMETYPE>GT_NORMAL_GAME</GAMETYPE>
<RATED>>true</RATED>
<TIMEOUT>IDS_NORMAL</TIMEOUT>
<TIMECONTROL1>IDS_NULL</TIMECONTROL1>
<TIMECONTROL2>IDS_NULL</TIMECONTROL2>
<TIMECREATED>1171665045</TIMECREATED>
<DESCRIPTION>demo01 VS demo03</DESCRIPTION>
<INITIATOR>5</INITIATOR>
<WHITE>5</WHITE>
<BLACK>6</BLACK>
<NEXTMOVE />
<GAMEID>7A96F0F783FE0299F530DA0F8F07B173</GAMEID>
</GAMES>
</RESPONSE>
</SERVER>
```

Playerlist

Playerlist retrieves a list of players from the selected phpChess server. It allows a logged in player to identify potential opponents. This is likely to be extended by teams, clans and buddy attributes in the future.

Format:

http://www.server.com/mobile.php?action=playerlist&sid=SESSION_ID

Outbound Parameters:

action=playerlist
sid=SESSION_ID

Return Tags:

Server
Response
Players
PID
Userid
Signuptime
Points
Inclub
Inbuddylist
status

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=playerlist&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==>

Returns:

Failed to log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <PLAYERS>
      <PID>1</PID>
      <USERID>demo1</USERID>
      <SIGNUPTIME>02-26-2006</SIGNUPTIME>
      <POINTS>1200</POINTS>
      <INCLUB>>false</INCLUB>
      <INBUDDYLIST>>true</INBUDDYLIST>
      <STATUS>PLAYER_OFFLINE</STATUS>
    </PLAYERS>
    <PLAYERS>
      <PID>3</PID>
      <USERID>test1</USERID>
      <SIGNUPTIME>03-01-2006</SIGNUPTIME>
      <POINTS>1200</POINTS>
      <INCLUB>>true</INCLUB>
      <INBUDDYLIST>>true</INBUDDYLIST>
      <STATUS>PLAYER_ONLINE</STATUS>
    </PLAYERS>
  </RESPONSE>
</SERVER>
```

Getmsg

Getmsg is part of the process of communicating between players without having to post within a game. A message can be send to any player using the ID. Getmsg is the direct opposite of [SendMsg](#).

Format:

http://www.server.com/mobile.php?action=getmsg&sid=SESSION_ID

Outbound Parameters:

action=getmsg
sid=SESSION_ID

Return Tags:

Server
Response
Message
ID
Type
Sendername
Playerid
Tournamentid
Msgtextraw
Msgtext
unixtime

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=getmsg&sid=QTAzMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==>

Returns:

Returned Messages

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <MESSAGE>
      <ID>28</ID>
      <TYPE>TEXT_MESSAGE</TYPE>
      <SENDERNAME>demo01</SENDERNAME>
      <PLAYERID />
      <GAMEID />
      <TOURNAMENTID />
      <MSGTEXTRAW>"thanks"</MSGTEXTRAW>
      <MSGTEXT>demo01 sent the following message:
      "thanks"</MSGTEXT>
      <UNIXTIME>1169909412</UNIXTIME>
    </MESSAGE>
    <MESSAGE>
      <ID>30</ID>
      <TYPE>TEXT_MESSAGE</TYPE>
      <SENDERNAME>demo01</SENDERNAME>
      <PLAYERID />
      <GAMEID />
      <TOURNAMENTID />
      <MSGTEXTRAW>Testing again</MSGTEXTRAW>
```

```
<MSGTEXT>demo01 sent the following message: Testing  
again</MSGTEXT>  
<UNIXTIME>1169909687</UNIXTIME>  
</MESSAGE>  
</RESPONSE>  
</SERVER>
```

Playerid

In order to ensure you can identify the current own player ID you can call this action. It will retrieve the current logged in player ID.

Format:

http://www.server.com/mobile.php?action=playerid&sid=SESSION_ID

Outbound Parameters:

action=fen
sid=SESSION_ID
playerid=PLAYER_ID

Return Tags:

Server
Response
PID

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=playerid&sid=QTAzMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==#>

Returns:

PlayerID Return

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <PID>5</PID>
  </RESPONSE>
</SERVER>
```

Sendmsg

Sendmsg allows a player to send a receiving player (ID) a text message. This will be delivered to the receiving player's inbox. Received messages can be read using [getmsg](#).

Format:

http://www.server.com/mobile.php?action=sendmsg&sid=SESSION_ID&rplayerid=RECIVER_PLAYER_ID&message=MESSAGE

Outbound Parameters:

action=sendmsg
sid=SESSION_ID
rplayerid=PLAYER_ID
message=MESSAGE

Return Tags:

Server
Response
Sent

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=sendmsg&sid=QTAzMEE1MjNGNDkwQkE5MjlEMTEyQzYyMDUwMDhCNzJ8NQ==&rplayerid=5&message=MESSAGE%20Test>

Returns:

Message sent

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <SENT>true</SENT>
  </RESPONSE>
</SERVER>
```

Pgn

The pgn call is designed to retrieve a standard game notation file for a specific game ID. This allows users to pass the result over to a pgn viewer or interpret the results via a database. The pgn file format is supported by many chess programs and ensures compatibility with many chess databases for game analysis.

Format:

http://www.server.com/mobile.php?action=pgn&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=pgn
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
PGN

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=pgn&sid=QTAzMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

PGN Returned

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <PGN>[Event "80E981A5739550CAB372C3DC089E58AE"] [CustomFEN ""]
      [Mode "ICS"] [Round "1"] [White "demo03"] [Black "demo01"] [WhiteELO
        ""] [BlackELO ""] [Result "*" ] 1. d4 *</PGN>
    </RESPONSE>
  </SERVER>
```

Playerstats

This command returns the statistics and information elements he or she maintained on the phpChess server. There is currently no command to write this information back from a client via the mobile code.

Format:

http://www.server.com/mobile.php?action=playerstats&sid=SESSION_ID&playerid=PLAYER_ID

Outbound Parameters:

action=playerstats
sid=SESSION_ID
rplayerid=PLAYER_ID

Return Tags:

Server
Response
Statistics
Win
Loss
Draw
Points
Memberdays
Signupdate
Lastlogindate
Lastmovedate
Realname
Location
Age
Selfrating
Comment
Chessplayer

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=playerstats&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&playerid=5>

Returns:

Response

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <STATISTICS>
      <WIN>6</WIN>
      <LOSS>5</LOSS>
      <DRAW>3</DRAW>
      <POINTS>1201</POINTS>
      <MEMBERDAYS>49</MEMBERDAYS>
      <SIGNUPDATE>01-20-2007</SIGNUPDATE>
      <LASTLOGINDATE>2007-03-04 02:33:57</LASTLOGINDATE>
      <LASTMOVEDATE>2007-02-25 8:23:07</LASTMOVEDATE>
      <REALNAME />
      <LOCATION />
      <AGE />
      <SELFRATING />
```

```
<COMMENT />  
<CHESSPLAYER />  
</STATISTICS>  
</RESPONSE>  
</SERVER>
```

News

News is the first thing you see when coming to a phpChess server. It sets the general tone for why the phpChess server was installed in the first place. It is decided upon by the admin and is textual.

Format:

<http://www.server.com/mobile.php?action=news>

Outbound Parameters:

action=news

Return Tags:

Server
Response
News
Title
Message
Date

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=news>

Returns:

News returned

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <NEWS>
      <TITLE>Welcome to ToB2 chess</TITLE>
      <MESSAGE>We thought the nearest thing to having fun with Tower
of Babel is chess, so we have installed this great chess server on
our site thanks to phpChess.net :) Remember when registering to
add the sender of this server to your e-mail list, or spam filters will
interrupt your move messages to your e-mail address. Have fun,
and keep it clean </MESSAGE>
      <DATE>2006-02-26 15:38:21</DATE>
    </NEWS>
  </RESPONSE>
</SERVER>
```

Servermsg

The Servermsg is designed to allow the Administrator of a phpChess server to communicate the status of the server to the members. These messages should normally include upgrade plans, and down-time schedules as well as other information around the phpChess server. However in the end this is a text message from the admin to all registered players. The server message does not require a log in.

Format:

http://www.server.com/mobile.php?action=servermsg&sid=SESSION_ID

Outbound Parameters:

action=servermsg
sid=SESSION_ID

Return Tags:

Server
Response
Servermsg
Message
Date

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=servermsg&sid=QTAzMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==#>

Returns:

News

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <SERVERMSG>
      <MESSAGE>This is a test server</MESSAGE>
      <DATE>2007-03-11 10:00:01</DATE>
    </SERVERMSG>
    <SERVERMSG>
      <MESSAGE>We are planning to in early April</MESSAGE>
      <DATE>2007-03-11 10:00:23</DATE>
    </SERVERMSG>
  </RESPONSE>
</SERVER>
```

Playersonline

This lists all players which are currently online or had an online activity within the time-out parameter. Time-out parameters can be set in the config.php file in the bin directory of the phpChess server (measured in seconds). Default is one hour (3600 sec).

Format:

http://www.server.com/mobile.php?action=playersonline&sid=SESSION_ID

Outbound Parameters:

action=playersonline
sid=SESSION_ID

Return Tags:

Server
Response
Players
Playerid
Userid
sessiontime

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=playersonline&sid=QTAzMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==>

Returns:

Failed to log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <PLAYERS>
      <PLAYERID>5</PLAYERID>
      <USERID>demo01</USERID>
      <SESSIONTIME>1173607549</SESSIONTIME>
    </PLAYERS>
  </RESPONSE>
</SERVER>
```

Acceptgame

Acceptgame is the response to a game challenge.. It requires a specific game id and session id to execute. The standard execution sequence would be:

1. Run gamelist
2. Filter game challenges
3. Identify gamechallengestatus
4. Filter inbound challenges
5. Accept or reject game using Acceptgame or revokegame

Format:

http://www.server.com/mobile.php?action=acceptgame&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=acceptgame
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
Acceptgame

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=acceptgame&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

Success

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <ACCEPTGAME>true</ACCEPTGAME>
  </RESPONSE>
</SERVER>
```

Failed

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <ACCEPTGAME>>false</ACCEPTGAME>
  </RESPONSE>
</SERVER>
```

Revokegame

The idea behind the revoke game command is to give the player to give the control to withdraw a challenge. This became necessary due to the fact that people who where challenged didn't decline them. The revoke command is there to allow players to clean up their challenge list

Format:

http://www.server.com/mobile.php?action=revokegame&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=revokegame
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
revokegame

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=revokegame&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=22A53027E098805DB538BAB4AA6FE2D1>

Returns:

Success

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <REVOKEGAME>true</REVOKEGAME>
  </RESPONSE>
</SERVER>
```

Failed

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <REVOKEGAME>>false</REVOKEGAME>
  </RESPONSE>
</SERVER>
```

Creategame

The Creategame command allows for the creation of new games (ie the creation of a challenge to another player) It has a fair number of parameters and is extremely versatile. The following list shows the use and impact of each parameter:

oplayerid - cannot equal the challenger's Player ID and must be an integer value.

mypiececolor - must equal either w or b.

fen - set this value only if you are creating a game with a custom FEN

move1, time1 - set these values only if you are creating an active real time game.

move1 - Integer value only, this sets the amount of moves you want in a given time.

time1 - Integer value only, this sets the time in seconds.

move2, time2 - set these values only if you are creating an active real time game.

move2 - Integer value only, this sets the second amount of moves you want in a given time.

time2 - Integer value only, this sets the second time in seconds.

brtGame - set this value if you are creating an active real time game. Must equal 1 (true) or 0 (false).

precreate - Set this value to 0.

brealtimeposs - set this value if you are creating a pasv real time game. Must equal 1 (true) or 0 (false).

ratingtype - must equal either grated (rated Game), gunrated (Unrated Game).

If you are creating a pasv real-time game or a normal game, the following "gametime" values must be set:

gametime - C-Normal, C-Blitz, C-Short, C-Slow, C-Snail

If you are creating an active real-time game, the following values must be set:

gametime - RT-Custom (User sets move1, time1, move2, time2 values), RT-Blitz (move1=50, time1=15), RT-Short (move1=50, time1=60), RT-Normal (move1=50, time1=120), RT-Slow (move1=50, time1=720)..

Format:

```
http://elite.goliathsystems.com/command2/mobile.php?action=creategame&sid=SESSION_ID&oplayerid=C
HALLENGED_PLAYERID&mypiececolor=PIECE_COLOR&fen=GAME_FEN&move1=RT_MOVE_1&time1=
RT_TIME_1&move2=RT_MOVE_2&time2=RT_TIME_2&brtGame=IS_GAME_ACTIVE_REALTIME&precre
ate=PRECREATED_GAME_ID&brealtimeposs=IS_PASV_REALTIME&ratingtype=RATING_TYPE&gameti
me=GAME_TIMEOUT
```

Outbound Parameters:

see above

Return Tags:

Server

Response

Creategamecode

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=creategame&sid=ODRCMEQyNDM0MDVEQ0FFNUY3RUFDMkFCQjJBRTMzMEN8NQ==&oplayerid=6&mypiececolor=w&brtGame=0&precreate=0&ratingtype=grated&gametime=C-Blitz>

Returns:

Failed to log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <CREATEGAMECODE>27A308C4302BDD7BFF4B5DD873649E8B0
00000500000006</CREATEGAMECODE>
  </RESPONSE>
</SERVER>
```

Gamedrawstatus

This information request should be considered as mandatory for any game status. In future versions this might be moved into the game status itself, but will always stay available as a stand alone call.

The gamedrawstatus call identifies if a player has made a request to draw the game (drawgame).

Format:

`http://www.server.com/mobile.php?action=gamedrawstatus&sid=SESSION_ID&gameid=GAME_ID`

Outbound Parameters:

action=gamedrawstatus
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
Drawcode

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=gamedrawstatus&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

No Draw requested

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <DRAWCODE>IDS_NO_DRAW</DRAWCODE>
  </RESPONSE>
</SERVER>
```

Draw requested

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <DRAWCODE>IDS_DRAW_REQUESTED</DRAWCODE>
  </RESPONSE>
</SERVER>
```

Drawgame

The drawgame call simply sends a request or confoirmation for a specific Game ID. The drawgame process requires two steps, a request for a draw, and the acceptance of a draw.

We would expect the draw game action to go as follows:

- 1 - Player A) send a draw request
- 2 – Player A) makes move
- 3 – Player B) check drawgamestatus
- 4) – Player B) either send drawgame to confirm draw or send revokeddrawgame to decline the draw.

It is technically possible to ignore the draw game request, but this would not be sporting.

Format:

http://www.server.com/mobile.php?action=drawgame&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=drawgame
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
Draw

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=drawgame&sid=QTAzMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

Failed to log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <DRAW>true</DRAW>
  </RESPONSE>
</SERVER>
```

Revokedrawgame

This command allows a player, who has been offered a draw in a game to decline it. This often happens because players feel that a draw is not the optimal outcome for a game.

Format:

`http://www.server.com/mobile.php?action=revokedrawgame&sid=SESSION_ID&gameid=GAME_ID`

Outbound Parameters:

action=revokedrawgame
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
drawcode

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=revokedrawgame&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

Revoke Draw successful

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <DRAWCODE>true</DRAWCODE>
  </RESPONSE>
</SERVER>
```

Revoke Draw not required

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <DRAWCODE>>false</DRAWCODE>
  </RESPONSE>
</SERVER>
```

Resinggame

This is a unilateral action of a player to end a game he or she fails can not be won or continued. While games can also time out, it is by far the more sporting response to a game that can not be won. The server will recognize draws and check mates, however most players will resign once a situation has become untenable to avoid a lengthy but clear end-game.

Format:

http://www.server.com/mobile.php?action=resinggame&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=resinggame
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
Resign

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=resinggame&sid=QUNFRkI1M0NEM0E2QUE5MDREQTdGQUQwRjU2MTQyOUV8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

Failed to log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <RESIGN>true</RESIGN>
  </RESPONSE>
</SERVER>
```

Sendgamechat

Sendgamechat is designed to allow players to post message to each other during a game. It is received using [Getgamechat](#).

Format:

`http://www.server.com/mobile.php?action=sendgamechat&sid=SESSION_ID&gameid=GAME_ID&msg=MESSAGE`

Outbound Parameters:

action=revokegame
sid=SESSION_ID
gameid=GAME_ID
msg=MESSAGE

Return Tags:

Server
Response
sentmsg

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=sendgamechat&sid=QTAzMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=25489002A7D3F6E3B5E0012ECB793075&msg=hello123>

Returns:

Chat sent ok

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <SENTMSG>true</SENTMSG>
  </RESPONSE>
</SERVER>
```

Failed to sent ok

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <SENTMSG>>false</SENTMSG>
  </RESPONSE>
</SERVER>
```

Getgamechat

phpChess lets players chat during the game. This is asynchronous to game moves. Getgamechat retrieves the chat that has taken place on the game. It will always be the complete list. GetGameChat is different from message sending (getmsg). Getgamechat is directly linked to [SendGameChat](#)

Format:

http://www.server.com/mobile.php?action=getgamechat&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=getgamechat
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
msg

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=getgamechat&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

GameChat returned

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <MSG><demo03> Maybe I will have more luck this time :) <demo03> Good
    luck <demo03> Hi</MSG>
  </RESPONSE>
</SERVER> >
```

Switchrealtime

phpChess allows for many different game modes. One of the unique capabilities is the temporary use of real-time within a turn based game (passive real-time). The idea behind this command is that if two players happen to be online at the same time, it would sometimes be desirable to move the game into real-time for a while. A game only permits this if it was started in passive real-time mode.

Format:

http://www.server.com/mobile.php?action=switchrealtime&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=switchrealtime
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
setrt

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=switchrealtime&sid=ODRCMEQyNDM0MDVEQ0FFNUY3RUFDmKFCQjJBRTMzMEN8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

Failed to log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <SETRT>true</SETRT>
  </RESPONSE>
</SERVER>
```

Exitrealtime

This command is designed to allow passive real-time games to stop real time interaction. The result positions games back from real-time into correspondence mode.

Format:

`http://www.server.com/mobile.php?action=exitrealtime&sid=SESSION_ID&gameid=GAME_ID`

Outbound Parameters:

`action=exitrealtime`
`sid=SESSION_ID`
`gameid=GAME_ID`

Return Tags:

Server
Response
Exitrt

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=exitrealtime&sid=QTazMEE1MjNGNDkwQkE5MjIEMTEyQzYyMDUwMDhCNzJ8NQ==&gameid=80E981A5739550CAB372C3DC089E58AE>

Returns:

Exit Realtime response

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <EXITRT>true</EXITRT>
  </RESPONSE>
</SERVER>
```

Realtimestatus

This command shows the realtime game status for any game. Please note that the return is simply showing if a game is in realtime or not. It goes through two stages:

- 1) Real time requested
- 2) Game is in realtime

Please note that both players have to accept real-time mode to be in real-time mode.

Format:

http://www.server.com/mobile.php?action=realtimestatus&sid=SESSION_ID&gameid=GAME_ID

Outbound Parameters:

action=realtimestatus
sid=SESSION_ID
gameid=GAME_ID

Return Tags:

Server
Response
Rtstatus

Example Use:

<http://www.phpchess.com/phpchess/mobile.php?action=logout&sid=REYyMTgzRTNGNjU1Qjc2RjkxRTAxNzJBNDc2RTdFQjR8MTkx>

Returns:

Failed to log out

```
<?xml version="1.0" encoding="UTF-8" ?>
<SERVER>
  <RESPONSE>
    <RTSTATUS>IDS_USER_REQUESTED_REAL_TIME</RTSTATUS>
  </RESPONSE>
</SERVER>
```