

# PHP OTServ Toolkit



# Contents

<a href="#">POT</a>	1
<a href="#">PHP 5.0</a>	3
<a href="#">POT class preview</a>	5
<a href="#">Quick start</a>	6
<a href="#">DAO objects</a>	9
<a href="#">Guilds</a>	13
<a href="#">Guild action drivers</a>	15
<a href="#">Account number hack</a>	18
<a href="#">Server online status</a>	19
<a href="#">Package POT Procedural Elements</a>	22
<a href="#">E_OTS_NoDriver.php</a>	22
<a href="#">E_OTS_NotLoaded.php</a>	23
<a href="#">IOTS_DAO.php</a>	24
<a href="#">IOTS_DB.php</a>	25
<a href="#">IOTS_GuildAction.php</a>	26
<a href="#">OTS.php</a>	27
<a href="#">OTS_Account.php</a>	28
<a href="#">OTS_Accounts_List.php</a>	29
<a href="#">OTS_Base_DAO.php</a>	30
<a href="#">OTS_Base_List.php</a>	31
<a href="#">OTS_Container.php</a>	32
<a href="#">OTS_DB_MySQL.php</a>	33
<a href="#">OTS_DB_ODBC.php</a>	34
<a href="#">OTS_DB_PostgreSQL.php</a>	35
<a href="#">OTS_DB_SQLite.php</a>	36
<a href="#">OTS_Group.php</a>	37
<a href="#">OTS_Groups_List.php</a>	38
<a href="#">OTS_Guild.php</a>	39
<a href="#">OTS_GuildRank.php</a>	40
<a href="#">OTS_GuildRanks_List.php</a>	41
<a href="#">OTS_Guilds_List.php</a>	42
<a href="#">OTS_InfoRespond.php</a>	43
<a href="#">OTS_Item.php</a>	44
<a href="#">OTS_Player.php</a>	45
<a href="#">OTS_Players_List.php</a>	46
<a href="#">OTS_SQLField.php</a>	47
<a href="#">OTS_SQLFilter.php</a>	48
<a href="#">OTS_SQLite_Results.php</a>	49
<a href="#">Package POT Classes</a>	50
<a href="#">Class E_OTS_NoDriver</a>	50
<a href="#">Class E_OTS_NotLoaded</a>	50

<a href="#">Class IOTS DAO</a>	51
<a href="#">Constructor construct</a>	51
<a href="#">Class IOTS DB</a>	52
<a href="#">Constructor construct</a>	52
<a href="#">Method fieldName</a>	53
<a href="#">Method lastInsertId</a>	53
<a href="#">Method limit</a>	53
<a href="#">Method SQLquery</a>	54
<a href="#">Method SQLquote</a>	54
<a href="#">Method tableName</a>	55
<a href="#">Class IOTS GuildAction</a>	55
<a href="#">Constructor construct</a>	56
<a href="#">Method addRequest</a>	56
<a href="#">Method deleteRequest</a>	57
<a href="#">Method listRequests</a>	57
<a href="#">Method submitRequest</a>	58
<a href="#">Class OTS Account</a>	58
<a href="#">Method ban</a>	59
<a href="#">Method block</a>	59
<a href="#">Method count</a>	59
<a href="#">Method create</a>	60
<a href="#">example: account.php</a>	60
<a href="#">Method createEx</a>	61
<a href="#">example: account.php</a>	61
<a href="#">Method delete</a>	62
<a href="#">Method find</a>	63
<a href="#">Method getCustomField</a>	63
<a href="#">Method getEmail</a>	64
<a href="#">Method getGroup</a>	64
<a href="#">Method getId</a>	65
<a href="#">Method getIterator</a>	65
<a href="#">Method getPACCDays</a>	65
<a href="#">Method getPassword</a>	66
<a href="#">Method getPlayers</a>	66
<a href="#">Method getPlayersList</a>	67
<a href="#">Method isBanned</a>	67
<a href="#">Method isBlocked</a>	68
<a href="#">Method isLoaded</a>	68
<a href="#">Method load</a>	68
<a href="#">Method save</a>	69
<a href="#">Method setCustomField</a>	69
<a href="#">Method setEmail</a>	70
<a href="#">Method setGroup</a>	70
<a href="#">Method setPACCDays</a>	71
<a href="#">Method setPassword</a>	71
<a href="#">Method unban</a>	72
<a href="#">Method unblock</a>	72
<a href="#">Class OTS Accounts List</a>	73
<a href="#">Method deleteAccount</a>	73

<a href="#">Method init</a>	73
<a href="#">Class OTS_Base_DAO</a>	74
<a href="#">Var \$db</a>	74
<a href="#">Constructor construct</a>	75
<a href="#">Method clone</a>	75
<a href="#">Method set_state</a>	76
<a href="#">Method sleep</a>	76
<a href="#">Method wakeup</a>	76
<a href="#">Class OTS_Base_List</a>	77
<a href="#">Var \$class</a>	77
<a href="#">Var \$table</a>	78
<a href="#">Constructor construct</a>	78
<a href="#">Method count</a>	78
<a href="#">Method current</a>	79
<a href="#">Method init</a>	79
<a href="#">Method key</a>	79
<a href="#">Method next</a>	80
<a href="#">Method orderBy</a>	80
<a href="#">Method resetOrder</a>	81
<a href="#">Method rewind</a>	81
<a href="#">Method setFilter</a>	81
<a href="#">Method setLimit</a>	82
<a href="#">Method setOffset</a>	82
<a href="#">Method valid</a>	83
<a href="#">Method set_state</a>	83
<a href="#">Method sleep</a>	83
<a href="#">Method wakeup</a>	84
<a href="#">Class OTS_Container</a>	84
<a href="#">Method addItem</a>	85
<a href="#">Method count</a>	85
<a href="#">Method current</a>	86
<a href="#">Method key</a>	86
<a href="#">Method next</a>	86
<a href="#">Method removeItem</a>	86
<a href="#">Method rewind</a>	87
<a href="#">Method valid</a>	87
<a href="#">Class OTS_DB_MySQL</a>	88
<a href="#">Constructor construct</a>	88
<a href="#">Method fieldName</a>	89
<a href="#">Method limit</a>	89
<a href="#">Method SQLquery</a>	90
<a href="#">Method SQLquote</a>	90
<a href="#">Method tableName</a>	91
<a href="#">Class OTS_DB_ODBC</a>	91
<a href="#">Constructor construct</a>	92
<a href="#">Method fieldName</a>	92
<a href="#">Method limit</a>	93
<a href="#">Method SQLquery</a>	93
<a href="#">Method SQLquote</a>	94

<a href="#">Method tableName</a>	94
<a href="#">Class OTS_DB_PostgreSQL</a>	95
<a href="#">Constructor construct</a>	95
<a href="#">Method fieldName</a>	96
<a href="#">Method limit</a>	96
<a href="#">Method SQLquery</a>	97
<a href="#">Method SQLquote</a>	97
<a href="#">Method tableName</a>	98
<a href="#">Class OTS_DB_SQLite</a>	98
<a href="#">Constructor construct</a>	99
<a href="#">Method fieldName</a>	99
<a href="#">Method limit</a>	100
<a href="#">Method SQLquery</a>	100
<a href="#">Method SQLquote</a>	101
<a href="#">Method tableName</a>	101
<a href="#">Class OTS_Group</a>	102
<a href="#">Method count</a>	102
<a href="#">Method delete</a>	103
<a href="#">Method getAccess</a>	103
<a href="#">Method getCustomField</a>	104
<a href="#">Method getFlags</a>	104
<a href="#">Method getId</a>	105
<a href="#">Method getIterator</a>	105
<a href="#">Method getMaxDepotItems</a>	105
<a href="#">Method getMaxVIPList</a>	106
<a href="#">Method getName</a>	106
<a href="#">Method getPlayers</a>	107
<a href="#">Method getPlayersList</a>	107
<a href="#">Method isLoaded</a>	107
<a href="#">Method load</a>	108
<a href="#">Method save</a>	108
<a href="#">Method setAccess</a>	109
<a href="#">Method setCustomField</a>	109
<a href="#">Method setFlags</a>	110
<a href="#">Method setMaxDepotItems</a>	110
<a href="#">Method setMaxVIPList</a>	111
<a href="#">Method setName</a>	111
<a href="#">Class OTS_Groups_List</a>	112
<a href="#">Method deleteGroup</a>	112
<a href="#">Method init</a>	113
<a href="#">Class OTS_Guild</a>	113
<a href="#">Method acceptInvite</a>	114
<a href="#">Method acceptRequest</a>	114
<a href="#">Method count</a>	115
<a href="#">Method delete</a>	115
<a href="#">Method deleteInvite</a>	115
<a href="#">Method deleteRequest</a>	116
<a href="#">Method find</a>	116
<a href="#">Method getCreationData</a>	117

<a href="#">Method <code>getCustomField</code></a>	117
<a href="#">Method <code>getGuildRanks</code></a>	118
<a href="#">Method <code>getGuildRanksList</code></a>	118
<a href="#">Method <code>getId</code></a>	119
<a href="#">Method <code>getIterator</code></a>	119
<a href="#">Method <code>getName</code></a>	120
<a href="#">Method <code>getOwner</code></a>	120
<a href="#">Method <code>invite</code></a>	120
<a href="#">Method <code>isLoading</code></a>	121
<a href="#">Method <code>listInvites</code></a>	121
<a href="#">Method <code>listRequests</code></a>	122
<a href="#">Method <code>load</code></a>	122
<a href="#">Method <code>request</code></a>	122
<a href="#">Method <code>save</code></a>	123
<a href="#">Method <code>setCreationData</code></a>	123
<a href="#">Method <code>setCustomField</code></a>	124
<a href="#">Method <code>setInvitesDriver</code></a>	125
<a href="#">Method <code>setName</code></a>	125
<a href="#">Method <code>setOwner</code></a>	125
<a href="#">Method <code>setRequestsDriver</code></a>	126
<a href="#">Method <code>clone</code></a>	126
<a href="#">Method <code>sleep</code></a>	127
<a href="#">Class <code>OTS_GuildRank</code></a>	127
<a href="#">Method <code>count</code></a>	128
<a href="#">Method <code>delete</code></a>	128
<a href="#">Method <code>find</code></a>	128
<a href="#">Method <code>getCustomField</code></a>	129
<a href="#">Method <code>getGuild</code></a>	130
<a href="#">Method <code>getId</code></a>	130
<a href="#">Method <code>getIterator</code></a>	130
<a href="#">Method <code>getLevel</code></a>	131
<a href="#">Method <code>getName</code></a>	131
<a href="#">Method <code>getPlayers</code></a>	131
<a href="#">Method <code>getPlayersList</code></a>	132
<a href="#">Method <code>isLoading</code></a>	132
<a href="#">Method <code>load</code></a>	133
<a href="#">Method <code>save</code></a>	133
<a href="#">Method <code>setCustomField</code></a>	133
<a href="#">Method <code>setGuild</code></a>	134
<a href="#">Method <code>setLevel</code></a>	135
<a href="#">Method <code>setName</code></a>	135
<a href="#">Class <code>OTS_GuildRanks_List</code></a>	136
<a href="#">Method <code>deleteGuildRank</code></a>	136
<a href="#">Method <code>init</code></a>	137
<a href="#">Class <code>OTS_Guilds_List</code></a>	137
<a href="#">Method <code>deleteGuild</code></a>	137
<a href="#">Method <code>init</code></a>	138
<a href="#">Class <code>OTS_InfoRespond</code></a>	138
<a href="#">Method <code>getClientVersion</code></a>	139

<a href="#">Method getEmail</a>	139
<a href="#">Method getIP</a>	139
<a href="#">Method getLocation</a>	140
<a href="#">Method getMapAuthor</a>	140
<a href="#">Method getMapHeight</a>	140
<a href="#">Method getMapName</a>	141
<a href="#">Method getMapWidth</a>	141
<a href="#">Method getMaxPlayers</a>	141
<a href="#">Method getMonstersCount</a>	142
<a href="#">Method getMOTD</a>	142
<a href="#">Method getName</a>	142
<a href="#">Method getOnlinePlayers</a>	143
<a href="#">Method getOwner</a>	143
<a href="#">Method getPlayersPeak</a>	143
<a href="#">Method getPort</a>	144
<a href="#">Method getServer</a>	144
<a href="#">Method getServerVersion</a>	144
<a href="#">Method getTSPQVersion</a>	145
<a href="#">Method getUptime</a>	145
<a href="#">Method getURL</a>	145
<a href="#">Class OTS_Item</a>	146
<a href="#">Constructor construct</a>	146
<a href="#">Method count</a>	146
<a href="#">Method getAttributes</a>	147
<a href="#">Method getCount</a>	147
<a href="#">Method getId</a>	147
<a href="#">Method setAttributes</a>	148
<a href="#">Method setCount</a>	148
<a href="#">Class OTS_Player</a>	149
<a href="#">Method ban</a>	149
<a href="#">Method delete</a>	150
<a href="#">Method find</a>	150
<a href="#">Method getAccount</a>	151
<a href="#">Method getCap</a>	151
<a href="#">Method getConditions</a>	151
<a href="#">Method getCustomField</a>	152
<a href="#">Method getDepot</a>	152
<a href="#">Method getDirection</a>	153
<a href="#">Method getExperience</a>	153
<a href="#">Method getGroup</a>	154
<a href="#">Method getGuildNick</a>	154
<a href="#">Method getHealth</a>	155
<a href="#">Method getHealthMax</a>	155
<a href="#">Method getId</a>	155
<a href="#">Method getLastIP</a>	156
<a href="#">Method getLastLogin</a>	156
<a href="#">Method getLevel</a>	157
<a href="#">Method getLookAddons</a>	157
<a href="#">Method getLookBody</a>	157

<a href="#">Method getLookFeet</a>	158
<a href="#">Method getLookHead</a>	158
<a href="#">Method getLookLegs</a>	158
<a href="#">Method getLookType</a>	159
<a href="#">Method getLossExperience</a>	159
<a href="#">Method getLossMana</a>	160
<a href="#">Method getLossSkills</a>	160
<a href="#">Method getMagLevel</a>	160
<a href="#">Method getMana</a>	161
<a href="#">Method getManaMax</a>	161
<a href="#">Method getManaSpent</a>	162
<a href="#">Method getName</a>	162
<a href="#">Method getPosX</a>	162
<a href="#">Method getPosY</a>	163
<a href="#">Method getPosZ</a>	163
<a href="#">Method getPremiumEnd</a>	164
<a href="#">Method getRank</a>	164
<a href="#">Method getRankId</a>	164
<a href="#">Method getRedSkullTime</a>	165
<a href="#">Method getSex</a>	165
<a href="#">Method getSkill</a>	166
<a href="#">Method getSkillTries</a>	166
<a href="#">Method getSlot</a>	167
<a href="#">Method getSoul</a>	167
<a href="#">Method getTownId</a>	168
<a href="#">Method getVocation</a>	168
<a href="#">Method hasRedSkull</a>	168
<a href="#">Method isBanned</a>	169
<a href="#">Method isLoading</a>	169
<a href="#">Method isSaveSet</a>	170
<a href="#">Method load</a>	170
<a href="#">Method save</a>	170
<a href="#">Method setAccount</a>	171
<a href="#">Method setCap</a>	171
<a href="#">Method setConditions</a>	172
<a href="#">Method setCustomField</a>	172
<a href="#">Method setDepot</a>	173
<a href="#">Method setDirection</a>	174
<a href="#">Method setExperience</a>	174
<a href="#">Method setGroup</a>	175
<a href="#">Method setGuildNick</a>	175
<a href="#">Method setHealth</a>	175
<a href="#">Method setHealthMax</a>	176
<a href="#">Method setLastIP</a>	176
<a href="#">Method setLastLogin</a>	177
<a href="#">Method setLevel</a>	177
<a href="#">Method setLookAddons</a>	178
<a href="#">Method setLookBody</a>	178
<a href="#">Method setLookFeet</a>	179



<a href="#">Method setLookHead</a>	179
<a href="#">Method setLookLegs</a>	180
<a href="#">Method setLookType</a>	180
<a href="#">Method setLossExperience</a>	180
<a href="#">Method setLossMana</a>	181
<a href="#">Method setLossSkills</a>	181
<a href="#">Method setMagLevel</a>	182
<a href="#">Method setMana</a>	182
<a href="#">Method setManaMax</a>	183
<a href="#">Method setManaSpent</a>	183
<a href="#">Method setName</a>	184
<a href="#">Method setPosX</a>	184
<a href="#">Method setPosY</a>	184
<a href="#">Method setPosZ</a>	185
<a href="#">Method setPremiumEnd</a>	185
<a href="#">Method setRank</a>	186
<a href="#">Method setRankId</a>	186
<a href="#">Method setRedSkull</a>	187
<a href="#">Method setRedSkullTime</a>	187
<a href="#">Method setSave</a>	188
<a href="#">Method setSex</a>	188
<a href="#">Method setSkill</a>	188
<a href="#">Method setSkillTries</a>	189
<a href="#">Method setSlot</a>	190
<a href="#">Method setSoul</a>	190
<a href="#">Method setTownId</a>	191
<a href="#">Method setVocation</a>	191
<a href="#">Method unban</a>	192
<a href="#">Method unsetRedSkull</a>	192
<a href="#">Method unsetSave</a>	192
<a href="#">Method sleep</a>	193
<a href="#">Class OTS_Players_List</a>	193
<a href="#">Method deletePlayer</a>	193
<a href="#">Method init</a>	194
<a href="#">Class OTS_SQLField</a>	194
<a href="#">Constructor construct</a>	195
<a href="#">Method getName</a>	195
<a href="#">Method getTable</a>	196
<a href="#">Class OTS_SQLFilter</a>	196
<a href="#">Class Constant CRITERIUM_AND</a>	196
<a href="#">Class Constant CRITERIUM_OR</a>	197
<a href="#">Class Constant OPERATOR_EQUAL</a>	197
<a href="#">Class Constant OPERATOR_GREATER</a>	197
<a href="#">Class Constant OPERATOR_LIKE</a>	198
<a href="#">Class Constant OPERATOR_LOWER</a>	198
<a href="#">Class Constant OPERATOR_NEQUAL</a>	198
<a href="#">Class Constant OPERATOR_NGREATER</a>	199
<a href="#">Class Constant OPERATOR_NLIKE</a>	199
<a href="#">Class Constant OPERATOR_NLOWER</a>	199

<a href="#">Method addFilter</a>	200
<a href="#">Method compareField</a>	200
<a href="#">Method getTables</a>	201
<a href="#">Method sleep</a>	201
<a href="#">Method toString</a>	202
<a href="#">Class POT</a>	202
<a href="#">Class Constant BAN_ACCOUNT</a>	203
<a href="#">Class Constant BAN_IP</a>	203
<a href="#">Class Constant BAN_PLAYER</a>	203
<a href="#">Class Constant DB_MYSQL</a>	204
<a href="#">Class Constant DB_ODBC</a>	204
<a href="#">Class Constant DB_PGSQL</a>	204
<a href="#">Class Constant DB_SQLITE</a>	205
<a href="#">Class Constant DEPOT_SID_FIRST</a>	205
<a href="#">Class Constant DIRECTION_EAST</a>	206
<a href="#">Class Constant DIRECTION_NORTH</a>	206
<a href="#">Class Constant DIRECTION_SOUTH</a>	206
<a href="#">Class Constant DIRECTION_WEST</a>	207
<a href="#">Class Constant ORDER_ASC</a>	207
<a href="#">Class Constant ORDER_DESC</a>	207
<a href="#">Class Constant SEX_FEMALE</a>	208
<a href="#">Class Constant SEX_MALE</a>	208
<a href="#">Class Constant SKILL_AXE</a>	208
<a href="#">Class Constant SKILL_CLUB</a>	209
<a href="#">Class Constant SKILL_DISTANCE</a>	209
<a href="#">Class Constant SKILL_FISHING</a>	210
<a href="#">Class Constant SKILL_FIST</a>	210
<a href="#">Class Constant SKILL_SHIELDING</a>	211
<a href="#">Class Constant SKILL_SWORD</a>	211
<a href="#">Class Constant SLOT_AMMO</a>	211
<a href="#">Class Constant SLOT_ARMOR</a>	212
<a href="#">Class Constant SLOT_BACKPACK</a>	212
<a href="#">Class Constant SLOT_FEET</a>	213
<a href="#">Class Constant SLOT_HEAD</a>	213
<a href="#">Class Constant SLOT_LEFT</a>	213
<a href="#">Class Constant SLOT_LEGS</a>	214
<a href="#">Class Constant SLOT_NECKLACE</a>	214
<a href="#">Class Constant SLOT_RIGHT</a>	215
<a href="#">Class Constant SLOT_RING</a>	215
<a href="#">Class Constant VOCATION_DRUID</a>	215
<a href="#">Class Constant VOCATION_KNIGHT</a>	216
<a href="#">Class Constant VOCATION_NONE</a>	216
<a href="#">Class Constant VOCATION_PALADIN</a>	217
<a href="#">Class Constant VOCATION_SORCERER</a>	217
<a href="#">Method banIP</a>	217
<a href="#">Method connect</a>	218
<a href="#">example: connect.php</a>	218
<a href="#">Method createFilter</a>	219
<a href="#">Method createObject</a>	220

<a href="#">Method getDBHandle</a>	220
<a href="#">Method getInstance</a>	221
<a href="#">Method getVocationID</a>	221
<a href="#">Method getVocationName</a>	221
<a href="#">Method getVocationsList</a>	222
<a href="#">Method isIPBanned</a>	222
<a href="#">Method loadClass</a>	223
<a href="#">Method loadVocations</a>	223
<a href="#">Method serverStatus</a>	224
<a href="#">example: example</a>	224
<a href="#">Method setPOTPath</a>	225
<a href="#">example: fakeroot.php</a>	225
<a href="#">Method unbanIP</a>	226
<a href="#">compat.php</a>	228
<a href="#">Appendices</a>	229
<a href="#">Appendix A - Class Trees</a>	230
<a href="#">POT</a>	230
<a href="#">Appendix B - README/CHANGELOG/INSTALL</a>	233
<a href="#">CHANGELOG</a>	234
<a href="#">README</a>	234
<a href="#">INSTALL</a>	236
<a href="#">NEWS</a>	236
<a href="#">Appendix D - Todo List</a>	237

# POT

*This is documenation of POT - official toolkit for [OTServ AAC scripts](#).*

## PHP OTServ Toolkit

There are several reasons why POT was created:

- Just because it was needed - OTServ should have had that long time ago.
- To unify AAC scripts - there are tons of them, and you never know how to write even a single line of code to them as each of them are created different way.
- To provide reliable way of database accessing - most of people who create AAC scripts are (to be honest...) idiots - they don't know what PHP is, how to use it, they just "want to make own AAC script".
- To provide easy interface - people who write in PHP want to write in PHP, not using SQL, XML and many other languages. POT provides abstract PHP interface for data stored in database.

POT has been created for latest SVN release, it will work best with pure SVN servers. However it provides routines to access custom database structure elements. However it won't work with broken database - it relies on database foreign key constraints, triggers etc.

## System requirements

To use POT you need [PHP](#) version at least 5.0 with [PDO extension installed](#) (so it means you will mostly need PHP 5.1, but it is possible to download PDO as external libraries for PHP 5.0.x).

## What POT is

POT is a toolkit/library for accessing OTServ database from PHP. It provides PHP classes that represents OTServ database information as an objects.

## What POT is not

- It is not AAC script - this is a toolkit for making them, but you can't directly run it as website. It has only programming interface.
- It is not application/system framework - you won't create website with only POT. POT has only functionality connected with OTServ database, it doesn't contain for example templates engine. You also won't be able to use it as an ordinary database connection engine - it makes use of [PDO](#) so you can use PDO by itself, POT doesn't provide any additional universal functionality. All it's classes are strictly connected with OTServ database.

## What about XML?

Sorry to say, XML guys - go out. OTServ will never leave XML - it is good to store some flat parts of database there. But not for main database which requires more advanced relationship between data. However of course maybe someone would want to create DB\_XML driver for POT? If you really are a masochist - you're welcome, we will be glad to contribute with you ;).

If you are interested in why XML so sux, and you with it, check out [OTFans thread](#).

## How to use

This is toolkit - set of classes/methods for OTServ database. It abstracts database mechanisms for you so you can work on "physical" PHP objects. But you must know how to use them. This documentation describes some basic steps and toolkit API, but you must know PHP in order to make use of them - the best place to get some knowledge is [PHP manual](#).

Don't copy any of included examples, neither codes provided as examples - they probably won't work. Mainly it's because you have to put your database configuration into them and your script paths. But it's not enough. If you have your own `__autoload()` mechanism you won't be able to just include example codes - you would need to redefine `__autoload()` function, which PHP doesn't allow to (but you should know that very well). Example codes are examples - write your own (if you want them to work the best way for you).

## Link

If you use POT in your script and want to show that you can put this image on your website:

You can use following code for that:

```
1 <a href="http://www.otserv-aac.info/pot/" >
2 
3 </a>
```

# PHP 5.0

*Some things that you should know if you use POT under PHP 5.0.x.*

## PHP 5.0

PHP5 was a huge step in PHP history. It is completely other language than PHP4 (and older versions). POT is written for PHP5 but currently most PHP5 installations are done with PHP 5.1 and higher versions. PHP 5.0 differs from next versions in few details (or even not details, but huge changes, but those mostly doesn't affect POT). There are some important things you should know if you use POT with PHP 5.0.

## PDO

POT requires [PDO extension](#). It is bundled with PHP since 5.1 version. If you use PHP 5.0 you still can install PDO, but you need to do that using [PECL extensions](#). Detailed information about how to do that are in [PHP manual PDO page](#).

## Sub package "compat"

If you use PHP 5.0 you should include special [compatibility assurance library](#). POT uses some mechanisms that exists since PHP 5.1 like [Countable interface](#). It doesn't disallow you using POT with PHP 5.0. Compatibility library will create unexisting interfaces, classes, functions, constants etc. However keep in mind that you won't be able to use PHP 5.1 and newer language mechanisms as it is not possible to redefine PHP behaviour. Here is an example:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // do that before any POT operations!
12 include('compat.php');
13
14 // to not repeat all that stuff
15 include('quickstart.php');
16
17 // STEP 1: no error here - even though we loaded class that implements Countable interface which does not
18 // exists in PHP 5.0 SPL library, because 'compat' library defines it.
19 $list= POT::getInstance()-> createObject('Players_List');
20
21 // STEP 2: we can do that in every version - count() is in fact just a public method
22 echo $list-> count();
23
24 // STEP 3: it won't work correctly in PHP 5.0 - PHP won't call internal count() method of object, will print trivial
25 // count() evaluation result on object
26 echo count( $list);
```

25  
26 ?>

### *Nothin new*

Compatibility library makes you sure, that POT scripts won't cause FATAL errors if you run them on older versions of PHP. However it doesn't introduce any new mechanisms so you won't find anything new in this package. It is safe to include compat.php file even if you work with PHP version 5.1 or newer, but there is no point in doing that.

### `__autoload()`

POT registers own `__autoload()` handler with `spl_autoload_register()`. This function exists since PHP 5.1.2. Compatibility library defines this function as definer of another function - ordinary `__autoload()`. If you have own `__autoload()` function, compat's `spl_autoload_register()` won't redefine `__autoload()` to avoid `E_ERROR`. You then need to bind [POT::loadClass\(\) method](#) to your `__autoload()` function manually.

## Type hinting

In PHP5 new feature was introduced - [type hinting](#). In PHP 5.0 only class names are supported, array type is supported since PHP 5.1. So if you want to use POT with PHP 5.0 you must remove all array hinted types in classes methods.

## What about older PHP versions?

No way. POT was written using new PHP5 object engine - you cant use it with PHP4 and older versions of PHP, PHP/FI.

# POT class preview

*Here main POT class will be described in more guided way.*

## What it is

[POT](#) class is main class of this toolkit. You will access any other classes using this one. It creates for you instances of other classes when you call it's methods and handles class files loading.

## Creating instance of POT class

To get POT object you have to use [POT::getInstance\(\)](#) static method. You should never ever create POT class instances directly! [POT::getInstance\(\)](#) will save static instance and return it globally so you won't need to re-create instances of this class. It is important, as object of this class contains another resources like database connection, or classes directory path so after creating new instance it would not contain them from previous one.

## \_\_autoload() and POT classes

PHP5 provides nice [autoloading mechanism](#). POT makes use of [spl\\_autoload\\_register\(\) function](#) to bind own mechanism with it automatically. If you have your own \_\_autoload function defined, after including POT class you have to register your function with [spl\\_autoload\\_register\(\)](#) as well.

## DAO classes

Key part of this toolbox are Data Access Objects which provides abstraction layer in PHP for plain database data. You create them via main POT class using [createObject\(\) method](#).



# Quick start

*Quick start guide.*

## Putting this all together

To set POT up for using you have to create it's instance and connect to database (it will automatically bind [POT classes loading mechanism](#) to `__autoload()` function. Here is a startup code example:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // binds your __autoload code
12 if( function_exists('__autoload') )
13 {
14     spl_autoload_register('__autoload');
15 }
16
17 // includes POT main file
18 include( './classes/OTS.php' );
19
20 // database configuration - can be simply moved to external file, eg. config.php
21 $config= array(
22     'driver' =>  POT::DB_MYSQL,
23     'host' =>    'localhost',
24     'user' =>    'wrzasq',
25     'database' => 'otserv'
26 );
27
28 // creates POT instance (or get existing one)
29 $ots= POT::getInstance();
30 $ots-> connect(null, $config;
31
32 ?>
```

## Account creation

It is very simple to create account with POT. Here is example code that is self-explainable:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
```

```

9  */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Account object
15 $account= $ots->    createObject('Account');
16
17 // generates new account number
18 $number= $account->    create();
19
20 /*
21 to generate number from 111111 to 999999 use:
22 $number = $account->create(111111, 999999);
23 */
24
25 // sets account info
26 $account->    setPassword('secret');// $account->setPassword( md5('secret') );
27 $account->    setEmail('foo@example.com');
28 $account->    unblock();// remember to unblock!
29 $account->    setPACCDays(0);
30 $account->    save();
31
32 // give user his number
33 echo 'Your account number is: ',    $number
34
35 ?>

```

It is important to remember that [create\(\) method](#) sets `blocked` field of record to true by default, so for smaller projects where you, for example, wouldn't need e-mail activation unblock it after creation.

## Character reading

Here comes also simple example for character search:

```

1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Player object
15 $player= $ots->    createObject('Player');
16
17 // loads player
18 $player->    find('Wrzasq');
19
20 // checks if player exists
21 if( $player->    isLoading() )
22 {
23     // prints character info

```

```

24     echo 'Player \'' . $player> getName() . '\' has ' . $player> getLevel() . ' level.', "\n"
25
26     // example of associated objects retrieving
27     echo 'Player \'' . $player> getName() . '\' is member of ' . $player> getGroup()-> getName() . '
group.', "\n"
28 }
29 else
30 {
31     echo 'Player does not exists.', "\n"
32 }
33
34 ?>

```

## Objects listings

There are also classes for entire sets of records. For each of row classes there is list class. Throught list object you can read single objects and/or delete them from database. Also you can set limitation (for example for pagination). All list classes implements Countable and Iterator interfaces:

```

1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Player object
15 $players= $ots> createObject('Players_List');
16
17 // count of all players - Countable interface implemented
18 echo 'There are ' . count( $players) . ' players in our database.', "\n"
19
20 // sets limitation
21 $players> setLimit(10);
22 $players> setOffset(2);
23
24 // iterates through selected players
25 foreach($playersas $index=> $player)
26 {
27     // each returned item is instance of OTS_Player class
28     echo (2 + $index) . ': ' . $player> getName(), "\n"
29 }
30
31 ?>

```

# DAO objects

*Main part of POT are Data Access Objects objects*

## What are DAO objects?

DAO stands for Data Access Objects. Those are objects which you use mostly - players, accounts, groups, objects lists. They use database resource to fetch/store data and provides you programming interface to access that data without using additional languages like SQL, or XML.

## Why this way?

PHP is a PHP. When you write a code in PHP each element has a meaning. While using SQL you have to use database queries. In code they are simply a strings which doesn't represent any particular data for programming environment. DAO objects wraps database operations in objective aspect, so "dead" string queries becomes a fully functional objects which you can control more strictly, allows you to assign relations and automate some parts.

## Basic operations

Most basic operations are loading, editing and saving data. To see examples of this, see [Quick start guide](#).

## Lists objects

For each table there exist single object class and objects list class. List classes implements [Iterator interface](#) so to list their's content you must use [foreach\(\) loop](#). Each element returned for this loop will be instance of single DAO object. You also use lists to delete items.

## Custom fields

POT was created for basic SVN database structure. However you can access custom fields with POT. You do that with `getCustomField()` and `setCustomField()` methods of DAO objects (single, not lists).

While accessing custom fields you have to remember about using proper PHP types of passed values. POT doesn't know anything about those fields so it uses value type to check the way it should serve it for a query. Don't worry about safety - it doesn't create any hole for SQL injections. But you must remember, that 1 (integer) is not same as '1' (string), or 1.0 (float). POT will quote strings to fit SQL query and to prevent from SQL injections so make sure you [cast](#) your values to type that represents field type to prevent (mainly) from quoting numeric fields.

You should use those methods only to access custom fields that are not accessible through standard POT API. Those methods executes SQL query each time you call them so it would be a huge effectivity loss to access standard fields with `getCustomField()/setCustomField()`.

Also it is important that in difference to fields accessible with standard setters you can set custom field value

on not loaded/saved object. You must either load object from database, or save standard record before using custom fields as they need record primary key assigned to object for queries. Here is an example:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Player object
15 $player= $ots-> createObject('Player');
16
17 // sets basic fields
18 $player-> setName('Wrzasq');
19 $player-> setSex(POT::SEX_MALE);
20 $player-> setVocation(POT::VOCATION_KNIGHT);
21 /* etc... */
22
23 /*
24  this is bad! we can't call this now as we dont have object ID assinged yet
25
26  $player->setCustomField('my_field', 2);
27
28  must save before that to get automatic ID:
29  */
30 $player-> save();
31
32 // now we can call that:
33 // 2 won't be quoted - it's integer
34 $player-> setCustomField('my_field', 2);
35 // 3 will be quoted - '3' is a string!
36 $player-> setCustomField('another_field', '3');
37
38 ?>
```

## Player items

POT provides also objective way of browsing/editing player items (body slots and depot items with all containers). You have [OTS\\_Item](#) and [OTS\\_Container](#) classes for that. OTS\_Item represents single item, OTS\_Container can contain sub-items (either OTS\_Item objects, or next level OTS\_Container objects).

There is important thing to mention - POT doesn't know anything about item types! Items tree only contains item IDs from database, it doesn't load any information from items.otb, nor items.xml files.

Detailed API you will find in documentation of those classes. Here are examples of how you use slot and depot items fetching and saving:

```
1  <?php
2
3  /**
4   * @ignore
```

```

5  * @package examples
6  * @author Wrzasq <wrzasq@gmail.com>
7  * @copyright 2007 (C) by Wrzasq
8  * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9  */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Player object
15 $player= $ots-> createObject('Player');
16 $player-> find('Wrzasq');
17
18 /*
19  Items loading example.
20 */
21
22 // loading item from ammunition slot
23 $item= $player-> getSlot(POT::SLOT_AMMO);
24
25 echo $player-> getName(), ' has item with id ', $item-> getId(), ' in his/her ammo slot.', "\n" ;
26
27 // checks if item is a container
28 if($item instanceof OTS_Container)
29 {
30     // list backpack content
31     foreach($item as $inside)
32     {
33         echo 'Container contains item with id ', $inside-> getId(), ' ', "\n" ;
34     }
35 }
36
37 /*
38  Items tree composing example.
39 */
40
41 // creates container - here it would be a depot locker (we pass ID of item to create)
42 $container= new OTS_Container(2590);
43
44 // now let's create depot chest
45 $chest= new OTS_Container(2594);
46
47 // let's put chest inside locker
48 $container-> addItem($chest);
49
50 // now let's put something deeper - into the chest
51 $item1= new OTS_Item(3015);
52 $chest-> addItem($item1);
53
54 // and more...
55 $item2= new OTS_Item(3013);
56 $chest-> addItem($item2);
57
58 // let's set count for an item
59 $item2-> setCount(2);
60
61 /*
62  Here is a tree of items which we created:
63

```

```

64 $container [depot locker]
65 `-- $chest [depot chest]
66     |-- $item1 [first item inserted into chest]
67     `-- $item2 [second item inserted into chest] count=2
68 */
69
70 /*
71     Items saving example.
72 */
73
74 // now we simply put those items into players depot (2 is depot ID)
75 $player-> setDepot(2, $container);
76
77 ?>

```

Important thing - OTS\_Container class is subclass of OTS\_Item. Each container is also an item.

# Guilds

*Guilds system basics.*

## Baiscs

Like for most other data types, for guilds and ranks there are two kinds of classes - single object class and list class. For guilds those are [OTS\\_Guild](#) and [OTS\\_Guilds\\_List](#), for ranks - [OTS\\_GuildRank](#) and [OTS\\_GuildRanks\\_List](#).

## Guild management

Listing guilds is simple so there is no need to explain it more. More complex is listing guild members. Guild membership is not assigned directly - it is done throught guild ranks. To list guild members you first need to list it's ranks. Here is an example solution to list members in oryiginal Tibia-like way:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // loads guild
15 $guild= $ots-> createObject('Guild');
16 $guild-> load(1);
17
18 $color= '#FFFFCC';
19
20 echo '<h1>Members of ' , htmlspecialchars( $guild->getName() ), '</h1>' ;
21
22 ?>
23 <table>
24     <thead>
25         <tr>
26             <th>Rank</th>
27             <th>Members</th>
28         </tr>
29     </thead>
30     <tbody>
31 <?php
32
33 // lists members of all ranks
34 foreach( $guild-> getGuildRanks()as $guildRank)
35 {
36     // display rank in first row
37     $first= true;
```



```

38 // switches rank rows color
39 $color= $color== '#FFFFCC' ? '#FFCCFF' : '#FFFFCC';
40
41 // list members of this rank
42 foreach( $guildRank> getPlayers() as $player)
43 {
44     echo '<tr style="background-color: ' . $color
45     <td>' . $first?htmlspecialchars( $guildRank> getName() ) : ", '</td>
46     <td>' . $player> getName(), '</td>
47 </tr>' ;
48     $first= false;
49 }
50 }
51
52 ?>
53 </tbody>
54 </table>

```

# Guild action drivers

*Handling invites/requests system for guilds.*

## How does it work?

OTServ database contains all guilds contents. But it is very common in AAC world to create invites system (or also requests system, but invitations are more common). It is not provided by standard OTServ database, though nearly all AAC scripts contains such mechanisms. POT classes allows you to set own drivers for invitations and requests to extend basic OTS functionality.

You have to write a driver class and assign it's object to guild object - then guild object will call requested actions on driver which will execute action code dependent on your script.

## Driver structure

Both invites and requests drivers are similar - they must implement [IOTS\\_GuildAction interface](#). When the driver is assigned to guild object, each time a method of [OTS\\_Guild](#) object is called, it will forward this to action driver.

## Sample driver

Driver implements your logic for invites (or membership requests). Here is sample code that you can base on:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 /**
15  POT guilds invites driver.
16  */
17
18 /**
19  * @ignore
20  */
21 class InvitesDriver implements IOTS_GuildAction
22 {
23     // assigned guild
24     private $guild;
25
26     // initializes driver
27     public function __construct(OTS_Guild $guild)
28     {
```

```

29     $this> guild= $guild
30     // this line automates the process - you can call it manually from outside, but why?
31     $this> guild> setInvitesDriver($this);
32 }
33
34 // returns all invited players to current guild
35 public functionlistRequests()
36 {
37     $invites= array();
38
39     /* here you must create OTS_Player object for each invited player */
40
41     return$invites
42 }
43
44 // invites player to current guild
45 public functionaddRequest(OTS_Player $player)
46 {
47     /* here you must save invitation for given player */
48 }
49
50 // un-invites player
51 public functiondeleteRequest(OTS_Player $player)
52 {
53     /* here you must delete invitation for given player */
54 }
55
56 // commits invitation
57 public functionsubmitRequest(OTS_Player $player)
58 {
59     $rank= null;
60
61     // finds normal member rank
62     foreach( $this> guild> getGuildRanks(as $guildRank
63     {
64         if( $guildRank> getLevel() == 1)
65         {
66             $rank= $guildRank
67             break;
68         }
69     }
70
71     $player> setRank($rank);
72     $player> save();
73
74     // clears invitation
75     $this> deleteRequest($player);
76 }
77 }
78
79 /*
80 Parts of this class driver has been taken from OTSCMS (http://otscms.sourceforge.net/) project source
81 code.
82 */
83 // loads player wiht ID 1
84 $player= $ots> createObject('Player');
85 $player> load(1);
86

```

```
87 // loads guild with ID 1
88 $guild= $ots->    createObject('Guild');
89 $guild->    load(1);
90
91 // creates invitation logic driver for your implementation for current guild
92 new InvitesDriver( $guild);
93
94 // note that you call guild method!
95 $guild->    invite($player);
96
97 ?>
```

# Account number hack

*Example code of how to use prepared account number instead of random.*

## Walkaround

POT always generates random account number - [it is the way your script should work](#). It is done that way with premeditation. However you can walk around it with simple code:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // your non-random number
15 $number= 123456;
16
17 // creates new OTS_Account object
18 $account= $ots->createObject('Account');
19 $account->load($number);
20
21 // number is busy
22 if( $account->isLoaded() )
23 {
24     echo 'Account number ', $number, 'is used.', "\n" ;
25 }
26 // it is not
27 else
28 {
29     // generate number from exactly $number - $number range
30     $number= $account->create($number, $number);
31     echo 'Your account number is: ', $number, "\n" ;
32 }
33
34 ?>
```

# Server online status

*This tutorial will describe how to test server status with POT.*

## Such a simple way

[POT class](#) contains [serverStatus\(\) method](#) which sends 'info' packet to OTS and handles results. It returns object of class [OTS\\_InfoRespond](#) which provides access methods for all OTServ respond info. It will return false if server is offline. Here is a simple example of this method usage:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // server and port
15 $server= '127.0.0.1';
16 $port= 7171;
17
18 // queries server of status info
19 $status= $ots-> serverStatus($server, $port);
20
21 // offline
22 if(!$status)
23 {
24     echo 'Server ', $server, ' is offline.', "\n" ;
25 }
26 // displays various info
27 else
28 {
29     echo 'Server name: ', $status-> getName(), "\n" ;
30     echo 'Server owner: ', $status-> getOwner(), "\n" ;
31     echo 'Players online: ', $status-> getOnlinePlayers(), "\n" ;
32     echo 'Maximum allowed number of players: ', $status-> getMaxPlayers(), "\n" ;
33     echo 'Required client version: ', $status-> getClientVersion(), "\n" ;
34     echo 'All monsters: ', $status-> getMonstersCount(), "\n" ;
35     echo 'Server message: ', $status-> getMOTD(), "\n" ;
36 }
37
38 ?>
```

## DOM way

In case you would want to use this method for some non-SVN server which contains custom fields in respond packet you can still use it. OTS\_InfoRespond class is child of DOMDocument class and doesn't overwrite it's

interface neither behaviour in any way. Returned object is standard DOM document so you can work with it in standard DOM-way.





# Package POT Procedural Elements

## E\_OTS\_NoDriver.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com) >
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

# E\_OTS\_NotLoaded.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.3
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.3
- **License** [GNU Lesser General Public License, Version 3](#)

# IOTS\_DAO.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# IOTS\_DB.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# IOTS\_GuildAction.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS.php

**This file contains main toolkit class.**

This file contains main toolkit class. Please read README file for quick startup guide and/or tutorials for more info.

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 0.1.0: Get rid of POT::getInstance()->create\*() calls - use POT::getInstance()->getDBHandle() in constructors.
- **TODO** 0.1.0: Items list (items.xml + items.otb -> cache).
- **TODO** 0.1.0: Implement \_\_get()/\_\_set()/\_\_call()/\_\_toString(); ArrayAccess interface.
- **TODO** 1.0.0: More detailed documentation.
- **TODO** 1.0.0: Complete phpUnit test.
- **TODO** 0.0.6: Spawns support (OTBM support -> cache).
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Account.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Accounts\_List.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)



# OTS\_Base\_DAO.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.5
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Base\_List.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.5
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Container.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.3
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.3
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_DB\_MySQL.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com) >
- **Version** 0.0.5
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_DB\_ODBC.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com) >
- **Version** 0.0.5
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_DB\_PostgreSQL.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_DB\_SQLite.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com) >
- **Version** 0.0.5
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Group.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)



# OTS\_Groups\_List.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Guild.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_GuildRank.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_GuildRanks\_List.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Guilds\_List.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_InfoRespond.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.2
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.2
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Item.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.3
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.3
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_Player.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)



# OTS\_Players\_List.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_SQLField.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.5
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_SQLFilter.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.5
- **License** [GNU Lesser General Public License, Version 3](#)

# OTS\_SQLite\_Results.php

- **Package** POT
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

# Package POT Classes

## Class E\_OTS\_NoDriver

*[line 20]*

**Occurs when code attempts to execute driven action that has no assigned driver to handle it.**

Occurs when code attempts to execute driven action that has no assigned driver to handle it.

- **Package** POT
- **Version** 0.0.4
- **Since** 0.0.4

## Class E\_OTS\_NotLoaded

*[line 20]*

**Occurs when code attempts to access property of not loaded object.**

Occurs when code attempts to access property of not loaded object.

- **Package** POT

- **Version** 0.0.3
- **Since** 0.0.3

## Class IOTS\_DAO

*[line 22]*

### OTServ database object.

OTServ database object.

This interface indicates that class is a OTServ DAO class.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.1

Constructor *void* function IOTS\_DAO::\_\_construct(\$db) *[line 30]*

#### **Function Parameters:**

- *PDO* **\$db** Database connection object.

### DAO objects must be initialized with a database.

DAO objects must be initialized with a database.

- **Version** 0.0.5
- **Deprecated** 0.0.5 This constructor convention won't be part of interface in future.
- **Since** 0.0.1

- **Access** public

## Class IOTS\_DB

*[line 25]*

### OTServ database handler interface.

OTServ database handler interface.

This interface specifies routines requires by DAO classes.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.1
- **Deprecated** 0.0.5 Don't rely on this interface - it is for backward compatibility only. Check POT instance instead.
- **Since** 0.0.1

Constructor *void* function IOTS\_DB::\_\_construct(\$params) *[line 32]*

#### **Function Parameters:**

- *array* **\$params** Connection configuration.

### Connection parameters.

Connection parameters.

- **Version** 0.0.1

- **Since** 0.0.1
- **Access** public

*string* function IOTS\_DB::fieldName(\$name) [*line 40*]

**Function Parameters:**

- *string* **\$name** Field name.

**Query-quoted field name.**

Query-quoted field name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*int* function IOTS\_DB::lastInsertId() [*line 67*]

**ID of last created record.**

ID of last created record.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*string* function IOTS\_DB::limit([\$limit = false], [\$offset = false]) [*line 75*]

**Function Parameters:**

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).



- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

### **LIMIT/OFFSET clause for queries.**

LIMIT/OFFSET clause for queries.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*mixed* function IOTS\_DB::SQLquery(\$query) [*line 61*]

#### **Function Parameters:**

- *string* **\$query** Database query.

### **Evaluates query.**

Evaluates query.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*string* function IOTS\_DB::SQLquote(\$value) [*line 54*]

#### **Function Parameters:**

- *string* **\$value** Value to be quoted to be suitable for database query.

### **Query-quoted string value.**

Query-quoted string value.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*string* function IOTS\_DB::tableName(\$name) [*line 47*]

**Function Parameters:**

- *string* **\$name** Table name.

**Query-quoted table name.**

Query-quoted table name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

## Class IOTS\_GuildAction

[*line 32*]

**Guild action interface.**

Guild action interface.

This interface indicates that class can handle OTServ guild action.

You can use it for example to handle invites or membership requests.

If you want to serialise (for example save in session) your guild objects with assigned drivers you need to implement also `__sleep()` and `__wakeup()` methods in your drivers, as assigned drivers are also serialised.

- **Package** POT
- **Version** 0.0.4
- **Since** 0.0.4

Constructor *void* function IOTS\_GuildAction::\_\_construct(\$guild) [*line 41*]

**Function Parameters:**

- [\*OTS\\_Guild\*](#) **\$guild** Guild that this driver is assigned to.

**Objects are initialized with a guild that they are assigned to.**

Objects are initialized with a guild that they are assigned to.

It is recommended that your implementations calls assignment functions of \$guild to automatically assign itself as action handler.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void* function IOTS\_GuildAction::addRequest(\$player) [*line 54*]

**Function Parameters:**

- [\*OTS\\_Player\*](#) **\$player** Player which is object of request.

### Adds new request.

Adds new request.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void* function IOTS\_GuildAction::deleteRequest(\$player) [*line 60*]

#### **Function Parameters:**

- [\*OTS\\_Player\*](#) **\$player** Player which is object of request.

### Deletes request.

Deletes request.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*array* function IOTS\_GuildAction::listRequests() [*line 48*]

### List of saved pending actions.

List of saved pending actions.

- **Version** 0.0.4
- **Since** 0.0.4

- **Access** public

*void* function IOTS\_GuildAction::submitRequest(\$player) [*line 66*]

**Function Parameters:**

- [\*OTS\\_Player\*](#) **\$player** Player which is object of request.

**Finalizes request.**

Finalizes request.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

## Class OTS\_Account

[*line 22*]

**OTServ account abstraction.**

OTServ account abstraction.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1

*void* function OTS\_Account::ban([\$time = 0]) [*line 466*]

**Function Parameters:**

- *int* **\$time** Time for time until expires (0 - forever).

**Bans current account.**

Bans current account.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*void* function OTS\_Account::block() [*line 312*]

**Blocks account.**

Blocks account.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Account::count() [*line 557*]

**Returns number of player within.**

Returns number of player within.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Account::create([\$min = 1], [\$max = 9999999]) [*line 51*]  
**account.php**

```

1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Account object
15 $account = $ots->createObject('Account');
16
17 // generates new account number
18 $number = $account->create();
19
20 /*
21  to generate number from 111111 to 999999 use:
22  $number = $account->create(111111, 999999);
23  */
24
25 // sets account info
26 $account->setPassword('secret'); // $account->setPassword( md5('secret') );
27 $account->setEMail('foo@example.com');
28 $account->unblock(); // remember to unblock!
29 $account->setPACCDays(0);
30 $account->save();
31
32 // give user his number
33 echo 'Your account number is: ', $number;
34
35 ?>

```

#### **Function Parameters:**

- *int* **\$min** Minimum number.
- *int* **\$max** Maximum number.

#### **Creates new account.**

Creates new account.

Create new account in given range (1 - 9999999 by default).

Remember! This method sets blocked flag to true after account creation!

IMPORTANT: Since 0.0.4 there is group\_id field which this method does not support. Account's group\_id is set to first one found in database. You should use [createEx\(\)](#) method if you want to set group\_id field during creation.

- **Version** 0.0.4
- **Version** 0.0.1
- **Throws** Exception When there are no free account numbers.
- **Since** 0.0.1
- **Access** public
- **Example**

*int* function OTS\_Account::createEx(\$group, [\$min = 1], [\$max = 9999999]) [*line 77*]

## account.php

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Account object
15 $account = $ots-> createObject('Account');
16
17 // group for account
18 $group = $ots-> createObject('Group');
19
20 // loads group with id 1
21 $group-> load(1);
22
23 // generates new account number
24 $number = $account-> createEx($group);
25
26 // give user his number
27 echo 'Your account number is: ', $number;
28
29 ?>
```

### Function Parameters:



- [\*OTS\\_Group\*](#) **\$group** Group to be assigned to account.
- *int* **\$min** Minimum number.
- *int* **\$max** Maximum number.

### **Creates new account.**

Creates new account.

Create new account in given range (1 - 9999999 by default) in given group.

Remember! This method sets blocked flag to true after account creation!

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** Exception When there are no free account numbers.
- **Since** 0.0.1
- **Since** 0.0.4
- **Access** public
- **Example**

*void* function OTS\_Account::delete() [*line 520*]

### **Deletes account.**

Deletes account.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.5

- **Since** 0.0.1
- **Access** public

*void* function OTS\_Account::find(\$email) [*line 144*]

**Function Parameters:**

- *string* **\$email** Account's e-mail address.

**Loads account by it's e-mail address.**

Loads account by it's e-mail address.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public

*string* function OTS\_Account::getCustomField(\$field) [*line 359*]

**Function Parameters:**

- *string* **\$field** Field name.

**Reads custom field.**

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*string* function OTS\_Account::getEmail() [*line 264*]

**E-mail address.**

E-mail address.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Access** public

*OTS\_Group* function OTS\_Account::getGroup() [*line 208*]

**Returns group of this account.**

Returns group of this account.

- **Version** 0.0.4
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.4
- **Since** 0.0.1

- **Access** public

*int* function OTS\_Account::getId() [*line 190*]

**Account number.**

Account number.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Access** public

*Iterator* function OTS\_Account::getIterator() [*line 544*]

**Returns players iterator.**

Returns players iterator.

There is no need to implement entire Iterator interface since we have [players list class](#) for it.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Account::getPACCDays() [*line 325*]

**PACC days.**

PACC days.

- **Version** 0.0.4
- **Version** 0.0.1
- **Deprecated** 0.0.3 There is no more premdays field in accounts table.
- **Since** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Access** public

*string* function OTS\_Account::getPassword() [*line 237*]

#### **Account's password.**

Account's password.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Access** public

*array* function OTS\_Account::getPlayers() [*line 409*]

#### **List of characters on account.**

List of characters on account.

- **Version** 0.0.5
- **Version** 0.0.1
- **Deprecated** 0.0.5 Use getPlayersList().

- **Since** 0.0.1
- **Throws** `E_OTS_NotLoaded` If account is not loaded.
- **Access** public

*OTS\_Players\_List* function `OTS_Account::getPlayersList()` [*line 439*]

#### **List of characters on account.**

List of characters on account.

In difference to [getPlayers\(\) method](#) this method returns filtered [OTS\\_Players\\_List](#) object instead of array of [OTS\\_Player](#) objects. It is more effective since `OTS_Player_List` doesn't perform all rows loading at once.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** `E_OTS_NotLoaded` If account is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

*bool* function `OTS_Account::isBanned()` [*line 501*]

#### **Checks if account is banned.**

Checks if account is banned.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*bool* function OTS\_Account::isBlocked() [*line 291*]

**Checks if account is blocked.**

Checks if account is blocked.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Access** public

*bool* function OTS\_Account::isLoading() [*line 161*]

**Checks if object is loaded.**

Checks if object is loaded.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Account::load(\$id) [*line 131*]

**Function Parameters:**

- *int* **\$id** Account number.

**Loads account with given number.**

Loads account with given number.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Account::save() [line 172]*

#### **Updates account in database.**

Updates account in database.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded False if account doesn't have ID assigned.
- **Since** 0.0.1
- **Access** public

*void function OTS\_Account::setCustomField(\$field, \$value) [line 385]*

#### **Function Parameters:**

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

#### **Writes custom field.**

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field name. It is safe - it makes you sure that no unproper queries that could



lead to SQL injection will be executed, but it can make your code working wrong way. For example: `$object->setCustomField('foo', '1');` will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** `E_OTS_NotLoaded` If account is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*void function OTS\_Account::setEMail(\$email) [line 279]*

**Function Parameters:**

- *string* **\$email** E-mail address.

**Sets account's email.**

Sets account's email.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Account::setGroup(\$group) [line 225]*

**Function Parameters:**

- [OTS\\_Group](#) **\$group** Group to be a member.

### **Assigns account to group.**

Assigns account to group.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

`void function OTS_Account::setPACCDays($premdays, $pacc) [line 342]`

#### ***Function Parameters:***

- *int* **\$pacc** PACC days.
- **\$premdays**

### **Sets PACC days count.**

Sets PACC days count.

- **Version** 0.0.4
- **Version** 0.0.1
- **Deprecated** 0.0.3 There is no more premdays field in accounts table.
- **Since** 0.0.1
- **Access** public

`void function OTS_Account::setPassword($password) [line 252]`

#### ***Function Parameters:***

- *string* **\$password** Password.

### **Sets account's password.**

Sets account's password.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Account::unban() [line 483]*

### **Deletes ban from current account.**

Deletes ban from current account.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*void function OTS\_Account::unblock() [line 304]*

### **Unblocks account.**

Unblocks account.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

## Class OTS\_Accounts\_List

*[line 21]*

### List of accounts.

List of accounts.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.1

*void function OTS\_Accounts\_List::deleteAccount(\$account) [line 30]*

#### **Function Parameters:**

- [OTS Account](#) **\$account** Account to be deleted.

### Deletes account.

Deletes account.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS\_Account->delete().
- **Since** 0.0.1
- **Access** public

*void function OTS\_Accounts\_List::init() [line 43]*

### **Sets list parameters.**

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

## **Class OTS\_Base\_DAO**

*[line 20]*

### **Basic data access object routines.**

Basic data access object routines.

- **Package** POT
- **Version** 0.0.5
- **Abstract Element**
- **Since** 0.0.5

### **OTS\_Base\_DAO::\$db**

*PDO = [line 27]*

### **Database connection.**

Database connection.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** protected

Constructor *void* function OTS\_Base\_DAO::\_\_construct(\$db) [*line 34*]

**Function Parameters:**

- *PDO* **\$db** Database connection object.

**Sets database connection handler.**

Sets database connection handler.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void* function OTS\_Base\_DAO::\_\_clone() [*line 71*]

**Creates clone of object.**

Creates clone of object.

Copy of object needs to have different ID.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_DAO::\_\_set\_state(\$properties) [line 84]*

**Function Parameters:**

- **array \$properties** List of object properties.

**Magic PHP5 method.**

Magic PHP5 method.

Allows object importing from [var\\_export\(\)](#).

- **Version** 0.0.5
- **Static**
- **Since** 0.0.5
- **Access** public

*array function OTS\_Base\_DAO::\_\_sleep() [line 47]*

**Magic PHP5 method.**

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_DAO::\_\_wakeup() [line 59]*

**Magic PHP5 method.**

Magic PHP5 method.

Allows object unserialisation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

## Class OTS\_Base\_List

[line 20]

**Basic list class routines.**  
Basic list class routines.

- **Package** POT
- **Version** 0.0.5
- **Abstract Element**
- **Since** 0.0.5

**OTS\_Base\_List::\$class**

*string =* [line 76]

**Class of generated objects.**  
Class of generated objects.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** protected



## OTS\_Base\_List::\$table

*string = [line 69]*

### Default table name for queries.

Default table name for queries.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** protected

Constructor *void* function OTS\_Base\_List::\_\_construct(\$db) *[line 83]*

#### **Function Parameters:**

- **PDO \$db** Database connection object.

### Sets database connection handler.

Sets database connection handler.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*int* function OTS\_Base\_List::count() *[line 237]*

### Returns number of accounts on list in current criterium.

Returns number of accounts on list in current criterium.

- **Version** 0.0.5
- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*LOTS\_DAO* function *OTS\_Base\_List::current()* [*line 186*]

**Returns current row.**

Returns current row.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void* function *OTS\_Base\_List::init()* [*line 92*]

**Sets list parameters.**

Sets list parameters.

- **Version** 0.0.5
- **Abstract Element**
- **Since** 0.0.5
- **Access** public

*mixed* function *OTS\_Base\_List::key()* [*line 216*]

**Current cursor position.**

Current cursor position.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_List::next() [line 206]*

**Moves to next row.**

Moves to next row.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_List::orderBy(\$field, [\$order = POT::ORDER\_ASC], \$field) [line 269]*

**Function Parameters:**

- *string* **\$field** Field name.
- *int* **\$order** Sorting order (ascending by default).
- **\$field**

**Appends sorting rule.**

Appends sorting rule.

- **Version** 0.0.5

- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_List::resetOrder() [line 258]*

**Clears ORDER BY clause.**

Clears ORDER BY clause.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_List::rewind() [line 198]*

**Select rows from database.**

Select rows from database.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_List::setFilter([\$filter = null]) [line 250]*

**Function Parameters:**

- [OTS\\_SQLFilter](#)*|null* **\$filter** Filter for list.

**Sets filter on list.**

Sets filter on list.

Call without argument to reset filter.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_List::setLimit([\$limit = false]) [line 152]*

***Function Parameters:***

- *int/bool* **\$limit** Limit for SELECT (false to reset).

**Sets LIMIT.**

Sets LIMIT.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_List::setOffset([\$offset = false]) [line 169]*

***Function Parameters:***

- *int/bool* **\$offset** Offset for SELECT (false to reset).

**Sets OFFSET.**

Sets OFFSET.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*bool* function OTS\_Base\_List::valid() [*line 226*]

**Checks if there are any rows left.**

Checks if there are any rows left.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void* function OTS\_Base\_List::\_\_set\_state(\$properties) [*line 127*]

**Function Parameters:**

- *array* **\$properties** List of object properties.

**Magic PHP5 method.**

Magic PHP5 method.

Allows object importing from [var\\_export\(\)](#).

- **Version** 0.0.5
- **Static**
- **Since** 0.0.5
- **Access** public

*array* function OTS\_Base\_List::\_\_sleep() [*line 102*]

### **Magic PHP5 method.**

Magic PHP5 method.  
Allows object serialisation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void function OTS\_Base\_List::\_\_wakeup() [line 114]*

### **Magic PHP5 method.**

Magic PHP5 method.  
Allows object unserialisation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

## **Class OTS\_Container** *[line 20]*

### **Container item representation.**

Container item representation.

- **Package** POT

- **Version** 0.0.3
- **Since** 0.0.3

*void* function OTS\_Container::addItem(\$item) [*line 34*]

**Function Parameters:**

- [OTS\\_Item](#) \$item Item.

**Adds item to container.**

Adds item to container.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*int* function OTS\_Container::count() [*line 65*]

**Number of items inside container.**

Number of items inside container.

OTS\_Container implementation of Countable interface differs from OTS\_Item implementation. [OTS\\_Item::count\(\)](#) returns count of given item, OTS\_Container::count() returns number of items inside container. If somehow it would be possible to make container items with more than 1 in one place, you can use [OTS\\_Item::getCount\(\)](#) and [OTS\\_Item::setCount\(\)](#) in code where you are not sure if working with regular item, or container.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public



*OTS\_Item* function OTS\_Container::current() [*line 75*]

**Returns current item.**

Returns current item.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*mixed* function OTS\_Container::key() [*line 93*]

**Current cursor position.**

Current cursor position.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*void* function OTS\_Container::next() [*line 83*]

**Moves to next item.**

Moves to next item.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*void* function OTS\_Container::removeItem(\$item) [*line 46*]

**Function Parameters:**

- [OTS\\_Item](#) \$item Item.

### **Removes given item from current container.**

Removes given item from current container.

Passed item must be exactly instance of item which is stored in container, not it's copy.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*void* function OTS\_Container::rewind() [*line 111*]

### **Resets internal items array pointer.**

Resets internal items array pointer.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*bool* function OTS\_Container::valid() [*line 103*]

### **Checks if there are any items left.**

Checks if there are any items left.

- **Version** 0.0.3
- **Since** 0.0.3

- **Access** public

## Class OTS\_DB\_MySQL

[line 22]

### MySQL connection interface.

MySQL connection interface.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1

Constructor *void* function OTS\_DB\_MySQL::\_\_construct(\$params) [line 49]

#### **Function Parameters:**

- *array* **\$params** Connection parameters.

### **Creates database connection.**

Creates database connection.

Connects to MySQL database on given arguments.

List of parameters for this drivers:

- *host* - database server.
- *port* - port (optional, also it is possible to use host:port in *host* parameter).
- *database* - database name.
- *user* - user login.
- *password* - user password.

- **Version** 0.0.1
- **See** [POT::connect\(\)](#)
- **Since** 0.0.1
- **Access** public

*string* function OTS\_DB\_MySQL::fieldName(\$name) [*line 104*]

**Function Parameters:**

- *string* **\$name** Field name.

**Query-quoted field name.**

Query-quoted field name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*string* function OTS\_DB\_MySQL::limit([\$limit = false], [\$offset = false]) [*line 157*]

**Function Parameters:**

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

**LIMIT/OFFSET clause for queries.**

LIMIT/OFFSET clause for queries.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*PDOStatement* bool function OTS\_DB\_MySQL::SQLquery(\$query) [*line 145*]

**Function Parameters:**

- *string* **\$query** SQL query.

**IOTS\_DB method.**

IOTS\_DB method.

Overwrites PDO method.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Use PDO::query().
- **Since** 0.0.1
- **Access** public

*string* function OTS\_DB\_MySQL::SQLquote(\$string) [*line 130*]

**Function Parameters:**

- *string* **\$string** String to be quoted.

**IOTS\_DB method.**

IOTS\_DB method.

Overwrites PDO method - we won't use quoting against other values.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Use PDO::quote().
- **Since** 0.0.1
- **Access** public

*string* function OTS\_DB\_MySQL::tableName(\$name) [*line 115*]

**Function Parameters:**

- *string* **\$name** Table name.

**Query-quoted table name.**

Query-quoted table name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

## Class OTS\_DB\_ODBC

[*line 22*]

**ODBC connection interface.**

ODBC connection interface.

- **Package** POT
- **Version** 0.0.5

- **Version** 0.0.4
- **Since** 0.0.4

Constructor *void* function `OTS_DB_ODBC::__construct($params)` [*line 49*]

**Function Parameters:**

- *array* **\$params** Connection parameters.

**Creates database connection.**

Creates database connection.

Connects to ODBC data source on given arguments.

List of parameters for this drivers:

- *host* - database host.
- *port* - ODBC driver.
- *database* - database name.
- *user* - user login.
- *password* - user password.

- **Version** 0.0.4
- **See** [POT::connect\(\)](#)
- **Since** 0.0.4
- **Access** public

*string* function `OTS_DB_ODBC::fieldName($name)` [*line 97*]

**Function Parameters:**

- *string* **\$name** Field name.

**Query-quoted field name.**

Query-quoted field name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*string* function OTS\_DB\_ODBC::limit([\$limit = false], [\$offset = false]) [*line 150*]

**Function Parameters:**

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

**LIMIT/OFFSET clause for queries.**

LIMIT/OFFSET clause for queries.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*PDOStatement|bool* function OTS\_DB\_ODBC::SQLquery(\$query) [*line 138*]

**Function Parameters:**

- *string* **\$query** SQL query.

**IOTS\_DB method.**

IOTS\_DB method.

Overwrites PDO method.



- **Version** 0.0.4
- **Deprecated** 0.0.5 Use PDO::query().
- **Since** 0.0.4
- **Access** public

*string* function OTS\_DB\_ODBC::SQLquote(\$string) [*line 123*]

**Function Parameters:**

- *string* **\$string** String to be quoted.

**IOTS\_DB method.**

IOTS\_DB method.

Overwrites PDO method - we won't use quoting against other values.

- **Version** 0.0.4
- **Deprecated** 0.0.5 Use PDO::quote().
- **Since** 0.0.4
- **Access** public

*string* function OTS\_DB\_ODBC::tableName(\$name) [*line 108*]

**Function Parameters:**

- *string* **\$name** Table name.

**Query-quoted table name.**

Query-quoted table name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

## Class OTS\_DB\_PostgreSQL

[line 22]

**PostgreSQL connection interface.**

PostgreSQL connection interface.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4

Constructor *void* function OTS\_DB\_PostgreSQL::\_\_construct(\$params) [line 49]

**Function Parameters:**

- *array* **\$params** Connection parameters.

**Creates database connection.**

Creates database connection.

Connects to PgSQL database on given arguments.

List of parameters for this drivers:

- *host* - database server.
- *port* - port (optional, also it is possible to use host:port in *host* parameter).

- *database* - database name.
- *user* - user login.
- *password* - user password.

- **Version** 0.0.4
- **See** [POT::connect\(\)](#)
- **Since** 0.0.4
- **Access** public

*string* function OTS\_DB\_PostgreSQL::fieldName(\$name) [*line 104*]

**Function Parameters:**

- *string* **\$name** Field name.

**Query-quoted field name.**

Query-quoted field name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*string* function OTS\_DB\_PostgreSQL::limit([\$limit = false], [\$offset = false]) [*line 157*]

**Function Parameters:**

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

## LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*PDOStatement|bool* function OTS\_DB\_PostgreSQL::SQLquery(\$query) [*line 145*]

### **Function Parameters:**

- *string* **\$query** SQL query.

## IOTS\_DB method.

IOTS\_DB method.

Overwrites PDO method.

- **Version** 0.0.4
- **Deprecated** 0.0.5 Use PDO::query().
- **Since** 0.0.4
- **Access** public

*string* function OTS\_DB\_PostgreSQL::SQLquote(\$string) [*line 130*]

### **Function Parameters:**

- *string* **\$string** String to be quoted.

## IOTS\_DB method.

IOTS\_DB method.  
Overwrites PDO method - we won't use quoting against other values.

- **Version** 0.0.4
- **Deprecated** 0.0.5 Use PDO::quote().
- **Since** 0.0.4
- **Access** public

*string* function OTS\_DB\_PostgreSQL::tableName(\$name) [*line 115*]  
**Function Parameters:**

- *string* **\$name** Table name.

**Query-quoted table name.**  
Query-quoted table name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

## Class OTS\_DB\_SQLite

[*line 22*]

**SQLite connection interface.**  
SQLite connection interface.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1

Constructor *void* function OTS\_DB\_SQLite::\_\_construct(\$params) [*line 45*]

**Function Parameters:**

- *array* **\$params** Connection parameters.

**Creates database connection.**

Creates database connection.

Connects to SQLite database on given arguments.

List of parameters for this drivers:

- *database* - database name.

- **Version** 0.0.1
- **See** [POT::connect\(\)](#)
- **Since** 0.0.1
- **Access** public

*string* function OTS\_DB\_SQLite::fieldName(\$name) [*line 65*]

**Function Parameters:**

- *string* **\$name** Field name.

## Query-quoted field name.

Query-quoted field name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*string* function OTS\_DB\_SQLite::limit([\$limit = false], [\$offset = false]) [*line 118*]

### **Function Parameters:**

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

## LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*PDOStatement|bool* function OTS\_DB\_SQLite::SQLquery(\$query) [*line 106*]

### **Function Parameters:**

- *string* **\$query** SQL query.

## IOTS\_DB method.

IOTS\_DB method.

Overwrites PDO method.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Use PDO::query().
- **Since** 0.0.1
- **Access** public

*string* function OTS\_DB\_SQLite::SQLquote(\$string) [*line 91*]

**Function Parameters:**

- *string* **\$string** String to be quoted.

**IOTS\_DB method.**

IOTS\_DB method.

Overwrites PDO method - we won't use quoting against other values.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Use PDO::quote().
- **Since** 0.0.1
- **Access** public

*string* function OTS\_DB\_SQLite::tableName(\$name) [*line 76*]

**Function Parameters:**

- *string* **\$name** Table name.

**Query-quoted table name.**



Query-quoted table name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

## Class OTS\_Group

*[line 22]*

**OTServ user group abstraction.**

OTServ user group abstraction.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1

*int* function OTS\_Group::count() *[line 385]*

**Returns number of player within.**

Returns number of player within.

- **Version** 0.0.5
- **Version** 0.0.1

- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Group::delete() [*line 348*]

### **Deletes group.**

Deletes group.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Group::getAccess() [*line 154*]

### **Access level.**

Access level.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

*string* function OTS\_Group::getCustomField(\$field) [*line 241*]

**Function Parameters:**

- *string* **\$field** Field name.

**Reads custom field.**

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Group::getFlags() [*line 127*]

**Rights flags.**

Rights flags.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Group::getId() [*line 83*]

**Group ID.**

Group ID.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

*Iterator* function OTS\_Group::getIterator() [*line 372*]

**Returns players iterator.**

Returns players iterator.

There is no need to implement entire Iterator interface since we have [players list class](#) for it.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Group::getMaxDepotItems() [*line 181*]

**Maximum count of items in depot.**

Maximum count of items in depot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Group::getMaxVIPList() [*line 208*]

**Maximum count of players in VIP list.**

Maximum count of players in VIP list.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

*string* function OTS\_Group::getName() [*line 100*]

**Group name.**

Group name.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

*array* function OTS\_Group::getPlayers() [*line 291*]

**List of characters in given group.**

List of characters in given group.

- **Version** 0.0.5
- **Version** 0.0.1
- **Deprecated** 0.0.5 Use getPlayersList().
- **Since** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Access** public

*OTS\_Players\_List* function OTS\_Group::getPlayersList() [*line 321*]

**List of characters in group.**

List of characters in group.

In difference to [getPlayers\(\) method](#) this method returns filtered [OTS Players List](#) object instead of array of [OTS Player](#) objects. It is more effective since OTS\_Player\_List doesn't perform all rows loading at once.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

*bool* function OTS\_Group::isLoaded() [*line 48*]

### **Checks if object is loaded.**

Checks if object is loaded.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Group::load(\$id) [line 37]*

#### **Function Parameters:**

- *int* **\$id** Group number.

### **Loads group with given id.**

Loads group with given id.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Group::save() [line 58]*

### **Saves account in database.**

Saves account in database.

- **Version** 0.0.5

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS\_Group::setAccess(\$access) [line 169]

**Function Parameters:**

- *int* **\$access** Access level.

**Sets access level.**

Sets access level.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS\_Group::setCustomField(\$field, \$value) [line 267]

**Function Parameters:**

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

**Writes custom field.**

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field name. It is safe - it makes you sure that no improper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: \$object->setCustomField('foo', '1'); will quote 1 as as string ('1') instead of



passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If group is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*void function OTS\_Group::setFlags(\$flags) [line 142]*

***Function Parameters:***

- *int* **\$flags** Flags.

**Sets rights flags.**

Sets rights flags.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Group::setMaxDepotItems(\$maxdepotitems) [line 196]*

***Function Parameters:***

- *int* **\$maxdepotitems** Maximum value.

**Sets maximum count of items in depot.**

Sets maximum count of items in depot.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Group::setMaxVIPList(\$maxviplist, \$maxdepotitems) [line 223]*

***Function Parameters:***

- *int* **\$maxdepotitems** Maximum value.
- **\$maxviplist**

**Sets maximum count of players in VIP list.**

Sets maximum count of players in VIP list.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Group::setName(\$name) [line 115]*

***Function Parameters:***

- *string* **\$name** Name.

**Sets group's name.**

Sets group's name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

## Class OTS\_Groups\_List

*[line 21]*

**List of groups.**  
List of groups.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.1

*void* function OTS\_Groups\_List::deleteGroup(\$group) *[line 30]*

### **Function Parameters:**

- [\*OTS\\_Group\*](#) **\$group** Group to be deleted.

**Deletes group.**  
Deletes group.

- **Version** 0.0.5

- **Deprecated** 0.0.5 Use OTS\_Group->delete().
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Groups\_List::init() [*line 43*]

### **Sets list parameters.**

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

## Class OTS\_Guild

[*line 22*]

### **OTServ guild abstraction.**

OTServ guild abstraction.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4

*void function OTS\_Guild::acceptInvite(\$player) [line 442]*

**Function Parameters:**

- [\*OTS\\_Player\*](#) **\$player** Player to be joined.

**Finalise invitation.**

Finalise invitation.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Throws** E\_OTS\_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::acceptRequest(\$player) [line 534]*

**Function Parameters:**

- [\*OTS\\_Player\*](#) **\$player** Player to be accepted.

**Accepts player.**

Accepts player.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Throws** E\_OTS\_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

*int* function OTS\_Guild::count() [*line 594*]

**Returns number of ranks within.**

Returns number of ranks within.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

*void* function OTS\_Guild::delete() [*line 557*]

**Deletes guild.**

Deletes guild.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

*void* function OTS\_Guild::deleteInvite(\$player) [*line 419*]

**Function Parameters:**

- [\*OTS\\_Player\*](#) **\$player** Player to be un-invited.

### **Deletes invitation for player to guild.**

Deletes invitation for player to guild.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Throws** E\_OTS\_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::deleteRequest(\$player) [line 511]*

#### ***Function Parameters:***

- [OTS\\_Player](#) **\$player** Player to be rejected.

### **Deletes request from player.**

Deletes request from player.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Throws** E\_OTS\_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::find(\$name) [line 114]*

#### ***Function Parameters:***

- *string* **\$name** Guild's name.

### **Loads guild by it's name.**

Loads guild by it's name.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*int* function OTS\_Guild::getCreationData() [*line 235*]

### **Guild creation data.**

Guild creation data.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

*string* function OTS\_Guild::getCustomField(\$field) [*line 267*]

### **Function Parameters:**

- *string* **\$field** Field name.

### **Reads custom field.**

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.



Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTs\_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

*array* function OTS\_Guild::getGuildRanks() [*line 316*]

**Reads all ranks that are in this guild.**

Reads all ranks that are in this guild.

- **Version** 0.0.5
- **Version** 0.0.4
- **Deprecated** 0.0.5 Use getGuildRanksList().
- **Since** 0.0.4
- **Throws** E\_OTs\_NotLoaded If guild is not loaded.
- **Access** public

*OTS\_GuildRanks\_List* function OTS\_Guild::getGuildRanksList() [*line 346*]

**List of ranks in guild.**

List of ranks in guild.

In difference to [getGuildRanks\(\) method](#) this method returns filtered [OTS\\_GuildRanks\\_List](#) object instead of array of [OTS\\_GuildRank](#) objects. It is more effective since OTS\_GuildRanks\_List doesn't perform all rows loading at once.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

*int* function OTS\_Guild::getId() [*line 165*]

#### **Guild ID.**

Guild ID.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

*Iterator* function OTS\_Guild::getIterator() [*line 581*]

#### **Returns ranks iterator.**

Returns ranks iterator.

There is no need to implement entire Iterator interface since we have [ranks list class](#) for it.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.5

- **Since** 0.0.4
- **Access** public

*string* function OTS\_Guild::getName() [*line 181*]

**Guild name.**

Guild name.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

*OTS\_Player* function OTS\_Guild::getOwner() [*line 207*]

**Returns owning player of this player.**

Returns owning player of this player.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

*void* function OTS\_Guild::invite(\$player) [*line 396*]

**Function Parameters:**

- [\*OTS\\_Player\*](#) **\$player** Player to be invited.

**Invites player to guild.**

Invites player to guild.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Throws** E\_OTS\_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

*bool* function OTS\_Guild::isLoaded() [*line 131*]

**Checks if object is loaded.**

Checks if object is loaded.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*array* function OTS\_Guild::listInvites() [*line 373*]

**Returns list of invited players.**

Returns list of invited players.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Throws** E\_OTS\_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

*array function OTS\_Guild::listRequests() [line 465]*

**Returns list of players that requested membership.**

Returns list of players that requested membership.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Throws** E\_OTS\_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::load(\$id) [line 102]*

**Function Parameters:**

- *int* **\$id** Guild's ID.

**Loads guild with given id.**

Loads guild with given id.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::request(\$player) [line 488]*

**Function Parameters:**

- [OTS\\_Player](#) **\$player** Player that requested membership.

### **Requests membership in guild for player player.**

Requests membership in guild for player player.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Throws** E\_OTS\_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::save() [line 141]*

### **Saves guild in database.**

Saves guild in database.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::setCreationData(\$creationdata) [line 250]*

### **Function Parameters:**

- *int* **\$creationdata** Guild creation data.

### **Sets guild creation data.**

Sets guild creation data.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void* function OTS\_Guild::setCustomField(\$field, \$value) [*line 292*]

**Function Parameters:**

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

**Writes custom field.**

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field value. It is safe - it makes you sure that no unproper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: \$object->setCustomField('foo', '1'); will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::setInvitesDriver([\$invites = null]) [line 81]*

**Function Parameters:**

- [\*IOTS\\_GuildAction\*](#) **\$invites** Invites driver (don't pass it to clear driver).

**Assigns invites handler.**

Assigns invites handler.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::setName(\$name) [line 196]*

**Function Parameters:**

- *string* **\$name** Name.

**Sets players's name.**

Sets players's name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::setOwner(\$owner) [line 224]*

**Function Parameters:**



- [OTS\\_Player](#) **\$owner** Owning player.

### **Assigns guild to owner.**

Assigns guild to owner.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::setRequestsDriver([\$requests = null]) [line 91]*

#### **Function Parameters:**

- [IOTS\\_GuildAction](#) **\$requests** Membership requests driver (don't pass it to clear driver).

### **Assigns requests handler.**

Assigns requests handler.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_Guild::\_\_clone() [line 65]*

### **Creates clone of object.**

Creates clone of object.

Copy of object needs to have different ID.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

array function OTS\_Guild::\_\_sleep() [*line 53*]

#### **Magic PHP5 method.**

Magic PHP5 method.  
Allows object serialisation.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

## Class OTS\_GuildRank

[*line 22*]

#### **OTServ guild rank abstraction.**

OTServ guild rank abstraction.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4

*int* function OTS\_GuildRank::count() [*line 356*]

**Returns number of player within.**

Returns number of player within.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

*void* function OTS\_GuildRank::delete() [*line 319*]

**Deletes guild rank.**

Deletes guild rank.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If guild rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

*void* function OTS\_GuildRank::find(\$name, [\$guild = null]) [*line 52*]

**Function Parameters:**

- *string* **\$name** Rank's name.
- [\*OTS\\_Guild\*](#) **\$guild** Guild in which rank should be found.

### Loads rank by it's name.

Loads rank by it's name.

As there can be several ranks with same name in different guilds you can pass optional second parameter to specify in which guild script should look for rank.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*string* function OTS\_GuildRank::getCustomField(\$field) [*line 213*]

#### **Function Parameters:**

- *string* **\$field** Field name.

### Reads custom field.

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

*OTS\_Guild* function OTS\_GuildRank::getGuild() [*line 153*]

**Returns guild of this rank.**

Returns guild of this rank.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

*int* function OTS\_GuildRank::getId() [*line 111*]

**Rank ID.**

Rank ID.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

*Iterator* function OTS\_GuildRank::getIterator() [*line 343*]

**Returns players iterator.**

Returns players iterator.

There is no need to implement entire Iterator interface since we have [players list class](#) for it.

- **Version** 0.0.5
- **Version** 0.0.4

- **Throws** E\_OTS\_NotLoaded If rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

*int* function OTS\_GuildRank::getLevel() [*line 181*]

**Rank's access level.**

Rank's access level.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

*string* function OTS\_GuildRank::getName() [*line 127*]

**Rank name.**

Rank name.

- **Version** 0.0.4
- **Throws** E\_OTS\_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

*array* function OTS\_GuildRank::getPlayers() [*line 262*]

**Reads all players who has this rank set.**

Reads all players who has this rank set.

- **Version** 0.0.5
- **Version** 0.0.4
- **Deprecated** 0.0.5 Use `getPlayersList()`.
- **Since** 0.0.4
- **Throws** `E_OTS_NotLoaded` If rank is not loaded.
- **Access** public

*OTS\_Players\_List* function `OTS_GuildRank::getPlayersList()` [*line 292*]

#### **List of characters with current rank.**

List of characters with current rank.

In difference to [getPlayers\(\) method](#) this method returns filtered [OTS\\_Players\\_List](#) object instead of array of [OTS\\_Player](#) objects. It is more effective since `OTS_Player_List` doesn't perform all rows loading at once.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** `E_OTS_NotLoaded` If rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

*bool* function `OTS_GuildRank::isLoaded()` [*line 77*]

#### **Checks if object is loaded.**

Checks if object is loaded.

- **Version** 0.0.4

- **Since** 0.0.4
- **Access** public

*void function OTS\_GuildRank::load(\$id) [line 37]*

**Function Parameters:**

- *int* **\$id** Rank's ID.

**Loads rank with given id.**

Loads rank with given id.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_GuildRank::save() [line 87]*

**Saves rank in database.**

Saves rank in database.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_GuildRank::setCustomField(\$field, \$value) [line 238]*



### **Function Parameters:**

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

### **Writes custom field.**

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field value. It is safe - it makes you sure that no unproper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: \$object->setCustomField('foo', '1'); will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E\_OTs\_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

*void* function OTS\_GuildRank::setGuild(\$guild) [*line 170*]

### **Function Parameters:**

- [\*OTS\\_Guild\*](#) **\$guild** Owing guild.

### **Assigns rank to guild.**

Assigns rank to guild.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_GuildRank::setLevel(\$level) [line 196]*

**Function Parameters:**

- *int* **\$level** access level within guild.

**Sets rank's access level within guild.**

Sets rank's access level within guild.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

*void function OTS\_GuildRank::setName(\$name) [line 142]*

**Function Parameters:**

- *string* **\$name** Name.

**Sets rank's name.**

Sets rank's name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

## Class OTS\_GuildRanks\_List

[line 21]

### List of guild ranks.

List of guild ranks.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.4

void function OTS\_GuildRanks\_List::deleteGuildRank(\$guildRank) [line 30]

#### **Function Parameters:**

- [OTS\\_GuildRank](#) **\$guildRank** Rank to be deleted.

### Deletes guild rank.

Deletes guild rank.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS\_GuildRank->delete().
- **Since** 0.0.4
- **Access** public

*void function OTS\_GuildRanks\_List::init() [line 43]*

### **Sets list parameters.**

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.4
- **Since** 0.0.5
- **Access** public

## Class OTS\_Guilds\_List

*[line 21]*

### **List of guilds.**

List of guilds.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.4

*void function OTS\_Guilds\_List::deleteGuild(\$guild) [line 30]*

### **Function Parameters:**

- [OTS\\_Guild](#) **\$guild** Guild to be deleted.

### **Deletes guild.**

Deletes guild.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS\_Guild->delete().
- **Since** 0.0.4
- **Access** public

*void* function OTS\_Guilds\_List::init() [*line 43*]

#### **Sets list parameters.**

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.4
- **Since** 0.0.5
- **Access** public

## Class OTS\_InfoRespond

[*line 22*]

#### **Wrapper for 'info' respond's DOMDocument.**

Wrapper for 'info' respond's DOMDocument.

Note: as this class extends DOMDocument class and contains exactly respond XML tree you can work on it as on normal DOM tree.

- **Package** POT
- **Version** 0.0.2
- **Since** 0.0.2

*string* function OTS\_InfoRespond::getClientVersion() [*line 121*]

**Returns dedicated version of client.**

Returns dedicated version of client.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getEmail() [*line 141*]

**Returns owner e-mail.**

Returns owner e-mail.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getIP() [*line 49*]

**Returns server IP.**

Returns server IP.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getLocation() [*line 79*]

**Returns server location.**

Returns server location.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getMapAuthor() [*line 202*]

**Returns map author.**

Returns map author.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*int* function OTS\_InfoRespond::getMapHeight() [*line 222*]

**Returns map height.**

Returns map height.

- **Version** 0.0.2

- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getMapName() [*line 191*]

**Returns map name.**

Returns map name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*int* function OTS\_InfoRespond::getMapWidth() [*line 212*]

**Returns map width.**

Returns map width.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*int* function OTS\_InfoRespond::getMaxPlayers() [*line 161*]

**Returns maximum amount of players online.**

Returns maximum amount of players online.

- **Version** 0.0.2
- **Since** 0.0.2



- **Access** public

*int* function OTS\_InfoRespond::getMonstersCount() [*line 181*]

**Returns number of all monsters on map.**

Returns number of all monsters on map.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getMOTD() [*line 232*]

**Returns server's Message Of The Day**

Returns server's Message Of The Day

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getName() [*line 59*]

**Returns server name.**

Returns server name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*int* function OTS\_InfoRespond::getOnlinePlayers() [*line 151*]

**Returns current amount of players online.**

Returns current amount of players online.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getOwner() [*line 131*]

**Returns owner name.**

Returns owner name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*int* function OTS\_InfoRespond::getPlayersPeak() [*line 171*]

**Returns record of online players.**

Returns record of online players.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*int* function OTS\_InfoRespond::getPort() [*line 69*]

**Returns server port.**

Returns server port.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getServer() [*line 101*]

**Returns server attribute.**

Returns server attribute.

I have no idea what the hell is it representing :P.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getServerVersion() [*line 111*]

**Returns server version.**

Returns server version.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getTSPQVersion() [*line 29*]

**Returns version of root element.**

Returns version of root element.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*int* function OTS\_InfoRespond::getUptime() [*line 39*]

**Returns server uptime.**

Returns server uptime.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

*string* function OTS\_InfoRespond::getURL() [*line 89*]

**Returns server website.**

Returns server website.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

## Class OTS\_Item

[line 20]

### Single item representation.

Single item representation.

- **Package** POT
- **Version** 0.0.3
- **Since** 0.0.3

Constructor *void* function OTS\_Item::\_\_construct(\$id) [line 48]

#### **Function Parameters:**

- *int* **\$id** Item ID.

### Creates item of given ID.

Creates item of given ID.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*int* function OTS\_Item::count() [line 108]

### Count value for current item.

Count value for current item.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*string* function OTS\_Item::getAttributes() [*line 88*]

**Returns item custom attributes.**

Returns item custom attributes.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*int* function OTS\_Item::getCount() [*line 68*]

**Returns count of item.**

Returns count of item.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*int* function OTS\_Item::getId() [*line 58*]

**Returns item type.**

Returns item type.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*void function OTS\_Item::setAttributes(\$attributes) [line 98]*

**Function Parameters:**

- *string* **\$attributes** Item Attributes.

**Sets item attributes.**

Sets item attributes.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

*void function OTS\_Item::setCount(\$count) [line 78]*

**Function Parameters:**

- *int* **\$count** Count.

**Sets count of item.**

Sets count of item.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

# Class OTS\_Player

[line 22]

## OTServ character abstraction.

OTServ character abstraction.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1

*void function OTS\_Player::ban([\$time = 0]) [line 1648]*

### **Function Parameters:**

- *int* **\$time** Time for time until expires (0 - forever).

## Bans current player.

Bans current player.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public



*void function OTS\_Player::delete() [line 1702]*

**Deletes player.**

Deletes player.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::find(\$name) [line 84]*

**Function Parameters:**

- *string* **\$name** Player's name.

**Loads player by it's name.**

Loads player by it's name.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public

*OTS\_Account* function OTS\_Player::getAccount() [*line 186*]

**Returns account of this player.**

Returns account of this player.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getCap() [*line 841*]

**Capacity.**

Capacity.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*mixed* function OTS\_Player::getConditions() [*line 955*]

**Conditions.**

Conditions.

- **Version** 0.0.3

- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*string* function OTS\_Player::getCustomField(\$field) [*line 1254*]

**Function Parameters:**

- *string* **\$field** Field name.

**Reads custom field.**

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*OTS\_Item|null* function OTS\_Player::getDepot(\$depot) [*line 1529*]

**Function Parameters:**

- *int* **\$depot** Depot ID to get items.

### Returns items tree from given depot.

Returns items tree from given depot.

Note: OTS\_Player class has no information about item types. It returns all items as OTS\_Item, unless they have any contained items in database, so empty container will be instanced as OTS\_Item object, not OTS\_Container.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getDirection() [*line 571*]

### Looking direction.

Looking direction.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getExperience() [*line 328*]

### Experience points.

Experience points.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*OTS\_Group* function OTS\_Player::getGroup() [*line 215*]

**Returns group of this player.**

Returns group of this player.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*string* function OTS\_Player::getGuildNick() [*line 1042*]

**Guild nick.**

Guild nick.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getHealth() [*line 409*]

**Current HP.**

Current HP.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getHealthMax() [*line 436*]

**Maximum HP.**

Maximum HP.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getId() [*line 142*]

**Player ID.**

Player ID.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLastIP() [*line 895*]

**Last login IP.**

Last login IP.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLastLogin() [*line 868*]

**Last login timestamp.**

Last login timestamp.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLevel() [*line 355*]

**Experience level.**

Experience level.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLookAddons() [*line 733*]

**Addons.**

Addons.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLookBody() [*line 598*]

**Body color.**

Body color.

- **Version** 0.0.3
- **Version** 0.0.1



- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLookFeet() [*line 625*]

#### **Boots color.**

Boots color.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLookHead() [*line 652*]

#### **Hair color.**

Hair color.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLookLegs() [*line 679*]

#### **Legs color.**

Legs color.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLookType() [*line 706*]

**Outfit.**

Outfit.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLossExperience() [*line 1165*]

**Percentage of experience lost after dead.**

Percentage of experience lost after dead.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.

- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLossMana() [*line 1192*]

**Percentage of used mana lost after dead.**

Percentage of used mana lost after dead.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getLossSkills() [*line 1219*]

**Percentage of skills lost after dead.**

Percentage of skills lost after dead.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getMagLevel() [*line 382*]

**Magic level.**

Magic level.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getMana() [*line 463*]

#### **Current mana.**

Current mana.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getManaMax() [*line 490*]

#### **Maximum mana.**

Maximum mana.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1

- **Access** public

*int* function OTS\_Player::getManaSpent() [*line 517*]

**Mana spent.**

Mana spent.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*string* function OTS\_Player::getName() [*line 159*]

**Player name.**

Player name.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getPosX() [*line 760*]

**X map coordinate.**

X map coordinate.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getPosY() [*line 787*]

**Y map coordinate.**

Y map coordinate.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getPosZ() [*line 814*]

**Z map coordinate.**

Z map coordinate.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getPremiumEnd() [*line 245*]

**Player's Premium Account expiration timestamp.**

Player's Premium Account expiration timestamp.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*OTS\_GuildRank|null* function OTS\_Player::getRank() [*line 1086*]

**Assigned guild rank.**

Assigned guild rank.

- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getRankId() [*line 1070*]

**Guild rank ID.**

Guild rank ID.

- **Version** 0.0.3
- **Version** 0.0.1
- **Deprecated** 0.0.4 Use getRank().
- **Since** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Access** public

*int* function OTS\_Player::getRedSkullTime() [*line 982*]

### **Red skulled time remained.**

Red skulled time remained.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getSex() [*line 274*]

### **Player gender.**

Player gender.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public



*int* function OTS\_Player::getSkill(\$skill) [*line 1309*]

**Function Parameters:**

- *int* **\$skill** Skill ID.

**Returns player's skill.**

Returns player's skill.

- **Version** 0.0.2
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.2
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getSkillTries(\$skill) [*line 1341*]

**Function Parameters:**

- *int* **\$skill** Skill ID.

**Returns player's skill's tries for next level.**

Returns player's skill's tries for next level.

- **Version** 0.0.2
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.2

- **Since** 0.0.1
- **Access** public

*OTS\_Item|null* function OTS\_Player::getSlot(\$slot) [*line 1394*]

**Function Parameters:**

- *int* **\$slot** Slot to get items.

**Returns items tree from given slot.**

Returns items tree from given slot.

Note: OTS\_Player class has no information about item types. It returns all items as OTS\_Item, unless they have any contained items in database, so empty container will be instanced as OTS\_Item object, not OTS\_Container.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getSoul() [*line 544*]

**Soul points.**

Soul points.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.

- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getTownId() [*line 1138*]

#### **Residence town's ID.**

Residence town's ID.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*int* function OTS\_Player::getVocation() [*line 301*]

#### **Player proffesion.**

Player proffesion.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*bool* function OTS\_Player::hasRedSkull() [*line 1009*]

#### **Checks if player has red skull.**

Checks if player has red skull.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*bool* function OTS\_Player::isBanned() [*line 1683*]

**Checks if player is banned.**

Checks if player is banned.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*bool* function OTS\_Player::isLoaded() [*line 101*]

**Checks if object is loaded.**

Checks if object is loaded.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*bool* function OTS\_Player::isSaveSet() [*line 922*]

**Checks if save flag is set.**

Checks if save flag is set.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Player::load(\$id) [*line 62*]

**Function Parameters:**

- *int* **\$id** Player's ID.

**Loads player with given id.**

Loads player with given id.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Player::save() [*line 111*]

**Saves player in database.**

Saves player in database.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setAccount(\$account) [line 203]*

**Function Parameters:**

- [OTS Account](#) **\$account** Owning account.

**Assigns character to account.**

Assigns character to account.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setCap(\$cap) [line 856]*

**Function Parameters:**

- *int* **\$cap** Capacity.

**Sets capacity.**

Sets capacity.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setConditions(\$conditions) [line 970]*

**Function Parameters:**

- *mixed* **\$conditions** Condition binary field.

**Sets conditions.**

Sets conditions.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setCustomField(\$field, \$value) [line 1284]*

**Function Parameters:**

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

**Writes custom field.**

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field value. It is safe - it makes you sure that no improper queries that could

lead to SQL injection will be executed, but it can make your code working wrong way. For example: `$object->setCustomField('foo', '1');` will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setDepot(\$depot, [\$item = null], [\$pid = 0], [\$depot\_id = 0]) [line 1584]*

**Function Parameters:**

- *int* **\$depot** Depot ID to save items.
- [\*OTS Item\*](#) **\$item** Item (can be a container with content) for given depot. Leave this parameter blank to clear depot.
- *int* **\$pid** Deprecated, not used anymore.
- *int* **\$depot\_id** Internal, for further use.

**Sets depot content.**

Sets depot content.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.



- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*void function* OTS\_Player::setDirection(\$direction) [*line 586*]

**Function Parameters:**

- *int* **\$direction** Looking direction.

**Sets looking direction.**

Sets looking direction.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function* OTS\_Player::setExperience(\$experience) [*line 343*]

**Function Parameters:**

- *int* **\$experience** Experience points.

**Sets experience points.**

Sets experience points.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS\_Player::setGroup(\$group) [line 232]

**Function Parameters:**

- [OTS\\_Group](#) **\$group** Group to be a member.

**Assigns character to group.**

Assigns character to group.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS\_Player::setGuildNick(\$guildnick) [line 1057]

**Function Parameters:**

- *string* **\$guildnick** Name.

**Sets guild nick.**

Sets guild nick.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS\_Player::setHealth(\$health) [line 424]

**Function Parameters:**

- *int* **\$health** Current HP.

### **Sets current HP.**

Sets current HP.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Player::setHealthMax(\$healthmax) [*line 451*]

#### **Function Parameters:**

- *int* **\$healthmax** Maximum HP.

### **Sets maximum HP.**

Sets maximum HP.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Player::setLastIP(\$lastip) [*line 910*]

#### **Function Parameters:**

- *int* **\$lastip** Last login IP.

### **Sets last login IP.**

Sets last login IP.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLastLogin(\$lastlogin) [line 883]*

#### ***Function Parameters:***

- *int* **\$lastlogin** Last login timestamp.

### **Sets last login timestamp.**

Sets last login timestamp.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLevel(\$level) [line 370]*

#### ***Function Parameters:***

- *int* **\$level** Experience level.

### **Sets experience level.**

Sets experience level.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLookAddons(\$lookaddons) [line 748]*

***Function Parameters:***

- *int* **\$lookaddons** Addons.

**Sets addons.**

Sets addons.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLookBody(\$lookbody) [line 613]*

***Function Parameters:***

- *int* **\$lookbody** Body color.

**Sets body color.**

Sets body color.

- **Version** 0.0.1
- **Since** 0.0.1

- **Access** public

*void function OTS\_Player::setLookFeet(\$lookfeet) [line 640]*

***Function Parameters:***

- *int* **\$lookfeet** Boots color.

**Sets boots color.**

Sets boots color.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLookHead(\$lookhead) [line 667]*

***Function Parameters:***

- *int* **\$lookhead** Hair color.

**Sets hair color.**

Sets hair color.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLookLegs(\$looklegs) [line 694]*

**Function Parameters:**

- *int* **\$looklegs** Legs color.

**Sets legs color.**

Sets legs color.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLookType(\$looktype) [line 721]*

**Function Parameters:**

- *int* **\$looktype** Outfit.

**Sets outfit.**

Sets outfit.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLossExperience(\$loss\_experience) [line 1180]*

**Function Parameters:**

- *int* **\$loss\_experience** Percentage of experience lost after dead.

### **Sets percentage of experience lost after dead.**

Sets percentage of experience lost after dead.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLossMana(\$loss\_mana) [line 1207]*

#### ***Function Parameters:***

- *int* **\$loss\_mana** Percentage of used mana lost after dead.

### **Sets percentage of used mana lost after dead.**

Sets percentage of used mana lost after dead.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setLossSkills(\$loss\_skills) [line 1234]*

#### ***Function Parameters:***

- *int* **\$loss\_skills** Percentage of skills lost after dead.

### **Sets percentage of skills lost after dead.**

Sets percentage of skills lost after dead.



- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setMagLevel(\$maglevel) [line 397]*

***Function Parameters:***

- *int* **\$maglevel** Magic level.

**Sets magic level.**

Sets magic level.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setMana(\$mana) [line 478]*

***Function Parameters:***

- *int* **\$mana** Current mana.

**Sets current mana.**

Sets current mana.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setManaMax(\$manamax) [line 505]*

**Function Parameters:**

- *int* **\$manamax** Maximum mana.

**Sets maximum mana.**

Sets maximum mana.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setManaSpent(\$manaspent) [line 532]*

**Function Parameters:**

- *int* **\$manaspent** Mana spent.

**Sets mana spent.**

Sets mana spent.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setName(\$name) [line 174]*

**Function Parameters:**

- *string* **\$name** Name.

**Sets players's name.**

Sets players's name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setPosX(\$posx) [line 775]*

**Function Parameters:**

- *int* **\$posx** X map coordinate.

**Sets X map coordinate.**

Sets X map coordinate.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setPosY(\$posy) [line 802]*

**Function Parameters:**

- *int* **\$posy** Y map coordinate.

### **Sets Y map coordinate.**

Sets Y map coordinate.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Player::setPosZ(\$posz) [*line 829*]

#### **Function Parameters:**

- *int* **\$posz** Z map coordinate.

### **Sets Z map coordinate.**

Sets Z map coordinate.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void* function OTS\_Player::setPremiumEnd(\$premend) [*line 262*]

#### **Function Parameters:**

- *int* **\$premend** PACC expiration timestamp.

### Sets player's Premium Account expiration timestamp.

Sets player's Premium Account expiration timestamp.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3
- **Access** public

*void function OTS\_Player::setRank([\$guildRank = null]) [line 1119]*

#### **Function Parameters:**

- [OTS\\_GuildRank](#)|*null* **\$guildRank** Guild rank (null to clear assign).

### Assigns guild rank.

Assigns guild rank.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setRankId(\$rank\_id) [line 1109]*

#### **Function Parameters:**

- *int* **\$rank\_id** Guild rank ID.

### **Sets guild rank ID.**

Sets guild rank ID.

- **Version** 0.0.1
- **Deprecated** 0.0.4 Use setRank().
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setRedSkull() [line 1030]*

### **Sets red skull flag.**

Sets red skull flag.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setRedSkullTime(\$redskulltime) [line 997]*

#### **Function Parameters:**

- *int* **\$redskulltime** Red skulled time remained.

### **Sets red skulled time remained.**

Sets red skulled time remained.

- **Version** 0.0.1

- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setSave() [line 943]*

### **Sets save flag.**

Sets save flag.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setSex(\$sex) [line 289]*

### **Function Parameters:**

- *int* **\$sex** Player gender.

### **Sets player gender.**

Sets player gender.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setSkill(\$skill, \$value) [line 1327]*

### **Function Parameters:**

- *int* **\$skill** Skill ID.

- *int* **\$value** Skill value.

### **Sets skill value.**

Sets skill value.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public

*void* function OTS\_Player::setSkillTries(\$skill, \$tries) [*line 1359*]

#### **Function Parameters:**

- *int* **\$skill** Skill ID.
- *int* **\$tries** Skill tries.

### **Sets skill's tries for next level.**

Sets skill's tries for next level.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public



*void function OTS\_Player::setSlot(\$slot, [\$item = null], [\$pid = 0]) [line 1448]*

**Function Parameters:**

- *int* **\$slot** Slot to save items.
- [\*OTS\\_Item\*](#) **\$item** Item (can be a container with content) for given slot. Leave this parameter blank to clear slot.
- *int* **\$pid** Deprecated, not used anymore.

**Sets slot content.**

Sets slot content.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E\_OTS\_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setSoul(\$soul) [line 559]*

**Function Parameters:**

- *int* **\$soul** Soul points.

**Sets soul points.**

Sets soul points.

- **Version** 0.0.1
- **Since** 0.0.1

- **Access** public

*void function OTS\_Player::setTownId(\$town\_id) [line 1153]*

***Function Parameters:***

- *int* **\$town\_id** Residence town's ID.

**Sets residence town's ID.**

Sets residence town's ID.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::setVocation(\$vocation) [line 316]*

***Function Parameters:***

- *int* **\$vocation** Player proffesion.

**Sets player proffesion.**

Sets player proffesion.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::unban() [line 1665]*

**Deletes ban from current player.**

Deletes ban from current player.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*void function OTS\_Player::unsetRedSkull() [line 1022]*

**Unsets red skull flag.**

Unsets red skull flag.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void function OTS\_Player::unsetSave() [line 935]*

**Unsets save flag.**

Unsets save flag.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

array function OTS\_Player::\_\_sleep() *[line 51]*

**Magic PHP5 method.**

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4
- **Access** public

## Class OTS\_Players\_List

*[line 21]*

**List of players.**

List of players.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.1

void function OTS\_Players\_List::deletePlayer(\$player) *[line 30]*

**Function Parameters:**

- [\*OTS\\_Player\*](#) **\$player** Player to be deleted.

### **Deletes player.**

Deletes player.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS\_Player->delete().
- **Since** 0.0.1
- **Access** public

*void function OTS\_Players\_List::init() [line 43]*

### **Sets list parameters.**

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

## **Class OTS\_SQLField**

*[line 20]*

### **SQL identifier representation.**

SQL identifier representation.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.5

Constructor *void* function OTS\_SQLField::\_\_construct(\$name, [\$table = "]) [*line 41*]

**Function Parameters:**

- *string* **\$name** Field name.
- *string* **\$table** Table name.

**Creates new field representation.**

Creates new field representation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*string* function OTS\_SQLField::getName() [*line 52*]

**Returns field name.**

Returns field name.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*string* function OTS\_SQLField::getTable() [*line 62*]

**Returns table name.**

Returns table name.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

## Class OTS\_SQLFilter

[*line 20*]

**SQL WHERE clause object.**

SQL WHERE clause object.

- **Package** POT
- **Version** 0.0.5
- **Since** 0.0.5

### OTS\_SQLFilter::CRITERIUM\_AND

= 1 [*line 58*]

**AND sibling.**

AND sibling.

- **Version** 0.0.5
- **Since** 0.0.5

#### OTS\_SQLFilter::CRITERIUM\_OR

= 2 [*line 62*]

**OR sibling.**  
OR sibling.

- **Version** 0.0.5
- **Since** 0.0.5

#### OTS\_SQLFilter::OPERATOR\_EQUAL

= 1 [*line 25*]

**Equal operator.**  
Equal operator.

- **Version** 0.0.5
- **Since** 0.0.5

#### OTS\_SQLFilter::OPERATOR\_GREATER

= 3 [*line 33*]

**Greater-then operator.**  
Greater-then operator.



- **Version** 0.0.5
- **Since** 0.0.5

#### OTS\_SQLFilter::OPERATOR\_LIKE

= 7 *[line 49]*

##### **LIKE operator.**

LIKE operator.

- **Version** 0.0.5
- **Since** 0.0.5

#### OTS\_SQLFilter::OPERATOR\_LOWER

= 2 *[line 29]*

##### **Lower-then operator.**

Lower-then operator.

- **Version** 0.0.5
- **Since** 0.0.5

#### OTS\_SQLFilter::OPERATOR\_NEQUAL

= 4 *[line 37]*

##### **Not-equal operator.**

Not-equal operator.

- **Version** 0.0.5
- **Since** 0.0.5

**OTS\_SQLFilter::OPERATOR\_NGREATER**

= 6 [*line 45*]

**Not-greater-then operator.**

Not-greater-then operator.

- **Version** 0.0.5
- **Since** 0.0.5

**OTS\_SQLFilter::OPERATOR\_NLIKE**

= 8 [*line 53*]

**Not-LIKE operator.**

Not-LIKE operator.

- **Version** 0.0.5
- **Since** 0.0.5

**OTS\_SQLFilter::OPERATOR\_NLOWER**

= 5 [*line 41*]

**Not-lower-then operator.**

Not-lower-then operator.

- **Version** 0.0.5
- **Since** 0.0.5

*void* function OTS\_SQLFilter::addFilter(\$left, [\$right = null], [\$operator = self::OPERATOR\_EQUAL], [\$criterium = self::CRITERIUM\_AND]) [*line 238*]

**Function Parameters:**

- *mixed* **\$left** Left side ([OTS\\_SQLField class](#) object, or literal value).
- *mixed* **\$right** Right side ([OTS\\_SQLField class](#) object, or literal value).
- *int* **\$operator** Operator used for comparsion (equal check by default).
- *int* **\$criterium** Criterium merging method (AND by default).

**General-purpose filter.**

General-purpose filter.

Appends new filter in universal way.

To append subset of another filters us addFilter(\$OTS\_SQLFilterObject).

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*void* function OTS\_SQLFilter::compareField(\$field, \$value, [\$operator = self::OPERATOR\_EQUAL], [\$criterium = self::CRITERIUM\_AND]) [*line 251*]

**Function Parameters:**

- *string* **\$field** Field name.

- *mixed* **\$value** Literal value.
- *int* **\$operator** Operator used for comparsion (equal by default).
- *int* **\$criterium** Criterium merging method (AND by default).

### **Compares field with a literal value.**

Compares field with a literal value.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*array* function OTS\_SQLFilter::getTables() [*line 263*]

### **Returns list of all tables used by filter.**

Returns list of all tables used by filter.

This is required for FROM clause.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

*array* function OTS\_SQLFilter::\_\_sleep() [*line 79*]

### **Magic PHP5 method.**

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.5

- **Since** 0.0.5
- **Access** public

*string* function OTS\_SQLFilter::\_\_toString() [*line 92*]

**Returns string representation of WHERE clause.**

Returns string representation of WHERE clause.

Returned string can be easily inserted into SQL query.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

## Class POT

[*line 30*]

**Main POT class.**

Main POT class.

- **Package** POT
- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1

**POT::BAN\_ACCOUNT**

= 3 *[line 261]*

## **Account ban.**

Account ban.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

## **POT::BAN\_IP**

= 1 *[line 247]*

## **IP ban.**

IP ban.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

## **POT::BAN\_PLAYER**

= 2 *[line 254]*

## **Player ban.**

Player ban.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

**POT::DB\_MYSQL**

= 1 *[line 35]*

**MySQL driver.**  
MySQL driver.

- **Version** 0.0.1
- **Since** 0.0.1

**POT::DB\_ODBC**

= 4 *[line 53]*

**ODBC driver.**  
ODBC driver.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4

**POT::DB\_PGSQL**

= 3 *[line 46]*

## PostgreSQL driver.

PostgreSQL driver.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4

## POT::DB\_SQLITE

= 2 *[line 39]*

## SQLite driver.

SQLite driver.

- **Version** 0.0.1
- **Since** 0.0.1

## POT::DEPOT\_SID\_FIRST

= 100 *[line 239]*

## First depot item sid.

First depot item sid.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1



- **Since** 0.0.4

**POT::DIRECTION\_EAST**

= 1 [*line 102*]

**East.**

East.

- **Version** 0.0.1
- **Since** 0.0.1

**POT::DIRECTION\_NORTH**

= 0 [*line 98*]

**North.**

North.

- **Version** 0.0.1
- **Since** 0.0.1

**POT::DIRECTION\_SOUTH**

= 2 [*line 106*]

**South.**

South.

- **Version** 0.0.1

- **Since 0.0.1**

**POT::DIRECTION\_WEST**

= 3 [*line 110*]

**West.**

West.

- **Version 0.0.1**
- **Since 0.0.1**

**POT::ORDER\_ASC**

= 1 [*line 269*]

**Ascencind sorting order.**

Ascencind sorting order.

- **Version 0.0.5**
- **Version 0.0.1**
- **Since 0.0.1**
- **Since 0.0.5**

**POT::ORDER\_DESC**

= 2 [*line 276*]

**Descending sorting order.**

Descending sorting order.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

#### **POT::SEX\_FEMALE**

= 0 [*line 58*]

#### **Female gender.**

Female gender.

- **Version** 0.0.1
- **Since** 0.0.1

#### **POT::SEX\_MALE**

= 1 [*line 62*]

#### **Male gender.**

Male gender.

- **Version** 0.0.1
- **Since** 0.0.1

#### **POT::SKILL\_AXE**

= 3 [*line 139*]

## **Axe fighting.**

Axe fighting.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

### **POT::SKILL\_CLUB**

= 1 [*line 125*]

## **Club fighting.**

Club fighting.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

### **POT::SKILL\_DISTANCE**

= 4 [*line 146*]

## **Distance fighting.**

Distance fighting.

- **Version** 0.0.2

- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

## POT::SKILL\_FISHING

= 6 [*line 160*]

### **Fishing.**

Fishing.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

## POT::SKILL\_FIST

= 0 [*line 118*]

### **Fist fighting.**

Fist fighting.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

## POT::SKILL\_SHIELDING

= 5 [*line 153*]

### **Shielding.**

Shielding.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

### **POT::SKILL\_SWORD**

= 2 [*line 132*]

### **Sword fighting.**

Sword fighting.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

### **POT::SLOT\_AMMO**

= 10 [*line 231*]

### **Ammunition slot.**

Ammunition slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_ARMOR

= 4 [*line 189*]

### **Armor slot.**

Armor slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_BACKPACK

= 3 [*line 182*]

### **Backpack slot.**

Backpack slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_FEET

= 8 [*line 217*]

### Boots slot.

Boots slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_HEAD

= 1 [*line 168*]

### Head slot.

Head slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_LEFT

= 6 [*line 203*]

### Left hand slot.

Left hand slot.



- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_LEGS

= 7 [*line 210*]

### **Legs slot.**

Legs slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_NECKLACE

= 2 [*line 175*]

### **Necklace slot.**

Necklace slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_RIGHT

= 5 [*line 196*]

### Right hand slot.

Right hand slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::SLOT\_RING

= 9 [*line 224*]

### Ring slot.

Ring slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

## POT::VOCATION\_DRUID

= 2 [*line 81*]

### Druid.

Druid.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

#### POT::VOCATION\_KNIGHT

= 4 [*line 93*]

**Knight.**

Knight.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

#### POT::VOCATION\_NONE

= 0 [*line 69*]

**None vocation.**

None vocation.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

#### POT::VOCATION\_PALADIN

= 3 [line 87]

## Paladin.

Paladin.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

## POT::VOCATION\_SORCERER

= 1 [line 75]

## Sorcerer.

Sorcerer.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

*void* function POT::banIP(\$ip, [\$mask = '255.255.255.255'], [\$time = 0]) [line 617]

### **Function Parameters:**

- *string* **\$ip** IP to ban.
- *string* **\$mask** Mask for ban (by default bans only given IP).
- *int* **\$time** Time for time until expires (0 - forever).

## Bans given IP number.

Bans given IP number.

Adds IP/mask ban. You can call this function with only one parameter to ban only given IP

address without expiration.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*void* function POT::connect(\$driver, \$params) [*line 399*]

## connect.php

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // includes POT main file
12 include('../classes/OTS.php');
13
14 // you can easily store such structure in config.php
15 $config = array(
16     'driver' =>     POT::DB_MYSQL,
17     'prefix' =>     '',
18     'host' =>       'localhost',
19     'user' =>       'wrzasq',
20     'password' =>   '',
21     'database' =>   'otserv'
22 );
23
24 // connects to database
25 $ots = POT::getInstance();
26 $ots->connect(null, $config);
27 // could be: $ots->connect(POT::DB_MYSQL, $config);
28
29 ?>
```

### Function Parameters:

- *int*/*null* **\$driver** Database driver type.
- *array* **\$params** Connection info.

### Connects to database.

Connects to database.

Creates OTServ database connection object.

First parameter is one of database driver constants values. Currently MySQL, SQLite, PostgreSQL and ODBC drivers are supported.

This parameter can be null, then you have to specify '*driver*' parameter.

Such way is comfortable to store entire database configuration in one array and possibly runtime evaluation and/or configuration file saving.

For parameters list see driver documentation. Common parameters for all drivers are:

- *driver* - optional, specifies driver, applies when *\$driver* method parameter is *null*
- *prefix* - optional, prefix for database tables, use if you have more then one OTServ installed on one database.

- **Version** 0.0.4
- **Version** 0.0.1
- **Throws** Exception When driver is not supported.
- **Since** 0.0.1
- **Access** public
- **Example**

OTS\_SQLFilter function POT::createFilter() [line 709]

**Creates lists filter.**

Creates lists filter.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*IOTS\_DAO function POT::createObject(\$class) [line 450]*

**Function Parameters:**

- *string* **\$class** Class name.

**Creates OTServ DAO class instance.**

Creates OTServ DAO class instance.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*PDO function POT::getDBHandle() [line 525]*

**Returns database connection handle.**

Returns database connection handle.

At all you shouldn't use this method and work with database using POT classes, but it may be sometime necessary to use direct database access (mainly until POT won't provide many important features).

It is also important as serialised objects after unserialisation needs to be re-initialised with database connection.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4
- **Access** public

POT function POT::getInstance() [*line 283*]

### **Singleton.**

Singleton.

- **Version** 0.0.1
- **Static**
- **Since** 0.0.1
- **Access** public

*int|bool* function POT::getVocationID(\$name) [*line 569*]

#### **Function Parameters:**

- *string* **\$name** Vocation.

### **Returns vocation's ID.**

Returns vocation's ID.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*string|bool* function POT::getVocationName(\$id) [*line 582*]

#### **Function Parameters:**

- *int* **\$id** Vocation ID.



### **Returns name of given vocation's ID.**

Returns name of given vocation's ID.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*array* function POT::getVocationsList() [*line 601*]

### **Returns list (id => name) of loaded vocations.**

Returns list (id => name) of loaded vocations.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*bool* function POT::isIPBanned(\$ip) [*line 685*]

#### **Function Parameters:**

- *string* **\$ip** IP to ban.

### **Checks if given IP is banned.**

Checks if given IP is banned.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*void* function POT::loadClass(\$class) [*line 358*]

**Function Parameters:**

- *string* **\$class** Class name.

**Loads POT class file.**

Loads POT class file.

Runtime class loading on demand - usefull for \_\_autoload() function.

Note: Since 0.0.2 version this function is suitable for spl\_autoload\_register().

Note: Since 0.0.3 version this function handles also exceptions.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

*void* function POT::loadVocations(\$file) [*line 548*]

**Function Parameters:**

- *string* **\$file** vocations.xml file location.

## Loads vocations list.

Loads vocations list.

Loads vocations list from given file.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

*OTS\_InfoRespond|bool* function POT::serverStatus(\$server, \$port) [line 468]

### example

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // server and port
15 $server = '127.0.0.1';
16 $port = 7171;
17
18 // queries server of status info
19 $status = $ots-> serverStatus($server, $port);
20
21 // offline
22 if(!$status)
23 {
24     echo 'Server ', $server, ' is offline.', "\n" ;
25 }
26 // displays various info
27 else
28 {
29     echo 'Server name: ', $status-> getName(), "\n" ;
30     echo 'Server owner: ', $status-> getOwner(), "\n" ;
31     echo 'Players online: ', $status-> getOnlinePlayers(), "\n" ;
32     echo 'Maximum allowed number of players: ', $status-> getMaxPlayers(), "\n" ;
33     echo 'Required client version: ', $status-> getClientVersion(), "\n" ;
34     echo 'All monsters: ', $status-> getMonstersCount(), "\n" ;
35     echo 'Server message: ', $status-> getMOTD(), "\n" ;
36 }
37
38 ?>
```

### Function Parameters:

- *string* **\$server** Server IP/domain.
- *int* **\$port** OTServ port.

### Queries server status.

Queries server status.

Sends 'info' packet to OTS server and return output.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public
- **Example**

*void* function POT::setPOTPath(\$path) [line 314]

fakeroot.php

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // this is the way you should work with POT if you moved main OTS.php file outside POT's directory
12 include('path/to/OTS.php');
13
14 // dont use 'new POT()'!!!
15 $ots = POT::getInstance();
16 $ots-> setPOTPath('../classes/');
17
18 /*
19  here comes your stuff...
20 */
21
22 ?>
```

### Function Parameters:

- *string* **\$path** POT files path.

### **Set POT directory.**

Set POT directory.

Use this method if you keep your POT package in different directory then this file.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public
- **Example**

*void* function POT::unbanIP(\$ip, [\$mask = '255.255.255.255']) [*line 652*]

#### ***Function Parameters:***

- *string* **\$ip** IP to ban.
- *string* **\$mask** Mask for ban (by default 255.255.255.255).

### **Deletes ban from given IP number.**

Deletes ban from given IP number.

Removes given IP/mask ban.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public



## compat.php

### **POT compatibility assurance package.**

POT compatibility assurance package.

This package makes you sure that POT scripts won't cause FATAL errors on PHP older PHP 5.x versions. However remember that some PHP features won't be enabled with it. For example if you have PHP 5.0.x, this package will define Countable interface for you so PHP will know it, but it won't allow you to use count(\$countableObject) structure.

- **Package** POT
- **Sub-Package** compat
- **Author** Wrzasq < [wrzasq@gmail.com](mailto:wrzasq@gmail.com)>
- **Version** 0.0.2
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.2
- **License** [GNU Lesser General Public License, Version 3](#)

# Appendices



# Appendix A - Class Trees

## Package POT

### E\_OTS\_NoDriver

- Exception
  - [E\\_OTS\\_NoDriver](#)

### E\_OTS\_NotLoaded

- Exception
  - [E\\_OTS\\_NotLoaded](#)

### IOTS\_DAO

- [IOTS\\_DAO](#)

### IOTS\_DB

- [IOTS\\_DB](#)

### IOTS\_GuildAction

- [IOTS\\_GuildAction](#)

### OTS\_Base\_DAO

- [OTS\\_Base\\_DAO](#)

- [OTS Account](#)
- [OTS Group](#)
- [OTS Guild](#)
- [OTS GuildRank](#)
- [OTS Player](#)
- [OTS SQLFilter](#)

## OTS\_Base\_List

- [OTS\\_Base\\_List](#)
  - [OTS\\_Accounts\\_List](#)
  - [OTS\\_Groups\\_List](#)
  - [OTS\\_GuildRanks\\_List](#)
  - [OTS\\_Guilds\\_List](#)
  - [OTS\\_Players\\_List](#)

## OTS\_DB\_MySQL

- PDO
  - [OTS\\_DB\\_MySQL](#)

## OTS\_DB\_ODBC

- PDO
  - [OTS\\_DB\\_ODBC](#)

## OTS\_DB\_PostgreSQL

- PDO
  - [OTS\\_DB\\_PostgreSQL](#)

## OTS\_DB\_SQLite

- PDO
  - [OTS\\_DB\\_SQLite](#)

## OTS\_InfoRespond

- DOMDocument
  - [OTS\\_InfoRespond](#)

## OTS\_Item

- [OTS\\_Item](#)
  - [OTS\\_Container](#)

## OTS\_SQLField

- [OTS\\_SQLField](#)

## POT

- [POT](#)

# Appendix B - README/CHANGELOG/INSTALL

# CHANGELOG

## [0.0.5]

- \* Added support for vocations.xml file. <wrzasq>
- \* Added support for bans. <wrzasq>
- \* Added sorting and filtering for lists. <wrzasq>
- \* Code grouped into base classes. <wrzasq>
- \* Some code optimisation. <wrzasq>
- \* Fixed typos. <wrzasq>

## [0.0.4]

- \* Added guild system support (guilds, ranks, invitations and requests drivers mechanisms). <wrzasq>
- \* Added account group support. <wrzasq>
- \* Added support for depot\_id field (it is reserved in OTServ for further use). <wrzasq>
- \* Added PostgreSQL and ODBC drivers. <wrzasq>
- \* Added \_\_sleep() and \_\_wakeup() methods to allow POT objects to be stored in sessions. <wrzasq>
- \* Added \_\_clone() methods to allow save ID-losing cloning of POT objects. <wrzasq>
- \* Added \_\_set\_state() methods. <wrzasq>
- \* Updated players table structure. <wrzasq>
- \* Dropped REGEXP operator bindings - not used anywhere. <wrzasq>
- \* Fixed items loading and saving. <wrzasq>
- \* Fixed typos. <wrzasq>

## [0.0.3]

- \* Added custom fields support. <wrzasq>
- \* Added items and depots support. <wrzasq>
- \* Added support for players PACC timestamps. <wrzasq>
- \* Fixed loading skills. <wrzasq>
- \* Replaced E\_USER\_\* with exceptions. <wrzasq>
- \* Uses fetchAll() in loops to prevent MySQL buffering problems. <wrzasq>
- \* Restricted access to POT class constructor to make sure it won't be instantiated directly. <wrzasq>

## [0.0.2]

- \* Added "compat" library for POT. <wrzasq>
- \* Added skills support in OTS\_Player class. <wrzasq>
- \* Added 'info' serverStatus() method and respond handler for server status protocol. <wrzasq>
- \* Fixed `redskulltime` field name in OTS\_Player. <wrzasq>
- \* Fixed 'password' parameter for DB\_MYSQL driver. <wrzasq>
- \* Added find() to OTS\_Account class to load accounts by their's e-mail addresses. <wrzasq>
- \* POT class now automatically binds own \_\_autoload() handler with spl\_autoload\_register(). <wrzasq>

## [0.0.1]

- \* Initial release. <wrzasq>

# README

POT (PHP OTServ Toolkit) is a PHP toolkit for scripts that work with OTServ database.

## ===== About =====

This toolkit provides a way for PHP programmers that don't know SQL language to work with OTServ database.

For installation help check INSTALL file.

For usage tutorial/API documentation check <http://www.otserv-aac.info/pot/> or documentation.pdf file.

## ===== Contact =====

In case of any contact needed, please use following e-mail address: [wrzasq@gmail.com](mailto:wrzasq@gmail.com).

## ===== Files =====

classes/ - POT class files.  
examples/ - example files for learning.  
tutorials/ - phpDocumentor directory.  
CHANGELOG - changes history.  
INSTALL - installation tutorial.  
LICENSE - POT license (GNU LGPL v3), if you don't accept it - don't use any of those scripts.  
NEWS - changes in current release.  
README - this readme file.  
RULES - rules to be followed during developing contributed code.  
Makefile - make input, for documentation generation.  
documentation.pdf - phpDocumentor-generator documentation in PDF format.  
compat.php - Compatibility assurance library.  
test.php - phpUnit test suite.

## ===== Makefile =====

Makefile contains some targets for make that can help in development. Makefile requires following command-line commands:

php: PHP CLI interface.  
phpdoc: phpDocumentor.  
phpunit: PHPUnit testing framework.

Possible targets:

all: default one, runs all other targets (in order: clean, check, documentation, pdf, online, test, package).  
clean: deletes documentation.  
check: checks syntax of all PHP files.  
documentation: generates HTML documentation.  
pdf: generates PDF documentation.  
online: OTServ-AAC website documentation template used.  
test: runs test suite.  
package: creates pot.tar.gz file for distribution purposes.

For more readable output of phpUnit test run:  
php test.php

===== Credits =====

\* Wrzasq <wrzasq@gmail.com> - project initiator, main developer.

For more info see AUTHORS file in OTServ tree.

## INSTALL

POT is a toolkit which means you don't literally install it. You copy it's files and write code for it. All source files are located in classes/ subdirectory. Copy them to your script directory.

You can put main file - OTS.php in different directory then other files.

For information about how to include POT in your code see the documentation.

## NEWS

What's new in 0.0.5 version?

\* Added support for vocations.xml file.

Vocations are now loaded from vocations.xml file. You can get list of all vocations on server, not only default ones.

\* Added support for bans.

Can ban, unban and check if ip, account or player is banned.

\* Added sorting and filtering for lists.

All lists can now be sorted and/or filtered. It will be more described in further documentation.

\* Code grouped into base classes.

There are some generic classes that implements common routines to minimise the code and allows simple automatic subclassing.

\* Some code optimisation.

Changed some code to use more effective ways of solving same problems.

\* Fixed typos.

Like always there were some bugs which of course were fixed.

# Appendix D - Todo List

## In Package POT

In [OTS.php](#)

- 0.0.6: Spawns support (OTBM support -> cache).
- 0.1.0: Get rid of POT::getInstance()->create\*() calls - use POT::getInstance()->getDBHandle() in constructors.
- 0.1.0: Implement \_\_get()/\_\_set()/\_\_call()/\_\_toString(); ArrayAccess interface.
- 0.1.0: Items list (items.xml + items.otb -> cache).
- 1.0.0: Complete phpUnit test.
- 1.0.0: More detailed documentation.



# Index

## A

[Account number hack](#) . . . . . 18

## C

[constructor OTS\\_Item::\\_\\_construct\(\)](#) . . . . . 146  
*Creates item of given ID.*

[constructor OTS\\_DB\\_SQLite::\\_\\_construct\(\)](#) . . . . . 99  
*Creates database connection.*

[constructor OTS\\_SQLField::\\_\\_construct\(\)](#) . . . . . 195  
*Creates new field representation.*

[compat.php](#) . . . . . 228  
*POT compatibility assurance package.*

[CHANGELOG](#) . . . . . 234

[constructor OTS\\_DB\\_PostgreSQL::\\_\\_construct\(\)](#) . . . . . 95  
*Creates database connection.*

[constructor OTS\\_DB\\_ODBC::\\_\\_construct\(\)](#) . . . . . 92  
*Creates database connection.*

[constructor IOTS\\_GuildAction::\\_\\_construct\(\)](#) . . . . . 56  
*Objects are initialized with a guild that they are assigned to.*

[constructor IOTS\\_DB::\\_\\_construct\(\)](#) . . . . . 52  
*Connection parameters.*

[constructor OTS\\_Base\\_DAO::\\_\\_construct\(\)](#) . . . . . 75  
*Sets database connection handler.*

[constructor OTS\\_Base\\_List::\\_\\_construct\(\)](#) . . . . . 78  
*Sets database connection handler.*

[constructor OTS\\_DB\\_MySQL::\\_\\_construct\(\)](#) . . . . . 88  
*Creates database connection.*

[constructor IOTS\\_DAO::\\_\\_construct\(\)](#) . . . . . 51  
*DAO objects must be initialized with a database.*

## D

[DAO objects](#) . . . . . 9

## E

[E\\_OTS\\_NotLoaded](#) . . . . . 50  
*Occurs when code attempts to access property of not loaded object.*

[E\\_OTS\\_NoDriver](#) . . . . . 50  
*Occurs when code attempts to execute driven action that has no assigned driver to handle it.*

[E\\_OTS\\_NotLoaded.php](#) . . . . . 23

<a href="#">E_OTs_NoDriver.php</a>	22
------------------------------------	----

## G

<a href="#">Guild action drivers</a>	15
<a href="#">Guilds</a>	13

## I

<a href="#">IOTS_GuildAction::addRequest()</a>	56
<i>Adds new request.</i>	
<a href="#">IOTS_GuildAction</a>	55
<i>Guild action interface.</i>	
<a href="#">IOTS_DB::tableName()</a>	55
<i>Query-quoted table name.</i>	
<a href="#">IOTS_GuildAction::deleteRequest()</a>	57
<i>Deletes request.</i>	
<a href="#">IOTS_GuildAction::listRequests()</a>	57
<i>List of saved pending actions.</i>	
<a href="#">INSTALL</a>	236
<a href="#">IOTS_GuildAction::submitRequest()</a>	58
<i>Finalizes request.</i>	
<a href="#">IOTS_DB::SQLquote()</a>	54
<i>Query-quoted string value.</i>	
<a href="#">IOTS_DB::SQLquery()</a>	54
<i>Evaluates query.</i>	
<a href="#">IOTS_DAO</a>	51
<i>OTServ database object.</i>	
<a href="#">IOTS_GuildAction.php</a>	26
<a href="#">IOTS_DB.php</a>	25
<a href="#">IOTS_DB</a>	52
<i>OTServ database handler interface.</i>	
<a href="#">IOTS_DB::fieldName()</a>	53
<i>Query-quoted field name.</i>	
<a href="#">IOTS_DB::limit()</a>	53
<i>LIMIT/OFFSET clause for queries.</i>	
<a href="#">IOTS_DB::lastInsertId()</a>	53
<i>ID of last created record.</i>	
<a href="#">IOTS_DAO.php</a>	24

## N

<a href="#">NEWS</a>	236
----------------------	-----

## O

<a href="#">OTS_Player::getDirection()</a>	153
<i>Looking direction.</i>	
<a href="#">OTS_Player::getDepot()</a>	152

<i>Returns items tree from given depot.</i>	
<a href="#">OTS_Player::getCustomField()</a>	152
<i>Reads custom field.</i>	
<a href="#">OTS_Player::getConditions()</a>	151
<i>Conditions.</i>	
<a href="#">OTS_Player::getExperience()</a>	153
<i>Experience points.</i>	
<a href="#">OTS_Player::getGroup()</a>	154
<i>Returns group of this player.</i>	
<a href="#">OTS_Player::getHealthMax()</a>	155
<i>Maximum HP.</i>	
<a href="#">OTS_Player::getHealth()</a>	155
<i>Current HP.</i>	
<a href="#">OTS_Player::getGuildNick()</a>	154
<i>Guild nick.</i>	
<a href="#">OTS_Player::getCap()</a>	151
<i>Capacity.</i>	
<a href="#">OTS_Player::getAccount()</a>	151
<i>Returns account of this player.</i>	
<a href="#">OTS_Item::setAttributes()</a>	148
<i>Sets item attributes.</i>	
<a href="#">OTS_Item::getId()</a>	147
<i>Returns item type.</i>	
<a href="#">OTS_Item::getCount()</a>	147
<i>Returns count of item.</i>	
<a href="#">OTS_Item::getAttributes()</a>	147
<i>Returns item custom attributes.</i>	
<a href="#">OTS_Item::setCount()</a>	148
<i>Sets count of item.</i>	
<a href="#">OTS_Player</a>	149
<i>OTServ character abstraction.</i>	
<a href="#">OTS_Player::find()</a>	150
<i>Loads player by it's name.</i>	
<a href="#">OTS_Player::delete()</a>	150
<i>Deletes player.</i>	
<a href="#">OTS_Player::ban()</a>	149
<i>Bans current player.</i>	
<a href="#">OTS_Player::getId()</a>	155
<i>Player ID.</i>	
<a href="#">OTS_Player::getLastIP()</a>	156
<i>Last login IP.</i>	
<a href="#">OTS_Player::getManaSpent()</a>	162
<i>Mana spent.</i>	
<a href="#">OTS_Player::getManaMax()</a>	161
<i>Maximum mana.</i>	
<a href="#">OTS_Player::getMana()</a>	161
<i>Current mana.</i>	
<a href="#">OTS_Player::getMagLevel()</a>	160
<i>Magic level.</i>	
<a href="#">OTS_Player::getName()</a>	162
<i>Player name.</i>	
<a href="#">OTS_Player::getPosX()</a>	162
<i>X map coordinate.</i>	

<a href="#">OTS_Player::getPremiumEnd()</a>	164
<i>Player's Premium Account expiration timestamp.</i>	
<a href="#">OTS_Player::getPosZ()</a>	163
<i>Z map coordinate.</i>	
<a href="#">OTS_Player::getPosY()</a>	163
<i>Y map coordinate.</i>	
<a href="#">OTS_Player::getLossSkills()</a>	160
<i>Percentage of skills lost after dead.</i>	
<a href="#">OTS_Player::getLossMana()</a>	160
<i>Percentage of used mana lost after dead.</i>	
<a href="#">OTS_Player::getLookBody()</a>	157
<i>Body color.</i>	
<a href="#">OTS_Player::getLookAddons()</a>	157
<i>Addons.</i>	
<a href="#">OTS_Player::getLevel()</a>	157
<i>Experience level.</i>	
<a href="#">OTS_Player::getLastLogin()</a>	156
<i>Last login timestamp.</i>	
<a href="#">OTS_Player::getLookFeet()</a>	158
<i>Boots color.</i>	
<a href="#">OTS_Player::getLookHead()</a>	158
<i>Hair color.</i>	
<a href="#">OTS_Player::getLossExperience()</a>	159
<i>Percentage of experience lost after dead.</i>	
<a href="#">OTS_Player::getLookType()</a>	159
<i>Outfit.</i>	
<a href="#">OTS_Player::getLookLegs()</a>	158
<i>Legs color.</i>	
<a href="#">OTS_Item::count()</a>	146
<i>Count value for current item.</i>	
<a href="#">OTS_Item</a>	146
<i>Single item representation.</i>	
<a href="#">OTS_GuildRanks_List::deleteGuildRank()</a>	136
<i>Deletes guild rank.</i>	
<a href="#">OTS_GuildRanks_List</a>	136
<i>List of guild ranks.</i>	
<a href="#">OTS_GuildRank::setName()</a>	135
<i>Sets rank's name.</i>	
<a href="#">OTS_GuildRank::setLevel()</a>	135
<i>Sets rank's access level within guild.</i>	
<a href="#">OTS_GuildRanks_List::init()</a>	137
<i>Sets list parameters.</i>	
<a href="#">OTS_Guilds_List</a>	137
<i>List of guilds.</i>	
<a href="#">OTS_InfoRespond</a>	138
<i>Wrapper for 'info' respond's DOMDocument.</i>	
<a href="#">OTS_Guilds_List::init()</a>	138
<i>Sets list parameters.</i>	
<a href="#">OTS_Guilds_List::deleteGuild()</a>	137
<i>Deletes guild.</i>	
<a href="#">OTS_GuildRank::setGuild()</a>	134
<i>Assigns rank to guild.</i>	
<a href="#">OTS_GuildRank::setCustomField()</a>	133

<i>Writes custom field.</i>	
<a href="#">OTS_GuildRank::getName()</a>	131
<i>Rank name.</i>	
<a href="#">OTS_GuildRank::getLevel()</a>	131
<i>Rank's access level.</i>	
<a href="#">OTS_GuildRank::getIterator()</a>	130
<i>Returns players iterator.</i>	
<a href="#">OTS_GuildRank::getId()</a>	130
<i>Rank ID.</i>	
<a href="#">OTS_GuildRank::getPlayers()</a>	131
<i>Reads all players who has this rank set.</i>	
<a href="#">OTS_GuildRank::getPlayersList()</a>	132
<i>List of characters with current rank.</i>	
<a href="#">OTS_GuildRank::save()</a>	133
<i>Saves rank in database.</i>	
<a href="#">OTS_GuildRank::load()</a>	133
<i>Loads rank with given id.</i>	
<a href="#">OTS_GuildRank::isLoaded()</a>	132
<i>Checks if object is loaded.</i>	
<a href="#">OTS_InfoRespond::getClientVersion()</a>	139
<i>Returns dedicated version of client.</i>	
<a href="#">OTS_InfoRespond::getEmail()</a>	139
<i>Returns owner e-mail.</i>	
<a href="#">OTS_InfoRespond::getPort()</a>	144
<i>Returns server port.</i>	
<a href="#">OTS_InfoRespond::getPlayersPeak()</a>	143
<i>Returns record of online players.</i>	
<a href="#">OTS_InfoRespond::getOwner()</a>	143
<i>Returns owner name.</i>	
<a href="#">OTS_InfoRespond::getOnlinePlayers()</a>	143
<i>Returns current amount of players online.</i>	
<a href="#">OTS_InfoRespond::getServer()</a>	144
<i>Returns server attribute.</i>	
<a href="#">OTS_InfoRespond::getServerVersion()</a>	144
<i>Returns server version.</i>	
<a href="#">OTS_InfoRespond::getURL()</a>	145
<i>Returns server website.</i>	
<a href="#">OTS_InfoRespond::getUptime()</a>	145
<i>Returns server uptime.</i>	
<a href="#">OTS_InfoRespond::getTSPQVersion()</a>	145
<i>Returns version of root element.</i>	
<a href="#">OTS_InfoRespond::getName()</a>	142
<i>Returns server name.</i>	
<a href="#">OTS_InfoRespond::getMOTD()</a>	142
<i>Returns server's Message Of The Day</i>	
<a href="#">OTS_InfoRespond::getMapAuthor()</a>	140
<i>Returns map author.</i>	
<a href="#">OTS_InfoRespond::getLocation()</a>	140
<i>Returns server location.</i>	
<a href="#">OTS_InfoRespond::getIP()</a>	139
<i>Returns server IP.</i>	
<a href="#">OTS_InfoRespond::getMapHeight()</a>	140
<i>Returns map height.</i>	

<a href="#">OTS_InfoRespond::getMapName()</a>	141
<i>Returns map name.</i>	
<a href="#">OTS_InfoRespond::getMonstersCount()</a>	142
<i>Returns number of all monsters on map.</i>	
<a href="#">OTS_InfoRespond::getMaxPlayers()</a>	141
<i>Returns maximum amount of players online.</i>	
<a href="#">OTS_InfoRespond::getMapWidth()</a>	141
<i>Returns map width.</i>	
<a href="#">OTS_Player::getRank()</a>	164
<i>Assigned guild rank.</i>	
<a href="#">OTS_Player::getRankId()</a>	164
<i>Guild rank ID.</i>	
<a href="#">OTS_Player::setVocation()</a>	191
<i>Sets player proffesion.</i>	
<a href="#">OTS_Player::setTownId()</a>	191
<i>Sets residence town's ID.</i>	
<a href="#">OTS_Player::setSoul()</a>	190
<i>Sets soul points.</i>	
<a href="#">OTS_Player::setSlot()</a>	190
<i>Sets slot content.</i>	
<a href="#">OTS_Player::unban()</a>	192
<i>Deletes ban from current player.</i>	
<a href="#">OTS_Player::unsetRedSkull()</a>	192
<i>Unsets red skull flag.</i>	
<a href="#">OTS_Players_List</a>	193
<i>List of players.</i>	
<a href="#">OTS_Player::sleep()</a>	193
<i>Magic PHP5 method.</i>	
<a href="#">OTS_Player::unsetSave()</a>	192
<i>Unsets save flag.</i>	
<a href="#">OTS_Player::setSkillTries()</a>	189
<i>Sets skill's tries for next level.</i>	
<a href="#">OTS_Player::setSkill()</a>	188
<i>Sets skill value.</i>	
<a href="#">OTS_Player::setRank()</a>	186
<i>Assigns guild rank.</i>	
<a href="#">OTS_Player::setPremiumEnd()</a>	185
<i>Sets player's Premium Account expiration timestamp.</i>	
<a href="#">OTS_Player::setPosZ()</a>	185
<i>Sets Z map coordinate.</i>	
<a href="#">OTS_Player::setPosY()</a>	184
<i>Sets Y map coordinate.</i>	
<a href="#">OTS_Player::setRankId()</a>	186
<i>Sets guild rank ID.</i>	
<a href="#">OTS_Player::setRedSkull()</a>	187
<i>Sets red skull flag.</i>	
<a href="#">OTS_Player::setSex()</a>	188
<i>Sets player gender.</i>	
<a href="#">OTS_Player::setSave()</a>	188
<i>Sets save flag.</i>	
<a href="#">OTS_Player::setRedSkullTime()</a>	187
<i>Sets red skulled time remained.</i>	
<a href="#">OTS_Players_List::deletePlayer()</a>	193

<i>Deletes player.</i>	
<a href="#">OTS Players List::init()</a>	194
<i>Sets list parameters.</i>	
<a href="#">OTS SQLFilter::OPERATOR NLOWER</a>	199
<i>Not-lower-then operator.</i>	
<a href="#">OTS SQLFilter::OPERATOR NLIKE</a>	199
<i>Not-LIKE operator.</i>	
<a href="#">OTS SQLFilter::OPERATOR NGREATER</a>	199
<i>Not-greater-then operator.</i>	
<a href="#">OTS SQLFilter::OPERATOR NEQUAL</a>	198
<i>Not-equal operator.</i>	
<a href="#">OTS SQLFilter::addFilter()</a>	200
<i>General-purpose filter.</i>	
<a href="#">OTS SQLFilter::compareField()</a>	200
<i>Compares field with a literal value.</i>	
<a href="#">OTS SQLFilter::toString()</a>	202
<i>Returns string representation of WHERE clause.</i>	
<a href="#">OTS SQLFilter::sleep()</a>	201
<i>Magic PHP5 method.</i>	
<a href="#">OTS SQLFilter::getTables()</a>	201
<i>Returns list of all tables used by filter.</i>	
<a href="#">OTS SQLFilter::OPERATOR LOWER</a>	198
<i>Lower-then operator.</i>	
<a href="#">OTS SQLFilter::OPERATOR LIKE</a>	198
<i>LIKE operator.</i>	
<a href="#">OTS SQLField::getTable()</a>	196
<i>Returns table name.</i>	
<a href="#">OTS SQLField::getName()</a>	195
<i>Returns field name.</i>	
<a href="#">OTS SQLField</a>	194
<i>SQL identifier representation.</i>	
<a href="#">OTS SQLFilter</a>	196
<i>SQL WHERE clause object.</i>	
<a href="#">OTS SQLFilter::CRITERIUM AND</a>	196
<i>AND sibling.</i>	
<a href="#">OTS SQLFilter::OPERATOR GREATER</a>	197
<i>Greater-then operator.</i>	
<a href="#">OTS SQLFilter::OPERATOR EQUAL</a>	197
<i>Equal operator.</i>	
<a href="#">OTS SQLFilter::CRITERIUM OR</a>	197
<i>OR sibling.</i>	
<a href="#">OTS Player::setPosX()</a>	184
<i>Sets X map coordinate.</i>	
<a href="#">OTS Player::setName()</a>	184
<i>Sets players's name.</i>	
<a href="#">OTS Player::setAccount()</a>	171
<i>Assigns character to account.</i>	
<a href="#">OTS Player::save()</a>	170
<i>Saves player in database.</i>	
<a href="#">OTS Player::load()</a>	170
<i>Loads player with given id.</i>	
<a href="#">OTS Player::isSaveSet()</a>	170
<i>Checks if save flag is set.</i>	

<a href="#">OTS_Player::setCap()</a>	171
<i>Sets capacity.</i>	
<a href="#">OTS_Player::setConditions()</a>	172
<i>Sets conditions.</i>	
<a href="#">OTS_Player::setDirection()</a>	174
<i>Sets looking direction.</i>	
<a href="#">OTS_Player::setDepot()</a>	173
<i>Sets depot content.</i>	
<a href="#">OTS_Player::setCustomField()</a>	172
<i>Writes custom field.</i>	
<a href="#">OTS_Player::isLoading()</a>	169
<i>Checks if object is loaded.</i>	
<a href="#">OTS_Player::isBanned()</a>	169
<i>Checks if player is banned.</i>	
<a href="#">OTS_Player::getSkillTries()</a>	166
<i>Returns player's skill's tries for next level.</i>	
<a href="#">OTS_Player::getSkill()</a>	166
<i>Returns player's skill.</i>	
<a href="#">OTS_Player::getSex()</a>	165
<i>Player gender.</i>	
<a href="#">OTS_Player::getRedSkullTime()</a>	165
<i>Red skulled time remained.</i>	
<a href="#">OTS_Player::getSlot()</a>	167
<i>Returns items tree from given slot.</i>	
<a href="#">OTS_Player::getSoul()</a>	167
<i>Soul points.</i>	
<a href="#">OTS_Player::hasRedSkull()</a>	168
<i>Checks if player has red skull.</i>	
<a href="#">OTS_Player::getVocation()</a>	168
<i>Player proffesion.</i>	
<a href="#">OTS_Player::getTownId()</a>	168
<i>Residence town's ID.</i>	
<a href="#">OTS_Player::setExperience()</a>	174
<i>Sets experience points.</i>	
<a href="#">OTS_Player::setGroup()</a>	175
<i>Assigns character to group.</i>	
<a href="#">OTS_Player::setLossMana()</a>	181
<i>Sets percentage of used mana lost after dead.</i>	
<a href="#">OTS_Player::setLossExperience()</a>	180
<i>Sets percentage of experience lost after dead.</i>	
<a href="#">OTS_Player::setLookType()</a>	180
<i>Sets outfit.</i>	
<a href="#">OTS_Player::setLookLegs()</a>	180
<i>Sets legs color.</i>	
<a href="#">OTS_Player::setLossSkills()</a>	181
<i>Sets percentage of skills lost after dead.</i>	
<a href="#">OTS_Player::setMagLevel()</a>	182
<i>Sets magic level.</i>	
<a href="#">OTS_Player::setManaSpent()</a>	183
<i>Sets mana spent.</i>	
<a href="#">OTS_Player::setManaMax()</a>	183
<i>Sets maximum mana.</i>	
<a href="#">OTS_Player::setMana()</a>	182



<i>Sets current mana.</i>	
<a href="#">OTS_Player::setLookHead()</a>	179
<i>Sets hair color.</i>	
<a href="#">OTS_Player::setLookFeet()</a>	179
<i>Sets boots color.</i>	
<a href="#">OTS_Player::setHealthMax()</a>	176
<i>Sets maximum HP.</i>	
<a href="#">OTS_Player::setHealth()</a>	175
<i>Sets current HP.</i>	
<a href="#">OTS_Player::setGuildNick()</a>	175
<i>Sets guild nick.</i>	
<a href="#">OTS_Player::setLastIP()</a>	176
<i>Sets last login IP.</i>	
<a href="#">OTS_Player::setLastLogin()</a>	177
<i>Sets last login timestamp.</i>	
<a href="#">OTS_Player::setLookBody()</a>	178
<i>Sets body color.</i>	
<a href="#">OTS_Player::setLookAddons()</a>	178
<i>Sets addons.</i>	
<a href="#">OTS_Player::setLevel()</a>	177
<i>Sets experience level.</i>	
<a href="#">OTS_GuildRank::getGuild()</a>	130
<i>Returns guild of this rank.</i>	
<a href="#">OTS_GuildRank::getCustomField()</a>	129
<i>Reads custom field.</i>	
<a href="#">OTS_Base_DAO:: set_state()</a>	76
<i>Magic PHP5 method.</i>	
<a href="#">OTS_Base_DAO:: clone()</a>	75
<i>Creates clone of object.</i>	
<a href="#">OTS_Base_DAO::\$db</a>	74
<i>Database connection.</i>	
<a href="#">OTS_Base_DAO</a>	74
<i>Basic data access object routines.</i>	
<a href="#">OTS_Base_DAO:: sleep()</a>	76
<i>Magic PHP5 method.</i>	
<a href="#">OTS_Base_DAO:: wakeup()</a>	76
<i>Magic PHP5 method.</i>	
<a href="#">OTS_Base_List::\$table</a>	78
<i>Default table name for queries.</i>	
<a href="#">OTS_Base_List::\$class</a>	77
<i>Class of generated objects.</i>	
<a href="#">OTS_Base_List</a>	77
<i>Basic list class routines.</i>	
<a href="#">OTS_Accounts_List::init()</a>	73
<i>Sets list parameters.</i>	
<a href="#">OTS_Accounts_List::deleteAccount()</a>	73
<i>Deletes account.</i>	
<a href="#">OTS_Account::setGroup()</a>	70
<i>Assigns account to group.</i>	
<a href="#">OTS_Account::setEMail()</a>	70
<i>Sets account's email.</i>	
<a href="#">OTS_Account::setCustomField()</a>	69
<i>Writes custom field.</i>	

<a href="#">OTS Account::save()</a>	69
<i>Updates account in database.</i>	
<a href="#">OTS Account::setPACCDays()</a>	71
<i>Sets PACC days count.</i>	
<a href="#">OTS Account::setPassword()</a>	71
<i>Sets account's password.</i>	
<a href="#">OTS Accounts List</a>	73
<i>List of accounts.</i>	
<a href="#">OTS Account::unblock()</a>	72
<i>Unblocks account.</i>	
<a href="#">OTS Account::unban()</a>	72
<i>Deletes ban from current account.</i>	
<a href="#">OTS Base List::count()</a>	78
<i>Returns number of accounts on list in current criterium.</i>	
<a href="#">OTS Base List::current()</a>	79
<i>Returns current row.</i>	
<a href="#">OTS Container::addItem()</a>	85
<i>Adds item to container.</i>	
<a href="#">OTS Container</a>	84
<i>Container item representation.</i>	
<a href="#">OTS Base List:: wakeup()</a>	84
<i>Magic PHP5 method.</i>	
<a href="#">OTS Base List:: sleep()</a>	83
<i>Magic PHP5 method.</i>	
<a href="#">OTS Container::count()</a>	85
<i>Number of items inside container.</i>	
<a href="#">OTS Container::current()</a>	86
<i>Returns current item.</i>	
<a href="#">OTS Container::removeItem()</a>	86
<i>Removes given item from current container.</i>	
<a href="#">OTS Container::next()</a>	86
<i>Moves to next item.</i>	
<a href="#">OTS Container::key()</a>	86
<i>Current cursor position.</i>	
<a href="#">OTS Base List:: set state()</a>	83
<i>Magic PHP5 method.</i>	
<a href="#">OTS Base List::valid()</a>	83
<i>Checks if there are any rows left.</i>	
<a href="#">OTS Base List::orderBy()</a>	80
<i>Appends sorting rule.</i>	
<a href="#">OTS Base List::next()</a>	80
<i>Moves to next row.</i>	
<a href="#">OTS Base List::key()</a>	79
<i>Current cursor position.</i>	
<a href="#">OTS Base List::init()</a>	79
<i>Sets list parameters.</i>	
<a href="#">OTS Base List::resetOrder()</a>	81
<i>Clears ORDER BY clause.</i>	
<a href="#">OTS Base List::rewind()</a>	81
<i>Select rows from database.</i>	
<a href="#">OTS Base List::setOffset()</a>	82
<i>Sets OFFSET.</i>	
<a href="#">OTS Base List::setLimit()</a>	82

Sets LIMIT.	
<a href="#">OTS Base List::setFilter()</a>	81
Sets filter on list.	
<a href="#">OTS Account::load()</a>	68
Loads account with given number.	
<a href="#">OTS Account::isLoading()</a>	68
Checks if object is loaded.	
<a href="#">OTS Guilds List.php</a>	42
<a href="#">OTS GuildRanks List.php</a>	41
<a href="#">OTS GuildRank.php</a>	40
<a href="#">OTS Guild.php</a>	39
<a href="#">OTS InfoRespond.php</a>	43
<a href="#">OTS Item.php</a>	44
<a href="#">OTS SQLField.php</a>	47
<a href="#">OTS Players List.php</a>	46
<a href="#">OTS Player.php</a>	45
<a href="#">OTS Groups List.php</a>	38
<a href="#">OTS Group.php</a>	37
<a href="#">OTS Base List.php</a>	31
<a href="#">OTS Base DAO.php</a>	30
<a href="#">OTS Accounts List.php</a>	29
<a href="#">OTS Account.php</a>	28
<a href="#">OTS Container.php</a>	32
<a href="#">OTS DB MySQL.php</a>	33
<a href="#">OTS DB SQLite.php</a>	36
<a href="#">OTS DB PostgreSQL.php</a>	35
<a href="#">OTS DB ODBC.php</a>	34
<a href="#">OTS SQLFilter.php</a>	48
<a href="#">OTS SQLite Results.php</a>	49
<a href="#">OTS Account::getPACCDays()</a>	65
PACC days.	
<a href="#">OTS Account::getIterator()</a>	65
Returns players iterator.	
<a href="#">OTS Account::getId()</a>	65
Account number.	
<a href="#">OTS Account::getGroup()</a>	64
Returns group of this account.	
<a href="#">OTS Account::getPassword()</a>	66
Account's password.	
<a href="#">OTS Account::getPlayers()</a>	66
List of characters on account.	
<a href="#">OTS Account::isBlocked()</a>	68
Checks if account is blocked.	
<a href="#">OTS Account::isBanned()</a>	67
Checks if account is banned.	
<a href="#">OTS Account::getPlayersList()</a>	67
List of characters on account.	
<a href="#">OTS Account::getEmail()</a>	64
E-mail address.	
<a href="#">OTS Account::getCustomField()</a>	63
Reads custom field.	
<a href="#">OTS Account::block()</a>	59
Blocks account.	

<a href="#">OTS Account::ban()</a>	59
<i>Bans current account.</i>	
<a href="#">OTS Account</a>	58
<i>OTServ account abstraction.</i>	
<a href="#">OTS Account::count()</a>	59
<i>Returns number of player within.</i>	
<a href="#">OTS Account::create()</a>	60
<i>Creates new account.</i>	
<a href="#">OTS Account::find()</a>	63
<i>Loads account by it's e-mail address.</i>	
<a href="#">OTS Account::delete()</a>	62
<i>Deletes account.</i>	
<a href="#">OTS Account::createEx()</a>	61
<i>Creates new account.</i>	
<a href="#">OTS Container::rewind()</a>	87
<i>Resets internal items array pointer.</i>	
<a href="#">OTS Container::valid()</a>	87
<i>Checks if there are any items left.</i>	
<a href="#">OTS Guild::getCreationData()</a>	117
<i>Guild creation data.</i>	
<a href="#">OTS Guild::find()</a>	116
<i>Loads guild by it's name.</i>	
<a href="#">OTS Guild::deleteRequest()</a>	116
<i>Deletes request from player.</i>	
<a href="#">OTS Guild::deleteInvite()</a>	115
<i>Deletes invitation for player to guild.</i>	
<a href="#">OTS Guild::getCustomField()</a>	117
<i>Reads custom field.</i>	
<a href="#">OTS Guild::getGuildRanks()</a>	118
<i>Reads all ranks that are in this guild.</i>	
<a href="#">OTS Guild::getIterator()</a>	119
<i>Returns ranks iterator.</i>	
<a href="#">OTS Guild::getId()</a>	119
<i>Guild ID.</i>	
<a href="#">OTS Guild::getGuildRanksList()</a>	118
<i>List of ranks in guild.</i>	
<a href="#">OTS Guild::delete()</a>	115
<i>Deletes guild.</i>	
<a href="#">OTS Guild::count()</a>	115
<i>Returns number of ranks within.</i>	
<a href="#">OTS Groups List</a>	112
<i>List of groups.</i>	
<a href="#">OTS Group::setName()</a>	111
<i>Sets group's name.</i>	
<a href="#">OTS Group::setMaxVIPList()</a>	111
<i>Sets maximum count of players in VIP list.</i>	
<a href="#">OTS Group::setMaxDepotItems()</a>	110
<i>Sets maximum count of items in depot.</i>	
<a href="#">OTS Groups List::deleteGroup()</a>	112
<i>Deletes group.</i>	
<a href="#">OTS Groups List::init()</a>	113
<i>Sets list parameters.</i>	
<a href="#">OTS Guild::acceptRequest()</a>	114

Accepts player.	
<a href="#">OTS_Guild::acceptInvite()</a>	114
Finalise invitation.	
<a href="#">OTS_Guild</a>	113
OTServ guild abstraction.	
<a href="#">OTS_Guild::getName()</a>	120
Guild name.	
<a href="#">OTS_Guild::getOwner()</a>	120
Returns owning player of this player.	
<a href="#">OTS_Guild::clone()</a>	126
Creates clone of object.	
<a href="#">OTS_Guild::setRequestsDriver()</a>	126
Assigns requests handler.	
<a href="#">OTS_Guild::setOwner()</a>	125
Assigns guild to owner.	
<a href="#">OTS_Guild::setName()</a>	125
Sets players's name.	
<a href="#">OTS_Guild::sleep()</a>	127
Magic PHP5 method.	
<a href="#">OTS_GuildRank</a>	127
OTServ guild rank abstraction.	
<a href="#">OTS_GuildRank::find()</a>	128
Loads rank by it's name.	
<a href="#">OTS_GuildRank::delete()</a>	128
Deletes guild rank.	
<a href="#">OTS_GuildRank::count()</a>	128
Returns number of player within.	
<a href="#">OTS_Guild::setInvitesDriver()</a>	125
Assigns invites handler.	
<a href="#">OTS_Guild::setCustomField()</a>	124
Writes custom field.	
<a href="#">OTS_Guild::listInvites()</a>	121
Returns list of invited players.	
<a href="#">OTS_Guild::isLoading()</a>	121
Checks if object is loaded.	
<a href="#">OTS_Guild::invite()</a>	120
Invites player to guild.	
<a href="#">OTS_Guild::listRequests()</a>	122
Returns list of players that requested membership.	
<a href="#">OTS_Guild::load()</a>	122
Loads guild with given id.	
<a href="#">OTS_Guild::setCreationData()</a>	123
Sets guild creation data.	
<a href="#">OTS_Guild::save()</a>	123
Saves guild in database.	
<a href="#">OTS_Guild::request()</a>	122
Requests membership in guild for player player.	
<a href="#">OTS_Group::setFlags()</a>	110
Sets rights flags.	
<a href="#">OTS_Group::setCustomField()</a>	109
Writes custom field.	
<a href="#">OTS_DB_PostgreSQL::limit()</a>	96
LIMIT/OFFSET clause for queries.	

<a href="#">OTS_DB PostgreSQL::fieldName()</a>	96
<i>Query-quoted field name.</i>	
<a href="#">OTS_DB PostgreSQL</a>	95
<i>PostgreSQL connection interface.</i>	
<a href="#">OTS_DB ODBC::tableName()</a>	94
<i>Query-quoted table name.</i>	
<a href="#">OTS_DB PostgreSQL::SQLquery()</a>	97
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB PostgreSQL::SQLquote()</a>	97
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB SQLite::fieldName()</a>	99
<i>Query-quoted field name.</i>	
<a href="#">OTS_DB SQLite</a>	98
<i>SQLite connection interface.</i>	
<a href="#">OTS_DB PostgreSQL::tableName()</a>	98
<i>Query-quoted table name.</i>	
<a href="#">OTS_DB ODBC::SQLquote()</a>	94
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB ODBC::SQLquery()</a>	93
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB MySQL::SQLquery()</a>	90
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB MySQL::limit()</a>	89
<i>LIMIT/OFFSET clause for queries.</i>	
<a href="#">OTS_DB MySQL::fieldName()</a>	89
<i>Query-quoted field name.</i>	
<a href="#">OTS_DB MySQL</a>	88
<i>MySQL connection interface.</i>	
<a href="#">OTS_DB MySQL::SQLquote()</a>	90
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB MySQL::tableName()</a>	91
<i>Query-quoted table name.</i>	
<a href="#">OTS_DB ODBC::limit()</a>	93
<i>LIMIT/OFFSET clause for queries.</i>	
<a href="#">OTS_DB ODBC::fieldName()</a>	92
<i>Query-quoted field name.</i>	
<a href="#">OTS_DB ODBC</a>	91
<i>ODBC connection interface.</i>	
<a href="#">OTS_DB SQLite::limit()</a>	100
<i>LIMIT/OFFSET clause for queries.</i>	
<a href="#">OTS_DB SQLite::SQLquery()</a>	100
<i>IOTS_DB method.</i>	
<a href="#">OTS_Group::getPlayers()</a>	107
<i>List of characters in given group.</i>	
<a href="#">OTS_Group::getName()</a>	106
<i>Group name.</i>	
<a href="#">OTS_Group::getMaxVIPList()</a>	106
<i>Maximum count of players in VIP list.</i>	
<a href="#">OTS_Group::getMaxDepotItems()</a>	105
<i>Maximum count of items in depot.</i>	
<a href="#">OTS_Group::getPlayersList()</a>	107
<i>List of characters in group.</i>	
<a href="#">OTS_Group::isLoading()</a>	107

<i>Checks if object is loaded.</i>	
<a href="#">OTS_Group::setAccess()</a>	109
<i>Sets access level.</i>	
<a href="#">OTS_Group::save()</a>	108
<i>Saves account in database.</i>	
<a href="#">OTS_Group::load()</a>	108
<i>Loads group with given id.</i>	
<a href="#">OTS_Group::getIterator()</a>	105
<i>Returns players iterator.</i>	
<a href="#">OTS_Group::getId()</a>	105
<i>Group ID.</i>	
<a href="#">OTS_Group</a>	102
<i>OTServ user group abstraction.</i>	
<a href="#">OTS_DB_SQLite::tableName()</a>	101
<i>Query-quoted table name.</i>	
<a href="#">OTS_DB_SQLite::SQLquote()</a>	101
<i>IOTS_DB method.</i>	
<a href="#">OTS_Group::count()</a>	102
<i>Returns number of player within.</i>	
<a href="#">OTS_Group::delete()</a>	103
<i>Deletes group.</i>	
<a href="#">OTS_Group::getFlags()</a>	104
<i>Rights flags.</i>	
<a href="#">OTS_Group::getCustomField()</a>	104
<i>Reads custom field.</i>	
<a href="#">OTS_Group::getAccess()</a>	103
<i>Access level.</i>	
<a href="#">OTS.php</a>	27
<i>This file contains main toolkit class.</i>	

## P

<a href="#">POT::VOCATION_KNIGHT</a>	216
<i>Knight.</i>	
<a href="#">POT::VOCATION_DRUID</a>	215
<i>Druid.</i>	
<a href="#">POT::VOCATION_NONE</a>	216
<i>None vocation.</i>	
<a href="#">POT::VOCATION_PALADIN</a>	217
<i>Paladin.</i>	
<a href="#">POT::banIP()</a>	217
<i>Bans given IP number.</i>	
<a href="#">POT::VOCATION_SORCERER</a>	217
<i>Sorcerer.</i>	
<a href="#">POT::SLOT_RING</a>	215
<i>Ring slot.</i>	
<a href="#">POT::SLOT_RIGHT</a>	215
<i>Right hand slot.</i>	
<a href="#">POT::SLOT_HEAD</a>	213
<i>Head slot.</i>	
<a href="#">POT::SLOT_FEET</a>	213
<i>Boots slot.</i>	

<a href="#"><u>POT::SLOT_LEFT</u></a>	213
<i>Left hand slot.</i>	
<a href="#"><u>POT::SLOT_LEGS</u></a>	214
<i>Legs slot.</i>	
<a href="#"><u>POT::SLOT_NECKLACE</u></a>	214
<i>Necklace slot.</i>	
<a href="#"><u>POT::connect()</u></a>	218
<i>Connects to database.</i>	
<a href="#"><u>POT::createFilter()</u></a>	219
<i>Creates lists filter.</i>	
<a href="#"><u>POT::loadVocations()</u></a>	223
<i>Loads vocations list.</i>	
<a href="#"><u>POT::loadClass()</u></a>	223
<i>Loads POT class file.</i>	
<a href="#"><u>POT::serverStatus()</u></a>	224
<i>Queries server status.</i>	
<a href="#"><u>POT::setPOTPath()</u></a>	225
<i>Set POT directory.</i>	
<a href="#"><u>POT::unbanIP()</u></a>	226
<i>Deletes ban from given IP number.</i>	
<a href="#"><u>POT::isIPBanned()</u></a>	222
<i>Checks if given IP is banned.</i>	
<a href="#"><u>POT::getVocationsList()</u></a>	222
<i>Returns list (id =&gt; name) of loaded vocations.</i>	
<a href="#"><u>POT::getDBHandle()</u></a>	220
<i>Returns database connection handle.</i>	
<a href="#"><u>POT::createObject()</u></a>	220
<i>Creates OTServ DAO class instance.</i>	
<a href="#"><u>POT::getInstance()</u></a>	221
<i>Singleton.</i>	
<a href="#"><u>POT::getVocationID()</u></a>	221
<i>Returns vocation's ID.</i>	
<a href="#"><u>POT::getVocationName()</u></a>	221
<i>Returns name of given vocation's ID.</i>	
<a href="#"><u>POT::SLOT_BACKPACK</u></a>	212
<i>Backpack slot.</i>	
<a href="#"><u>POT::SLOT_ARMOR</u></a>	212
<i>Armor slot.</i>	
<a href="#"><u>POT::DB_PGSQL</u></a>	204
<i>PostgreSQL driver.</i>	
<a href="#"><u>POT::DB_ODBC</u></a>	204
<i>ODBC driver.</i>	
<a href="#"><u>POT::DB_SQLITE</u></a>	205
<i>SQLite driver.</i>	
<a href="#"><u>POT::DEPOT_SID_FIRST</u></a>	205
<i>First depot item sid.</i>	
<a href="#"><u>POT::DIRECTION_NORTH</u></a>	206
<i>North.</i>	
<a href="#"><u>POT::DIRECTION_EAST</u></a>	206
<i>East.</i>	
<a href="#"><u>POT::DB_MYSQL</u></a>	204
<i>MySQL driver.</i>	
<a href="#"><u>POT::BAN_PLAYER</u></a>	203



<i>Player ban.</i>	
<a href="#">POT class preview</a>	5
<a href="#">PHP 5.0</a>	3
<a href="#">POT</a>	202
<i>Main POT class.</i>	
<a href="#">POT::BAN_ACCOUNT</a>	203
<i>Account ban.</i>	
<a href="#">POT::BAN_IP</a>	203
<i>IP ban.</i>	
<a href="#">POT::DIRECTION_SOUTH</a>	206
<i>South.</i>	
<a href="#">POT::DIRECTION_WEST</a>	207
<i>West.</i>	
<a href="#">POT::SKILL_FIST</a>	210
<i>Fist fighting.</i>	
<a href="#">POT::SKILL_FISHING</a>	210
<i>Fishing.</i>	
<a href="#">POT::SKILL_SHIELDING</a>	211
<i>Shielding.</i>	
<a href="#">POT::SKILL_SWORD</a>	211
<i>Sword fighting.</i>	
<a href="#">POT::SLOT_AMMO</a>	211
<i>Ammunition slot.</i>	
<a href="#">POT::SKILL_DISTANCE</a>	209
<i>Distance fighting.</i>	
<a href="#">POT::SKILL_CLUB</a>	209
<i>Club fighting.</i>	
<a href="#">POT::ORDER_DESC</a>	207
<i>Descending sorting order.</i>	
<a href="#">POT::ORDER_ASC</a>	207
<i>Ascencind sorting order.</i>	
<a href="#">POT::SEX_FEMALE</a>	208
<i>Female gender.</i>	
<a href="#">POT::SEX_MALE</a>	208
<i>Male gender.</i>	
<a href="#">POT::SKILL_AXE</a>	208
<i>Axe fighting.</i>	
<a href="#">POT</a>	1

## Q

<a href="#">Quick start</a>	6
-----------------------------	---

## R

<a href="#">README</a>	234
------------------------	-----

## S

<a href="#">Server online status</a>	19
--------------------------------------	----