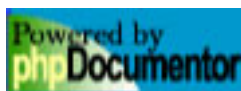


# POT



# Contents

<a href="#">POT</a>	1
<a href="#">POT class preview</a>	3
<a href="#">Quick start</a>	5
<a href="#">Account number hack</a>	9
<a href="#">Package POT Procedural Elements</a>	11
<a href="#">IOTS DAO.php</a>	11
<a href="#">IOTS DB.php</a>	12
<a href="#">OTS.php</a>	13
<a href="#">OTS Account.php</a>	14
<a href="#">OTS Accounts List.php</a>	15
<a href="#">OTS DB MySQL.php</a>	16
<a href="#">OTS DB SQLite.php</a>	17
<a href="#">OTS Group.php</a>	18
<a href="#">OTS Groups List.php</a>	19
<a href="#">OTS Player.php</a>	20
<a href="#">OTS Players List.php</a>	21
<a href="#">OTS SQLite Results.php</a>	22
<a href="#">Package POT Classes</a>	23
<a href="#">Class IOTS DAO</a>	23
<a href="#">Constructor construct</a>	23
<a href="#">Class IOTS DB</a>	24
<a href="#">Constructor construct</a>	24
<a href="#">Method fieldName</a>	24
<a href="#">Method lastInsertId</a>	25
<a href="#">Method limit</a>	25
<a href="#">Method SQLquery</a>	26
<a href="#">Method SQLquote</a>	26
<a href="#">Method tableName</a>	26
<a href="#">Class OTS Account</a>	27
<a href="#">Constructor construct</a>	27
<a href="#">Method block</a>	28
<a href="#">Method create</a>	28
<a href="#">example: account.php</a>	28
<a href="#">Method find</a>	29
<a href="#">Method getEmail</a>	30
<a href="#">Method getId</a>	30
<a href="#">Method getPACCDays</a>	30
<a href="#">Method getPassword</a>	30
<a href="#">Method getPlayers</a>	31
<a href="#">Method isBlocked</a>	31
<a href="#">Method isLoaded</a>	31
<a href="#">Method load</a>	32

<a href="#">Method save</a>	32
<a href="#">Method setEmail</a>	32
<a href="#">Method setPACCDays</a>	33
<a href="#">Method setPassword</a>	33
<a href="#">Method unblock</a>	34
<a href="#">Class OTS_Accounts_List</a>	34
<a href="#">Constructor construct</a>	34
<a href="#">Method count</a>	35
<a href="#">Method current</a>	35
<a href="#">Method deleteAccount</a>	35
<a href="#">Method key</a>	36
<a href="#">Method next</a>	36
<a href="#">Method rewind</a>	36
<a href="#">Method setLimit</a>	37
<a href="#">Method setOffset</a>	37
<a href="#">Method valid</a>	37
<a href="#">Class OTS_DB_MySQL</a>	38
<a href="#">Constructor construct</a>	38
<a href="#">Method fieldName</a>	39
<a href="#">Method limit</a>	39
<a href="#">Method SQLquery</a>	40
<a href="#">Method SQLquote</a>	40
<a href="#">Method tableName</a>	40
<a href="#">Class OTS_DB_SQLite</a>	41
<a href="#">Constructor construct</a>	41
<a href="#">Method fieldName</a>	42
<a href="#">Method limit</a>	42
<a href="#">Method regexp</a>	43
<a href="#">Method SQLquery</a>	43
<a href="#">Method SQLquote</a>	44
<a href="#">Method tableName</a>	44
<a href="#">Class OTS_Group</a>	45
<a href="#">Constructor construct</a>	45
<a href="#">Method getAccess</a>	45
<a href="#">Method getFlags</a>	46
<a href="#">Method getId</a>	46
<a href="#">Method getMaxDepotItems</a>	46
<a href="#">Method getMaxVIPList</a>	46
<a href="#">Method getName</a>	47
<a href="#">Method getPlayers</a>	47
<a href="#">Method isLoaded</a>	47
<a href="#">Method load</a>	48
<a href="#">Method save</a>	48
<a href="#">Method setAccess</a>	48
<a href="#">Method setFlags</a>	49
<a href="#">Method setMaxDepotItems</a>	49
<a href="#">Method setMaxVIPList</a>	50
<a href="#">Method setName</a>	50
<a href="#">Class OTS_Groups_List</a>	51

<a href="#">Constructor</a>	<a href="#">construct</a>	51
<a href="#">Method</a>	<a href="#">count</a>	51
<a href="#">Method</a>	<a href="#">current</a>	52
<a href="#">Method</a>	<a href="#">deleteGroup</a>	52
<a href="#">Method</a>	<a href="#">key</a>	52
<a href="#">Method</a>	<a href="#">next</a>	53
<a href="#">Method</a>	<a href="#">rewind</a>	53
<a href="#">Method</a>	<a href="#">setLimit</a>	53
<a href="#">Method</a>	<a href="#">setOffset</a>	54
<a href="#">Method</a>	<a href="#">valid</a>	54
<a href="#">Class</a>	<a href="#">OTS_Player</a>	54
<a href="#">Constructor</a>	<a href="#">construct</a>	55
<a href="#">Method</a>	<a href="#">find</a>	55
<a href="#">Method</a>	<a href="#">getAccount</a>	56
<a href="#">Method</a>	<a href="#">getCap</a>	56
<a href="#">Method</a>	<a href="#">getConditions</a>	56
<a href="#">Method</a>	<a href="#">getDirection</a>	56
<a href="#">Method</a>	<a href="#">getExperience</a>	57
<a href="#">Method</a>	<a href="#">getGroup</a>	57
<a href="#">Method</a>	<a href="#">getGuildNick</a>	57
<a href="#">Method</a>	<a href="#">getHealth</a>	58
<a href="#">Method</a>	<a href="#">getHealthMax</a>	58
<a href="#">Method</a>	<a href="#">getId</a>	58
<a href="#">Method</a>	<a href="#">getLastIP</a>	58
<a href="#">Method</a>	<a href="#">getLastLogin</a>	59
<a href="#">Method</a>	<a href="#">getLevel</a>	59
<a href="#">Method</a>	<a href="#">getLookAddons</a>	59
<a href="#">Method</a>	<a href="#">getLookBody</a>	60
<a href="#">Method</a>	<a href="#">getLookFeet</a>	60
<a href="#">Method</a>	<a href="#">getLookHead</a>	60
<a href="#">Method</a>	<a href="#">getLookLegs</a>	61
<a href="#">Method</a>	<a href="#">getLookType</a>	61
<a href="#">Method</a>	<a href="#">getLossExperience</a>	61
<a href="#">Method</a>	<a href="#">getLossMana</a>	61
<a href="#">Method</a>	<a href="#">getLossSkills</a>	62
<a href="#">Method</a>	<a href="#">getMagLevel</a>	62
<a href="#">Method</a>	<a href="#">getMana</a>	62
<a href="#">Method</a>	<a href="#">getManaMax</a>	63
<a href="#">Method</a>	<a href="#">getManaSpent</a>	63
<a href="#">Method</a>	<a href="#">getName</a>	63
<a href="#">Method</a>	<a href="#">getPosX</a>	63
<a href="#">Method</a>	<a href="#">getPosY</a>	64
<a href="#">Method</a>	<a href="#">getPosZ</a>	64
<a href="#">Method</a>	<a href="#">getRankId</a>	64
<a href="#">Method</a>	<a href="#">getRedSkullTime</a>	65
<a href="#">Method</a>	<a href="#">getSex</a>	65
<a href="#">Method</a>	<a href="#">getSkill</a>	65
<a href="#">Method</a>	<a href="#">getSkillTries</a>	66
<a href="#">Method</a>	<a href="#">getSoul</a>	66

<a href="#">Method getTownId</a>	66
<a href="#">Method getVocation</a>	67
<a href="#">Method hasRedSkull</a>	67
<a href="#">Method isLoading</a>	67
<a href="#">Method isSaveSet</a>	67
<a href="#">Method load</a>	68
<a href="#">Method save</a>	68
<a href="#">Method setAccount</a>	69
<a href="#">Method setCap</a>	69
<a href="#">Method setConditions</a>	69
<a href="#">Method setDirection</a>	70
<a href="#">Method setExperience</a>	70
<a href="#">Method setGroup</a>	71
<a href="#">Method setGuildNick</a>	71
<a href="#">Method setHealth</a>	71
<a href="#">Method setHealthMax</a>	72
<a href="#">Method setLastIP</a>	72
<a href="#">Method setLastLogin</a>	73
<a href="#">Method setLevel</a>	73
<a href="#">Method setLookAddons</a>	74
<a href="#">Method setLookBody</a>	74
<a href="#">Method setLookFeet</a>	74
<a href="#">Method setLookHead</a>	75
<a href="#">Method setLookLegs</a>	75
<a href="#">Method setLookType</a>	76
<a href="#">Method setLossExperience</a>	76
<a href="#">Method setLossMana</a>	76
<a href="#">Method setLossSkills</a>	77
<a href="#">Method setMagLevel</a>	77
<a href="#">Method setMana</a>	78
<a href="#">Method setManaMax</a>	78
<a href="#">Method setManaSpent</a>	79
<a href="#">Method setName</a>	79
<a href="#">Method setPosX</a>	79
<a href="#">Method setPosY</a>	80
<a href="#">Method setPosZ</a>	80
<a href="#">Method setRankId</a>	81
<a href="#">Method setRedSkull</a>	81
<a href="#">Method setRedSkullTime</a>	81
<a href="#">Method setSave</a>	82
<a href="#">Method setSex</a>	82
<a href="#">Method setSkill</a>	82
<a href="#">Method setSkillTries</a>	83
<a href="#">Method setSoul</a>	83
<a href="#">Method setTownId</a>	84
<a href="#">Method setVocation</a>	84
<a href="#">Method unsetRedSkull</a>	85
<a href="#">Method unsetSave</a>	85
<a href="#">Class OTS_Players_List</a>	85

<a href="#">Constructor <code>construct</code></a>	86
<a href="#">Method <code>count</code></a>	86
<a href="#">Method <code>current</code></a>	86
<a href="#">Method <code>deletePlayer</code></a>	87
<a href="#">Method <code>key</code></a>	87
<a href="#">Method <code>next</code></a>	87
<a href="#">Method <code>rewind</code></a>	88
<a href="#">Method <code>setLimit</code></a>	88
<a href="#">Method <code>setOffset</code></a>	88
<a href="#">Method <code>valid</code></a>	89
<a href="#">Class <code>POT</code></a>	89
<a href="#">Class Constant <code>DB_MYSQL</code></a>	90
<a href="#">Class Constant <code>DB_SQLITE</code></a>	90
<a href="#">Class Constant <code>DIRECTION_EAST</code></a>	90
<a href="#">Class Constant <code>DIRECTION_NORTH</code></a>	91
<a href="#">Class Constant <code>DIRECTION_SOUTH</code></a>	91
<a href="#">Class Constant <code>DIRECTION_WEST</code></a>	91
<a href="#">Class Constant <code>SEX_FEMALE</code></a>	91
<a href="#">Class Constant <code>SEX_MALE</code></a>	92
<a href="#">Class Constant <code>SKILL_AXE</code></a>	92
<a href="#">Class Constant <code>SKILL_CLUB</code></a>	92
<a href="#">Class Constant <code>SKILL_DISTANCE</code></a>	93
<a href="#">Class Constant <code>SKILL_FISHING</code></a>	93
<a href="#">Class Constant <code>SKILL_FIST</code></a>	93
<a href="#">Class Constant <code>SKILL_SHIELDING</code></a>	94
<a href="#">Class Constant <code>SKILL_SWORD</code></a>	94
<a href="#">Class Constant <code>VOCATION_DRUID</code></a>	94
<a href="#">Class Constant <code>VOCATION_KNIGHT</code></a>	95
<a href="#">Class Constant <code>VOCATION_NONE</code></a>	95
<a href="#">Class Constant <code>VOCATION_PALADIN</code></a>	95
<a href="#">Class Constant <code>VOCATION_SORCERER</code></a>	96
<a href="#">Constructor <code>construct</code></a>	96
<a href="#">Method <code>connect</code></a>	96
<a href="#">example: <code>connect.php</code></a>	96
<a href="#">Method <code>createObject</code></a>	97
<a href="#">Method <code>getInstance</code></a>	98
<a href="#">Method <code>loadClass</code></a>	98
<a href="#">example: <code>autoload.php</code></a>	98
<a href="#">Method <code>setPOTPath</code></a>	99
<a href="#">example: <code>fakeroot.php</code></a>	99

<a href="#">Appendices</a>	101
<a href="#">Appendix A - Class Trees</a>	102
<a href="#">POT</a>	102
<a href="#">Appendix B - README/CHANGELOG/INSTALL</a>	104
<a href="#">INSTALL</a>	105
<a href="#">CHANGELOG</a>	105
<a href="#">NEWS</a>	105
<a href="#">README</a>	105

# POT

*This is documentation 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 won't work with old database structure as well as 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 its 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>
```



# 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](#). You can combine [POT class loading mechanism](#) with it. For 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 // includes POT main file
12 include( './classes/OTS.php' );
13
14 function __autoload( $class )
15 {
16     // checks if it's POT class
17     if( preg_match( '/^OTS_/', $class ) != 0 )
18     {
19         POT::getInstance\(\)-> loadClass( $class );
20     }
21 }
22 // possibly call your own __autoload() handler
23 else
24 {
25     here comes your stuff...
26 }
27 */
28 }
29
30 ?>
```

## 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 (we also encourage you to 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 // includes POT main file
12 include( './classes/OTS.php' );
13
14 // for further POT classes
15 function __autoload( $class )
16 {
17     // checks if it's POT class
18     if ( preg_match( '/^?OTS_', $class ) != 0 )
19     {
20         POT::getInstance()-> loadClass( $class );
21     }
22 }
23 // possibly call your own __autoload() handler
24 else
25 {
26     here comes your stuff...
27 }
28 */
29 }
30
31 // database configuration - can be simply moved to external file, eg. config.php
32 $config= array(
33     'driver' => POT::DB_MYSQL,
34     'host' => 'localhost',
35     'user' => 'wrzasq',
36     'database' => 'otserv'
37 );
38
39 // creates POT instance (or get existing one)
40 $ots= POT::getInstance();
41 $ots-> connect( null, $config );
42
43 ?>
```

## 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-> isLoaded() )
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. Through 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( $players as $index=> $player)
26 {
27     // each returned item is instance of OTS_Player class
28     echo (2 + $index) . ': ' . $player-> getName(), "\n" ;

```

```
29 }  
30  
31 ?>
```

# 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 ?>
```





# Package POT Procedural Elements

## IOTS\_DAO.php

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

# IOTS\_DB.php

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

# OTS\_Account.php

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

# OTS\_Accounts\_List.php

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

# OTS\_DB\_MySQL.php

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

# OTS\_DB\_SQLite.php

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

# OTS\_Group.php

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



# OTS\_Groups\_List.php

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

# OTS\_Player.php

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

# OTS\_Players\_List.php

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

# Package POT Classes

## Class IOTS\_DAO

*[line 21]*

### **OTserv database object.**

OTserv database object.

This interface indicates that class is a OTServ DAO class.

- **Package** POT
- **Version** 0.0.1

Constructor *void* function IOTS\_DAO::\_\_construct(\$db) *[line 28]*

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

- [\*IOTS\\_DB\*](#) **\$db** Database connection object.

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

DAO objects must be initialized with a database.

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

# Class IOTS\_DB

[line 21]

## OTServ database handler interface.

OTServ database handler interface.

This interface specifies routines requires by DAO classes.

- **Package** POT
- **Version** 0.0.1

Constructor *void* function IOTS\_DB::\_\_construct(\$params) [line 28]

### **Function Parameters:**

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

## Connection parameters.

Connection parameters.

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

*string* function IOTS\_DB::fieldName(\$name) [line 36]

### **Function Parameters:**

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

### Query-quoted field name.

Query-quoted field name.

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

*int* function IOTS\_DB::lastInsertId() [*line 63*]

### ID of last created record.

ID of last created record.

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

*string* function IOTS\_DB::limit([\$limit = false], [\$offset = false]) [*line 71*]

### 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
- **Access** public

*mixed* function IOTS\_DB::SQLquery(\$query) [*line 57*]

**Function Parameters:**

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

**Evaluates query.**

Evaluates query.

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

*string* function IOTS\_DB::SQLquote(\$value) [*line 50*]

**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
- **Access** public

*string* function IOTS\_DB::tableName(\$name) [*line 43*]

**Function Parameters:**

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



### Query-quoted table name.

Query-quoted table name.

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

## Class OTS\_Account

*[line 21]*

### OTServ account abstraction.

OTServ account abstraction.

- **Package** POT
- **Version** 0.0.1
- **Version** 0.0.1+SVN

Constructor *void* function OTS\_Account::\_\_construct(\$db) *[line 42]*

#### **Function Parameters:**

- [\*IOTS\\_DB\*](#) \$db Database connection object.

### Sets database connection handler.

Sets database connection handler.

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

*void* function OTS\_Account::block() [*line 262*]

## **Blocks account.**

Blocks account.

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

*int* function OTS\_Account::create([\$min = 1], [\$max = 9999999]) [*line 62*]

## **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!

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

*void* function OTS\_Account::find(\$email) [*line 126*]

#### **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** 0.0.1+SVN
- **Version** 0.0.1
- **Access** public

*string/bool* function OTS\_Account::getEmail() [*line 214*]

**E-mail address.**

E-mail address.

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

*int/bool* function OTS\_Account::getId() [*line 172*]

**Account number.**

Account number.

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

*int/bool* function OTS\_Account::getPACCDays() [*line 272*]

**PACC days.**

PACC days.

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

*string/bool* function OTS\_Account::getPassword() [*line 188*]

**Account's password.**

Account's password.

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

*array|bool* function OTS\_Account::getPlayers() [*line 298*]

**List of characters on account.**

List of characters on account.

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

*bool|null* function OTS\_Account::isBlocked() [*line 240*]

**Checks if account is blocked.**

Checks if account is blocked.

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

*bool* function OTS\_Account::isLoaded() [*line 143*]

**Checks if object is loaded.**

Checks if object is loaded.

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

*void* function OTS\_Account::load(\$id) [*line 114*]

**Function Parameters:**

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

**Loads account with given number.**

Loads account with given number.

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

*bool* function OTS\_Account::save() [*line 153*]

**Updates account in database.**

Updates account in database.

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

*void* function OTS\_Account::setEMail(\$email) [*line 230*]

**Function Parameters:**

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

**Sets account's email.**

Sets account's email.

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

*void function* OTS\_Account::setPACCDays(\$premdays, \$pacc) [*line 288*]

***Function Parameters:***

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

**Sets PACC days count.**

Sets PACC days count.

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

*void function* OTS\_Account::setPassword(\$password) [*line 204*]

***Function Parameters:***

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

**Sets account's password.**

Sets account's password.

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

*void function* OTS\_Account::unlock() [*line 254*]

**Unblocks account.**

Unblocks account.

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

## Class OTS\_Accounts\_List

[*line 19*]

**List of accounts.**

List of accounts.

- **Package** POT
- **Version** 0.0.1

Constructor *void function* OTS\_Accounts\_List::\_\_construct(\$db) [*line 54*]

**Function Parameters:**

- [\*IOTS\\_DB\*](#) **\$db** Database connection object.

**Sets database connection handler.**

Sets database connection handler.



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

*int* function OTS\_Accounts\_List::count() [*line 166*]

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

Returns number of accounts on list in current criterium.

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

*OTS\_Account* function OTS\_Accounts\_List::current() [*line 116*]

**Returns current row.**

Returns current row.

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

*bool* function OTS\_Accounts\_List::deleteAccount(\$account) [*line 99*]

**Function Parameters:**

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

**Deletes account.**

Deletes account.

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

*mixed* function OTS\_Accounts\_List::key() [*line 138*]

**Current cursor position.**

Current cursor position.

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

*void* function OTS\_Accounts\_List::next() [*line 128*]

**Moves to next row.**

Moves to next row.

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

*void* function OTS\_Accounts\_List::rewind() [*line 156*]

**Select accounts from database.**

Select accounts from database.

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

*void function* OTS\_Accounts\_List::setLimit([\$limit = false]) [*line 64*]

**Function Parameters:**

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

**Sets LIMIT.**

Sets LIMIT.

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

*void function* OTS\_Accounts\_List::setOffset([\$offset = false]) [*line 81*]

**Function Parameters:**

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

**Sets OFFSET.**

Sets OFFSET.

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

*bool function* OTS\_Accounts\_List::valid() [*line 148*]

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

Checks if there are any rows left.

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

## Class OTS\_DB\_MySQL

[line 19]

### MySQL connection interface.

MySQL connection interface.

- **Package** POT
- **Version** 0.0.1

Constructor *void* function OTS\_DB\_MySQL::\_\_construct(\$params) [line 46]

#### **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\(\)](#)
- **Access** public

*string* function OTS\_DB\_MySQL::fieldName(\$name) [*line 101*]

**Function Parameters:**

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

**Query-quoted field name.**

Query-quoted field name.

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

*string* function OTS\_DB\_MySQL::limit([\$limit = false], [\$offset = false]) [*line 152*]

**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
- **Access** public

*PDOStatement|bool* function OTS\_DB\_MySQL::SQLquery(\$query) [*line 140*]

**Function Parameters:**

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

**IOTS\_DB method.**

IOTS\_DB method.

Overwrites PDO method.

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

*string* function OTS\_DB\_MySQL::SQLquote(\$string) [*line 126*]

**Function Parameters:**

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

**IOTS\_DB method.**

IOTS\_DB method.

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

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

*string* function OTS\_DB\_MySQL::tableName(\$name) [*line 112*]

**Function Parameters:**

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

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

Query-quoted table name.

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

## Class OTS\_DB\_SQLite

[line 19]

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

SQLite connection interface.

- **Package** POT
- **Version** 0.0.1

Constructor *void* function OTS\_DB\_SQLite::\_\_construct(\$params) [line 42]

#### **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\(\)](#)
- **Access** public

*string* function OTS\_DB\_SQLite::fieldName(\$name) [*line 64*]

**Function Parameters:**

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

### **Query-quoted field name.**

Query-quoted field name.

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

*string* function OTS\_DB\_SQLite::limit([\$limit = false], [\$offset = false]) [*line 128*]

**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
- **Access** public

*bool* function OTS\_DB\_SQLite::regexp(\$name, \$content) [*line 88*]

**Function Parameters:**

- *string* **\$name** Regular expression to test.
- *string* **\$content** String to test.

## **REGEXP operator for SQLite**

REGEXP operator for SQLite

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

*PDOStatement|bool* function OTS\_DB\_SQLite::SQLquery(\$query) [*line 116*]

**Function Parameters:**

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

## **IOTS\_DB method.**

IOTS\_DB method.  
Overwrites PDO method.

- **Version** 0.0.1

- **Access** public

*string* function OTS\_DB\_SQLite::SQLquote(\$string) [*line 102*]

**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
- **Access** public

*string* function OTS\_DB\_SQLite::tableName(\$name) [*line 75*]

**Function Parameters:**

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

**Query-quoted table name.**

Query-quoted table name.

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

# Class OTS\_Group

[line 19]

## OTServ user group abstraction.

OTServ user group abstraction.

- **Package** POT
- **Version** 0.0.1

Constructor *void* function OTS\_Group::\_\_construct(\$db) [line 40]

### **Function Parameters:**

- [\*IOTS\\_DB\*](#) **\$db** Database connection object.

## Sets database connection handler.

Sets database connection handler.

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

*int|bool* function OTS\_Group::getAccess() [line 160]

### **Access level.**

Access level.

- **Version** 0.0.1

- **Access** public

*int/bool* function OTS\_Group::getFlags() [*line 134*]

**Rights flags.**

Rights flags.

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

*int/bool* function OTS\_Group::getId() [*line 92*]

**Group ID.**

Group ID.

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

*int/bool* function OTS\_Group::getMaxDepotItems() [*line 186*]

**Maximum count of items in depot.**

Maximum count of items in depot.

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

*int/bool* function OTS\_Group::getMaxVIPList() [*line 212*]

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

Maximum count of players in VIP list.

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

*string/bool* function OTS\_Group::getName() [*line 108*]

**Group name.**

Group name.

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

*array/bool* function OTS\_Group::getPlayers() [*line 238*]

**List of characters in given group.**

List of characters in given group.

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

*bool* function OTS\_Group::isLoaded() [*line 61*]

**Checks if object is loaded.**

Checks if object is loaded.

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

*void function OTS\_Group::load(\$id) [line 50]*

**Function Parameters:**

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

**Loads group with given id.**

Loads group with given id.

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

*void function OTS\_Group::save() [line 69]*

**Saves account in database.**

Saves account in database.

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

*void function OTS\_Group::setAccess(\$access) [line 176]*

**Function Parameters:**

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

**Sets access level.**

Sets access level.

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

*void function OTS\_Group::setFlags(\$flags) [line 150]*

**Function Parameters:**

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

**Sets rights flags.**

Sets rights flags.

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

*void function OTS\_Group::setMaxDepotItems(\$maxdepotitems) [line 202]*

**Function Parameters:**

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

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

Sets maximum count of items in depot.

- **Version** 0.0.1

- **Access** public

*void function OTS\_Group::setMaxVIPList(\$maxviplist, \$maxdepotitems) [line 228]*

**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
- **Access** public

*void function OTS\_Group::setName(\$name) [line 124]*

**Function Parameters:**

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

**Sets group's name.**

Sets group's name.

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



# Class OTS\_Groups\_List

[line 19]

## List of groups.

List of groups.

- **Package** POT
- **Version** 0.0.1

Constructor *void* function OTS\_Groups\_List::\_\_construct(\$db) [line 54]

### **Function Parameters:**

- [\*IOTS\\_DB\*](#) **\$db** Database connection object.

## Sets database connection handler.

Sets database connection handler.

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

*int* function OTS\_Groups\_List::count() [line 166]

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

Returns number of groups on list in current criterium.

- **Version** 0.0.1

- **Access** public

*OTS\_Group* function OTS\_Groups\_List::current() [*line 116*]

**Returns current row.**

Returns current row.

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

*bool* function OTS\_Groups\_List::deleteGroup(\$group) [*line 99*]

**Function Parameters:**

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

**Deletes group.**

Deletes group.

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

*mixed* function OTS\_Groups\_List::key() [*line 138*]

**Current cursor position.**

Current cursor position.

- **Version** 0.0.1

- **Access** public

*void function OTS\_Groups\_List::next() [line 128]*

**Moves to next row.**

Moves to next row.

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

*void function OTS\_Groups\_List::rewind() [line 156]*

**Select groups from database.**

Select groups from database.

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

*void function OTS\_Groups\_List::setLimit([\$limit = false]) [line 64]*

**Function Parameters:**

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

**Sets LIMIT.**

Sets LIMIT.

- **Version** 0.0.1

- **Access** public

*void* function OTS\_Groups\_List::setOffset([\$offset = false]) [*line 81*]

**Function Parameters:**

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

**Sets OFFSET.**

Sets OFFSET.

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

*bool* function OTS\_Groups\_List::valid() [*line 148*]

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

Checks if there are any rows left.

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

## Class OTS\_Player

[*line 21*]

**OTServ character abstraction.**

OTServ character abstraction.

- **Package** POT
- **Version** 0.0.1
- **Version** 0.0.1+SVN

Constructor *void* function OTS\_Player::\_\_construct(\$db) [*line 51*]

**Function Parameters:**

- [\*IOTS\\_DB\*](#) **\$db** Database connection object.

**Sets database connection handler.**

Sets database connection handler.

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

*void* function OTS\_Player::find(\$name) [*line 82*]

**Function Parameters:**

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

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

Loads player by it's name.

- **Version** 0.0.1

- **Access** public

*OTS\_Account* function OTS\_Player::getAccount() [*line 180*]

**Returns account of this player.**

Returns account of this player.

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

*int/bool* function OTS\_Player::getCap() [*line 782*]

**Capacity.**

Capacity.

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

*mixed/bool* function OTS\_Player::getConditions() [*line 892*]

**Conditions.**

Conditions.

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

*int/bool* function OTS\_Player::getDirection() [*line 522*]

**Looking direction.**

Looking direction.

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

*int|bool* function OTS\_Player::getExperience() [*line 288*]

**Experience points.**

Experience points.

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

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

**Returns group of this player.**

Returns group of this player.

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

*string|bool* function OTS\_Player::getGuildNick() [*line 976*]

**Guild nick.**

Guild nick.

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

*int/bool* function OTS\_Player::getHealth() [*line 366*]

#### **Current HP.**

Current HP.

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

*int/bool* function OTS\_Player::getHealthMax() [*line 392*]

#### **Maximum HP.**

Maximum HP.

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

*int/bool* function OTS\_Player::getId() [*line 138*]

#### **Player ID.**

Player ID.

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

*int/bool* function OTS\_Player::getLastIP() [*line 834*]



**Last login IP.**  
Last login IP.

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

*int/bool* function OTS\_Player::getLastLogin() [*line 808*]

**Last login timestamp.**  
Last login timestamp.

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

*int/bool* function OTS\_Player::getLevel() [*line 314*]

**Experience level.**  
Experience level.

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

*int/bool* function OTS\_Player::getLookAddons() [*line 678*]

**Addons.**  
Addons.

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

*int/bool* function OTS\_Player::getLookBody() [*line 548*]

**Body color.**

Body color.

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

*int/bool* function OTS\_Player::getLookFeet() [*line 574*]

**Boots color.**

Boots color.

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

*int/bool* function OTS\_Player::getLookHead() [*line 600*]

**Hair color.**

Hair color.

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

*int/bool* function OTS\_Player::getLookLegs() [*line 626*]

**Legs color.**

Legs color.

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

*int/bool* function OTS\_Player::getLookType() [*line 652*]

**Outfit.**

Outfit.

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

*int/bool* function OTS\_Player::getLossExperience() [*line 1052*]

**Percentage of experience lost after dead.**

Percentage of experience lost after dead.

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

*int/bool* function OTS\_Player::getLossMana() [*line 1077*]

**Percentage of used mana lost after dead.**

Percentage of used mana lost after dead.

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

*int/bool* function OTS\_Player::getLossSkills() [*line 1102*]  
**Percentage of skills lost after dead.**  
 Percentage of skills lost after dead.

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

*int/bool* function OTS\_Player::getMagLevel() [*line 340*]  
**Magic level.**  
 Magic level.

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

*int/bool* function OTS\_Player::getMana() [*line 418*]  
**Current mana.**  
 Current mana.

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

*int/bool* function OTS\_Player::getManaMax() [*line 444*]

**Maximum mana.**

Maximum mana.

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

*int/bool* function OTS\_Player::getManaSpent() [*line 470*]

**Mana spent.**

Mana spent.

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

*string/bool* function OTS\_Player::getName() [*line 154*]

**Player name.**

Player name.

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

*int/bool* function OTS\_Player::getPosX() [*line 704*]

**X map coordinate.**

X map coordinate.

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

*int/bool* function OTS\_Player::getPosY() [*line 730*]

**Y map coordinate.**

Y map coordinate.

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

*int/bool* function OTS\_Player::getPosZ() [*line 756*]

**Z map coordinate.**

Z map coordinate.

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

*int/bool* function OTS\_Player::getRankId() [*line 1002*]

**Guild rank ID.**

Guild rank ID.

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

*int/bool* function OTS\_Player::getRedSkullTime() [*line 918*]

**Red skulled time remained.**

Red skulled time remained.

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

*int/bool* function OTS\_Player::getSex() [*line 236*]

**Player gender.**

Player gender.

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

*int* function OTS\_Player::getSkill(\$skill) [*line 1130*]

**Function Parameters:**

- *int* **\$skill** Skill ID.

**Returns player's skill.**

Returns player's skill.

- **Version** 0.0.1+SVN
- **Version** 0.0.1
- **Access** public

*int* function OTS\_Player::getSkillTries(\$skill) [*line 1154*]

**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** 0.0.1+SVN
- **Version** 0.0.1
- **Access** public

*int|bool* function OTS\_Player::getSoul() [*line 496*]

**Soul points.**

Soul points.

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

*int|bool* function OTS\_Player::getTownId() [*line 1027*]

**Residence town's ID.**

Residence town's ID.

- **Version** 0.0.1



- **Access** public

*int|bool* function OTS\_Player::getVocation() [*line 262*]

**Player proffesion.**

Player proffesion.

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

*bool|null* function OTS\_Player::hasRedSkull() [*line 944*]

**Checks if player has red skull.**

Checks if player has red skull.

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

*bool* function OTS\_Player::isLoading() [*line 99*]

**Checks if object is loaded.**

Checks if object is loaded.

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

*bool|null* function OTS\_Player::isSaveSet() [*line 860*]

### **Checks if save flag is set.**

Checks if save flag is set.

- **Version** 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** 0.0.1+SVN
- **Version** 0.0.1
- **Access** public

*void function OTS\_Player::save() [line 109]*

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

Saves account in database.

- **Version** 0.0.1+SVN
- **Version** 0.0.1
- **Access** public

*void function OTS\_Player::setAccount(\$account) [line 198]*

**Function Parameters:**

- [\*OTS Account\*](#) **\$account** Owning account.

**Assigns character to account.**

Assigns character to account.

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

*void function OTS\_Player::setCap(\$cap) [line 798]*

**Function Parameters:**

- *int* **\$cap** Capacity.

**Sets capacity.**

Sets capacity.

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

*void function OTS\_Player::setConditions(\$conditions) [line 908]*

**Function Parameters:**

- *mixed* **\$conditions** Condition binary field.

### **Sets conditions.**

Sets conditions.

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

*void function OTS\_Player::setDirection(\$direction) [line 538]*

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

- *int* **\$direction** Looking direction.

### **Sets looking direction.**

Sets looking direction.

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

*void function OTS\_Player::setExperience(\$experience) [line 304]*

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

- *int* **\$experience** Experience points.

### **Sets experience points.**

Sets experience points.

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

*void function OTS\_Player::setGroup(\$group) [line 226]*

***Function Parameters:***

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

**Assigns character to group.**

Assigns character to group.

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

*void function OTS\_Player::setGuildNick(\$guildnick) [line 992]*

***Function Parameters:***

- *string* **\$guildnick** Name.

**Sets guild nick.**

Sets guild nick.

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

*void function OTS\_Player::setHealth(\$health) [line 382]*

***Function Parameters:***

- ***int* \$health** Current HP.

### **Sets current HP.**

Sets current HP.

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

*void function OTS\_Player::setHealthMax(\$healthmax) [line 408]*

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

- ***int* \$healthmax** Maximum HP.

### **Sets maximum HP.**

Sets maximum HP.

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

*void function OTS\_Player::setLastIP(\$lastip) [line 850]*

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

- ***int* \$lastip** Last login IP.

### **Sets last login IP.**

Sets last login IP.

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

*void function OTS\_Player::setLastLogin(\$lastlogin) [line 824]*

***Function Parameters:***

- *int* **\$lastlogin** Last login timestamp.

**Sets last login timestamp.**

Sets last login timestamp.

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

*void function OTS\_Player::setLevel(\$level) [line 330]*

***Function Parameters:***

- *int* **\$level** Experience level.

**Sets experience level.**

Sets experience level.

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

*void function OTS\_Player::setLookAddons(\$lookaddons) [line 694]*

***Function Parameters:***

- *int* **\$lookaddons** Addons.

**Sets addons.**

Sets addons.

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

*void function OTS\_Player::setLookBody(\$lookbody) [line 564]*

***Function Parameters:***

- *int* **\$lookbody** Body color.

**Sets body color.**

Sets body color.

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

*void function OTS\_Player::setLookFeet(\$lookfeet) [line 590]*

***Function Parameters:***

- *int* **\$lookfeet** Boots color.



### **Sets boots color.**

Sets boots color.

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

*void function OTS\_Player::setLookHead(\$lookhead) [line 616]*

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

- *int* **\$lookhead** Hair color.

### **Sets hair color.**

Sets hair color.

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

*void function OTS\_Player::setLookLegs(\$looklegs) [line 642]*

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

- *int* **\$looklegs** Legs color.

### **Sets legs color.**

Sets legs color.

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

*void function OTS\_Player::setLookType(\$looktype) [line 668]*

***Function Parameters:***

- *int* **\$looktype** Outfit.

**Sets outfit.**

Sets outfit.

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

*void function OTS\_Player::setLossExperience(\$loss\_experience) [line 1068]*

***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
- **Access** public

*void function OTS\_Player::setLossMana(\$loss\_mana) [line 1093]*

***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
- **Access** public

*void* function OTS\_Player::setLossSkills(\$loss\_skills) [*line 1118*]

#### **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
- **Access** public

*void* function OTS\_Player::setMagLevel(\$maglevel) [*line 356*]

#### **Function Parameters:**

- *int* **\$maglevel** Magic level.

#### **Sets magic level.**

Sets magic level.

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

*void function OTS\_Player::setMana(\$mana) [line 434]*

***Function Parameters:***

- *int* **\$mana** Current mana.

**Sets current mana.**

Sets current mana.

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

*void function OTS\_Player::setManaMax(\$manamax) [line 460]*

***Function Parameters:***

- *int* **\$manamax** Maximum mana.

**Sets maximum mana.**

Sets maximum mana.

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

*void function OTS\_Player::setManaSpent(\$manaspent) [line 486]*

***Function Parameters:***

- *int* **\$manaspent** Mana spent.

**Sets mana spent.**

Sets mana spent.

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

*void function OTS\_Player::setName(\$name) [line 170]*

***Function Parameters:***

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

**Sets players's name.**

Sets players's name.

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

*void function OTS\_Player::setPosX(\$posx) [line 720]*

***Function Parameters:***

- *int* **\$posx** X map coordinate.

### **Sets X map coordinate.**

Sets X map coordinate.

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

*void function OTS\_Player::setPosY(\$posy) [line 746]*

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

- *int* **\$posy** Y map coordinate.

### **Sets Y map coordinate.**

Sets Y map coordinate.

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

*void function OTS\_Player::setPosZ(\$posz) [line 772]*

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

- *int* **\$posz** Z map coordinate.

### **Sets Z map coordinate.**

Sets Z map coordinate.

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

*void function OTS\_Player::setRankId(\$rank\_id) [line 1018]*

**Function Parameters:**

- *int* **\$rank\_id** Guild rank ID.

**Sets guild rank ID.**

Sets guild rank ID.

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

*void function OTS\_Player::setRedSkull() [line 966]*

**Sets red skull flag.**

Sets red skull flag.

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

*void function OTS\_Player::setRedSkullTime(\$redskulltime) [line 934]*

**Function Parameters:**

- *int* **\$redskulltime** Red skulled time remained.

### **Sets red skulled time remained.**

Sets red skulled time remained.

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

*void function OTS\_Player::setSave() [line 882]*

### **Sets save flag.**

Sets save flag.

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

*void function OTS\_Player::setSex(\$sex) [line 252]*

### **Function Parameters:**

- *int* **\$sex** Player gender.

### **Sets player gender.**

Sets player gender.

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

*void function OTS\_Player::setSkill(\$skill, \$value) [line 1142]*



**Function Parameters:**

- *int* **\$skill** Skill ID.
- *int* **\$value** Skill value.

**Sets skill value.**

Sets skill value.

- **Version** 0.0.1+SVN
- **Version** 0.0.1
- **Access** public

*void* function OTS\_Player::setSkillTries(\$skill, \$tries) [*line 1166*]

**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** 0.0.1+SVN
- **Version** 0.0.1
- **Access** public

*void* function OTS\_Player::setSoul(\$soul) [*line 512*]

**Function Parameters:**

- ***int* \$soul** Soul points.

### **Sets soul points.**

Sets soul points.

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

*void function OTS\_Player::setTownId(\$town\_id) [line 1043]*

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

- ***int* \$town\_id** Residence town's ID.

### **Sets residence town's ID.**

Sets residence town's ID.

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

*void function OTS\_Player::setVocation(\$vocation) [line 278]*

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

- ***int* \$vocation** Player proffesion.

### **Sets player proffesion.**

Sets player proffesion.

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

*void function OTS\_Player::unsetRedSkull() [line 958]*

**Unsets red skull flag.**  
Unsets red skull flag.

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

*void function OTS\_Player::unsetSave() [line 874]*

**Unsets save flag.**  
Unsets save flag.

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

## Class OTS\_Players\_List

*[line 19]*

**List of players.**  
List of players.

- **Package** POT
- **Version** 0.0.1

Constructor *void* function OTS\_Players\_List::\_\_construct(\$db) [*line 54*]

**Function Parameters:**

- [\*IOTS\\_DB\*](#) \$db Database connection object.

**Sets database connection handler.**

Sets database connection handler.

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

*int* function OTS\_Players\_List::count() [*line 166*]

**Returns number of characters on list in current criterium.**

Returns number of characters on list in current criterium.

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

OTS\_Player function OTS\_Players\_List::current() [*line 116*]

**Returns current row.**

Returns current row.

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

*bool* function OTS\_Players\_List::deletePlayer(\$player) [*line 99*]

**Function Parameters:**

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

**Deletes player.**

Deletes player.

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

*mixed* function OTS\_Players\_List::key() [*line 138*]

**Current cursor position.**

Current cursor position.

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

*void* function OTS\_Players\_List::next() [*line 128*]

**Moves to next row.**

Moves to next row.

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

*void function OTS\_Players\_List::rewind() [line 156]*

**Select players from database.**

Select players from database.

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

*void function OTS\_Players\_List::setLimit([\$limit = false]) [line 64]*

**Function Parameters:**

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

**Sets LIMIT.**

Sets LIMIT.

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

*void function OTS\_Players\_List::setOffset([\$offset = false]) [line 81]*

**Function Parameters:**

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

**Sets OFFSET.**  
Sets OFFSET.

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

*bool* function OTS\_Players\_List::valid() [*line 148*]

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

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

## Class POT

[*line 23*]

**Main POT class.**  
Main POT class.

- **Package** POT
- **Version** 0.0.1
- **Version** 0.0.1+SVN

**POT::DB\_MYSQL**

= 1 *[line 28]*

**MySQL driver.**

MySQL driver.

- **Version 0.0.1**

**POT::DB\_SQLITE**

= 2 *[line 32]*

**SQLite driver.**

SQLite driver.

- **Version 0.0.1**

**POT::DIRECTION\_EAST**

= 1 *[line 71]*

**East.**

East.

- **Version 0.0.1**

**POT::DIRECTION\_NORTH**



= 0 *[line 67]*

**North.**  
North.

- **Version 0.0.1**

**POT::DIRECTION\_SOUTH**

= 2 *[line 75]*

**South.**  
South.

- **Version 0.0.1**

**POT::DIRECTION\_WEST**

= 3 *[line 79]*

**West.**  
West.

- **Version 0.0.1**

**POT::SEX\_FEMALE**

= 0 *[line 37]*

**Female gender.**

Female gender.

- **Version 0.0.1**

**POT::SEX\_MALE**

= 1 [*line 41*]

**Male gender.**  
Male gender.

- **Version 0.0.1**

**POT::SKILL\_AXE**

= 3 [*line 104*]

**Axe fighting.**  
Axe fighting.

- **Version 0.0.1**
- **Version 0.0.1+SVN**

**POT::SKILL\_CLUB**

= 1 [*line 92*]

**Club fighting.**  
Club fighting.

- **Version** 0.0.1
- **Version** 0.0.1+SVN

#### **POT::SKILL\_DISTANCE**

= 4 [*line 110*]

**Distance fighting.**  
Distance fighting.

- **Version** 0.0.1
- **Version** 0.0.1+SVN

#### **POT::SKILL\_FISHING**

= 6 [*line 122*]

**Fishing.**  
Fishing.

- **Version** 0.0.1
- **Version** 0.0.1+SVN

#### **POT::SKILL\_FIST**

= 0 [*line 86*]

**Fist fighting.**  
Fist fighting.

- **Version** 0.0.1
- **Version** 0.0.1+SVN

## **POT::SKILL\_SHIELDING**

= 5 [*line 116*]

**Shielding.**  
Shielding.

- **Version** 0.0.1
- **Version** 0.0.1+SVN

## **POT::SKILL\_SWORD**

= 2 [*line 98*]

**Sword fighting.**  
Sword fighting.

- **Version** 0.0.1
- **Version** 0.0.1+SVN

## **POT::VOCATION\_DRUID**

= 2 [*line 54*]

**Druid.**

Druid.

- **Version 0.0.1**

**POT::VOCATION\_KNIGHT**

= 4 *[line 62]*

**Knight.**  
Knight.

- **Version 0.0.1**

**POT::VOCATION\_NONE**

= 0 *[line 46]*

**None vocation.**  
None vocation.

- **Version 0.0.1**

**POT::VOCATION\_PALADIN**

= 3 *[line 58]*

**Paladin.**  
Paladin.

- **Version 0.0.1**

**POT::VOCATION\_SORCERER**

= 1 *[line 50]*

**Sorcerer.**

Sorcerer.

- **Version 0.0.1**

Constructor *void* function POT::\_\_construct() *[line 179]*

**Class initialization tools.**

Class initialization tools.

Never create instance of this class by yourself! Use POT::getInstance()!

- **Version 0.0.1**
- **See** POT::getInstance();
- **Access** public

*void* function POT::connect(\$driver, \$params) *[line 236]*

**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 and SQLite drivers are supported. XML is not planned.

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** 0.0.1
- **Throws** Exception When driver is not supported.
- **Access** public
- **Example**

*IOTS\_DAO* function POT::createObject(\$class) [*line 281*]

### Function Parameters:

- *string* **\$class** Class name.

### Creates OTServ DAO class instance.

Creates OTServ DAO class instance.

Currently it means Account, or Player object.

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

POT function POT::getInstance() [line 129]

### Singleton.

Singleton.

- **Version** 0.0.1
- **Static**
- **Access** public

void function POT::loadClass(\$class) [line 194]

### autoload.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 function __autoload($class)
15 {
```



```

16     // checks if it's POT class
17     if( preg_match('/^I?OTS_/', $class) != 0)
18     {
19         POT::getInstance()-> loadClass($class);
20     }
21     /*
22     // possibly call your own __autoload() handler
23     else
24     {
25         here comes your stuff...
26     }
27     */
28 }
29
30 ?>

```

#### Function Parameters:

- *string* **\$class** Class name.

#### Loads POT class file.

Loads POT class file.

Runtime class loading on demand - usefull for `__autoload()` function.

- **Version** 0.0.1
- **Throws** Exception When give class is not POT toolkit class.
- **Access** public
- **Example**

`void` function POT::setPOTPath(\$path) [line 160]

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
- **Access** public
- **Example**

# Appendices

# Appendix A - Class Trees

## Package POT

### IOTS\_DAO

- [IOTS\\_DAO](#)

### IOTS\_DB

- [IOTS\\_DB](#)

### OTS\_Account

- [OTS\\_Account](#)

### OTS\_Accounts\_List

- [OTS\\_Accounts\\_List](#)

### OTS\_DB\_MySQL

- PDO
  - [OTS\\_DB\\_MySQL](#)

### OTS\_DB\_SQLite

- PDO
  - [OTS\\_DB\\_SQLite](#)

## OTS\_Group

- [OTS\\_Group](#)

## OTS\_Groups\_List

- [OTS\\_Groups\\_List](#)

## OTS\_Player

- [OTS\\_Player](#)

## OTS\_Players\_List

- [OTS\\_Players\\_List](#)

## POT

- [POT](#)

# Appendix B - README/CHANGELOG/INSTALL

# 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.

# CHANGELOG

## [SVN]

- \* Added skills support in OTS\_Player class. <wrzasq>
- \* HTML documentation removed from SVN (pointless to update it all the time, you can re-create it with phpdoc and make). <wrzasq>
- \* Fixed `redskulltime` field name in OTS\_Player. <wrzasq>
- \* Added find() to OTS\_Account class to load accounts by their's e-mail addresses. <wrzasq>
- \* Documentation fixes. <wrzasq>
- \* Additional info/example. <wrzasq>

## [0.0.1]

- \* Initial release. <wrzasq>

# NEWS

This is the very first release of this toolkit. Read README file for more info.

# 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 [documentation/index.html](#) or [documentation.pdf](#) files.

#### ==== 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.  
documentation/ - phpDocumentor-generator documentation.  
examples/ - example files for learning.  
tutorials/ - phpDocumentor directory.  
BUGS - known bugs.  
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.  
TODO - list of things to be done.  
Makefile - make input, for documentation generation.  
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, otserv-aac, test).  
clean: deletes documentation.  
check: checks syntax of all PHP files.  
documentation: generates HTML documentation.  
pdf: generates PDF documentation.  
otserv-aac: OTServ-AAC website documentation template used.  
test: runs test suite.

For more readable output of phpUnit test run:  
php test.php

#### ==== Credits ====

\* Wrzasq <[wrzasq@gmail.com](mailto:wrzasq@gmail.com)> - project initiator, main developer.



# Index

## A

[Account number hack](#) . . . . . 9

## C

[constructor OTS\\_Player::\\_\\_construct\(\)](#) . . . . . 55  
    *Sets database connection handler.*  
[constructor OTS\\_Groups\\_List::\\_\\_construct\(\)](#) . . . . . 51  
    *Sets database connection handler.*  
[constructor OTS\\_Players\\_List::\\_\\_construct\(\)](#) . . . . . 86  
    *Sets database connection handler.*  
[constructor POT::\\_\\_construct\(\)](#) . . . . . 96  
    *Class initialization tools.*  
[CHANGELOG](#) . . . . . 105  
[constructor OTS\\_Group::\\_\\_construct\(\)](#) . . . . . 45  
    *Sets database connection handler.*  
[constructor OTS\\_DB\\_SQLite::\\_\\_construct\(\)](#) . . . . . 41  
    *Creates database connection.*  
[constructor IOTS\\_DB::\\_\\_construct\(\)](#) . . . . . 24  
    *Connection parameters.*  
[constructor OTS\\_Account::\\_\\_construct\(\)](#) . . . . . 27  
    *Sets database connection handler.*  
[constructor OTS\\_Accounts\\_List::\\_\\_construct\(\)](#) . . . . . 34  
    *Sets database connection handler.*  
[constructor OTS\\_DB\\_MySQL::\\_\\_construct\(\)](#) . . . . . 38  
    *Creates database connection.*  
[constructor IOTS\\_DAO::\\_\\_construct\(\)](#) . . . . . 23  
    *DAO objects must be initialized with a database.*

## I

[IOTS\\_DB::SQLquery\(\)](#) . . . . . 26  
    *Evaluates query.*  
[IOTS\\_DB::SQLquote\(\)](#) . . . . . 26  
    *Query-quoted string value.*  
[IOTS\\_DB::tableName\(\)](#) . . . . . 26  
    *Query-quoted table name.*  
[INSTALL](#) . . . . . 105  
[IOTS\\_DB::limit\(\)](#) . . . . . 25  
    *LIMIT/OFFSET clause for queries.*  
[IOTS\\_DB::lastInsertId\(\)](#) . . . . . 25  
    *ID of last created record.*  
[IOTS\\_DB.php](#) . . . . . 12

<a href="#">IOTS_DAO</a>	OTserv database object.	23
<a href="#">IOTS_DB</a>	OTServ database handler interface.	24
<a href="#">IOTS_DB::fieldName()</a>	Query-quoted field name.	24
<a href="#">IOTS_DAO.php</a>		11

## N

<a href="#">NEWS</a>		105
----------------------	--	-----

## O

<a href="#">OTS_Player::getTownId()</a>	Residence town's ID.	66
<a href="#">OTS_Player::getVocation()</a>	Player profession.	67
<a href="#">OTS_Player::hasRedSkull()</a>	Checks if player has red skull.	67
<a href="#">OTS_Player::isLoading()</a>	Checks if object is loaded.	67
<a href="#">OTS_Player::getSoul()</a>	Soul points.	66
<a href="#">OTS_Player::getSkillTries()</a>	Returns player's skill's tries for next level.	66
<a href="#">OTS_Player::getRedSkullTime()</a>	Red skulled time remained.	65
<a href="#">OTS_Player::getSex()</a>	Player gender.	65
<a href="#">OTS_Player::getSkill()</a>	Returns player's skill.	65
<a href="#">OTS_Player::isSaveSet()</a>	Checks if save flag is set.	67
<a href="#">OTS_Player::load()</a>	Loads player with given id.	68
<a href="#">OTS_Player::setExperience()</a>	Sets experience points.	70
<a href="#">OTS_Player::setGroup()</a>	Assigns character to group.	71
<a href="#">OTS_Player::setGuildNick()</a>	Sets guild nick.	71
<a href="#">OTS_Player::setHealth()</a>	Sets current HP.	71
<a href="#">OTS_Player::setDirection()</a>	Sets looking direction.	70
<a href="#">OTS_Player::setConditions()</a>	Sets conditions.	69
<a href="#">OTS_Player::save()</a>	Saves account in database.	68
<a href="#">OTS_Player::setAccount()</a>		69

<i>Assigns character to account.</i>	
<a href="#">OTS_Player::setCap()</a>	69
<i>Sets capacity.</i>	
<a href="#">OTS_Player::getRankId()</a>	64
<i>Guild rank ID.</i>	
<a href="#">OTS_Player::getPosZ()</a>	64
<i>Z map coordinate.</i>	
<a href="#">OTS_Player::getLookBody()</a>	60
<i>Body color.</i>	
<a href="#">OTS_Player::getLookFeet()</a>	60
<i>Boots color.</i>	
<a href="#">OTS_Player::getLookHead()</a>	60
<i>Hair color.</i>	
<a href="#">OTS_Player::getLookLegs()</a>	61
<i>Legs color.</i>	
<a href="#">OTS_Player::getLookAddons()</a>	59
<i>Addons.</i>	
<a href="#">OTS_Player::getLevel()</a>	59
<i>Experience level.</i>	
<a href="#">OTS_Player::getId()</a>	58
<i>Player ID.</i>	
<a href="#">OTS_Player::getLastIP()</a>	58
<i>Last login IP.</i>	
<a href="#">OTS_Player::getLastLogin()</a>	59
<i>Last login timestamp.</i>	
<a href="#">OTS_Player::getLookType()</a>	61
<i>Outfit.</i>	
<a href="#">OTS_Player::getLossExperience()</a>	61
<i>Percentage of experience lost after dead.</i>	
<a href="#">OTS_Player::getManaSpent()</a>	63
<i>Mana spent.</i>	
<a href="#">OTS_Player::getName()</a>	63
<i>Player name.</i>	
<a href="#">OTS_Player::getPosX()</a>	63
<i>X map coordinate.</i>	
<a href="#">OTS_Player::getPosY()</a>	64
<i>Y map coordinate.</i>	
<a href="#">OTS_Player::getManaMax()</a>	63
<i>Maximum mana.</i>	
<a href="#">OTS_Player::getMana()</a>	62
<i>Current mana.</i>	
<a href="#">OTS_Player::getLossMana()</a>	61
<i>Percentage of used mana lost after dead.</i>	
<a href="#">OTS_Player::getLossSkills()</a>	62
<i>Percentage of skills lost after dead.</i>	
<a href="#">OTS_Player::getMagLevel()</a>	62
<i>Magic level.</i>	
<a href="#">OTS_Player::setHealthMax()</a>	72
<i>Sets maximum HP.</i>	
<a href="#">OTS_Player::setLastIP()</a>	72
<i>Sets last login IP.</i>	
<a href="#">OTS_Player::setSoul()</a>	83
<i>Sets soul points.</i>	

<a href="#">OTS_Player::setTownId()</a>	84
<i>Sets residence town's ID.</i>	
<a href="#">OTS_Player::setVocation()</a>	84
<i>Sets player proffesion.</i>	
<a href="#">OTS_Player::unsetRedSkull()</a>	85
<i>Unsets red skull flag.</i>	
<a href="#">OTS_Player::setSkillTries()</a>	83
<i>Sets skill's tries for next level.</i>	
<a href="#">OTS_Player::setSkill()</a>	82
<i>Sets skill value.</i>	
<a href="#">OTS_Player::setRedSkullTime()</a>	81
<i>Sets red skulled time remained.</i>	
<a href="#">OTS_Player::setSave()</a>	82
<i>Sets save flag.</i>	
<a href="#">OTS_Player::setSex()</a>	82
<i>Sets player gender.</i>	
<a href="#">OTS_Player::unsetSave()</a>	85
<i>Unsets save flag.</i>	
<a href="#">OTS_Players_List</a>	85
<i>List of players.</i>	
<a href="#">OTS_Players_List::rewind()</a>	88
<i>Select players from database.</i>	
<a href="#">OTS_Players_List::setLimit()</a>	88
<i>Sets LIMIT.</i>	
<a href="#">OTS_Players_List::setOffset()</a>	88
<i>Sets OFFSET.</i>	
<a href="#">OTS_Players_List::valid()</a>	89
<i>Checks if there are any rows left.</i>	
<a href="#">OTS_Players_List::next()</a>	87
<i>Moves to next row.</i>	
<a href="#">OTS_Players_List::key()</a>	87
<i>Current cursor position.</i>	
<a href="#">OTS_Players_List::count()</a>	86
<i>Returns number of characters on list in current criterium.</i>	
<a href="#">OTS_Players_List::current()</a>	86
<i>Returns current row.</i>	
<a href="#">OTS_Players_List::deletePlayer()</a>	87
<i>Deletes player.</i>	
<a href="#">OTS_Player::setRedSkull()</a>	81
<i>Sets red skull flag.</i>	
<a href="#">OTS_Player::setRankId()</a>	81
<i>Sets guild rank ID.</i>	
<a href="#">OTS_Player::setLookHead()</a>	75
<i>Sets hair color.</i>	
<a href="#">OTS_Player::setLookLegs()</a>	75
<i>Sets legs color.</i>	
<a href="#">OTS_Player::setLookType()</a>	76
<i>Sets outfit.</i>	
<a href="#">OTS_Player::setLossExperience()</a>	76
<i>Sets percentage of experience lost after dead.</i>	
<a href="#">OTS_Player::setLookFeet()</a>	74
<i>Sets boots color.</i>	
<a href="#">OTS_Player::setLookBody()</a>	74

<i>Sets body color.</i>	
<a href="#">OTS_Player::setLastLogin()</a>	73
<i>Sets last login timestamp.</i>	
<a href="#">OTS_Player::setLevel()</a>	73
<i>Sets experience level.</i>	
<a href="#">OTS_Player::setLookAddons()</a>	74
<i>Sets addons.</i>	
<a href="#">OTS_Player::setLossMana()</a>	76
<i>Sets percentage of used mana lost after dead.</i>	
<a href="#">OTS_Player::setLossSkills()</a>	77
<i>Sets percentage of skills lost after dead.</i>	
<a href="#">OTS_Player::setPosX()</a>	79
<i>Sets X map coordinate.</i>	
<a href="#">OTS_Player::setPosY()</a>	80
<i>Sets Y map coordinate.</i>	
<a href="#">OTS_Player::setPosZ()</a>	80
<i>Sets Z map coordinate.</i>	
<a href="#">OTS_Player::setName()</a>	79
<i>Sets players's name.</i>	
<a href="#">OTS_Player::setManaSpent()</a>	79
<i>Sets mana spent.</i>	
<a href="#">OTS_Player::setMagLevel()</a>	77
<i>Sets magic level.</i>	
<a href="#">OTS_Player::setMana()</a>	78
<i>Sets current mana.</i>	
<a href="#">OTS_Player::setManaMax()</a>	78
<i>Sets maximum mana.</i>	
<a href="#">OTS_Player::getHealthMax()</a>	58
<i>Maximum HP.</i>	
<a href="#">OTS_Player::getHealth()</a>	58
<i>Current HP.</i>	
<a href="#">OTS_Accounts_List</a>	34
<i>List of accounts.</i>	
<a href="#">OTS_Accounts_List::count()</a>	35
<i>Returns number of accounts on list in current criterium.</i>	
<a href="#">OTS_Accounts_List::current()</a>	35
<i>Returns current row.</i>	
<a href="#">OTS_Accounts_List::deleteAccount()</a>	35
<i>Deletes account.</i>	
<a href="#">OTS_Account::unblock()</a>	34
<i>Unblocks account.</i>	
<a href="#">OTS_Account::setPassword()</a>	33
<i>Sets account's password.</i>	
<a href="#">OTS_Account::save()</a>	32
<i>Updates account in database.</i>	
<a href="#">OTS_Account::setEMail()</a>	32
<i>Sets account's email.</i>	
<a href="#">OTS_Account::setPACCDays()</a>	33
<i>Sets PACC days count.</i>	
<a href="#">OTS_Accounts_List::key()</a>	36
<i>Current cursor position.</i>	
<a href="#">OTS_Accounts_List::next()</a>	36
<i>Moves to next row.</i>	

<a href="#">OTS_DB_MySQL::fieldName()</a>	39
<i>Query-quoted field name.</i>	
<a href="#">OTS_DB_MySQL::limit()</a>	39
<i>LIMIT/OFFSET clause for queries.</i>	
<a href="#">OTS_DB_MySQL::SQLquery()</a>	40
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB_MySQL::SQLquote()</a>	40
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB_MySQL</a>	38
<i>MySQL connection interface.</i>	
<a href="#">OTS_Accounts_List::valid()</a>	37
<i>Checks if there are any rows left.</i>	
<a href="#">OTS_Accounts_List::rewind()</a>	36
<i>Select accounts from database.</i>	
<a href="#">OTS_Accounts_List::setLimit()</a>	37
<i>Sets LIMIT.</i>	
<a href="#">OTS_Accounts_List::setOffset()</a>	37
<i>Sets OFFSET.</i>	
<a href="#">OTS_Account::load()</a>	32
<i>Loads account with given number.</i>	
<a href="#">OTS_Account::isLoading()</a>	31
<i>Checks if object is loaded.</i>	
<a href="#">OTS_Groups_List.php</a>	19
<a href="#">OTS_Player.php</a>	20
<a href="#">OTS_Players_List.php</a>	21
<a href="#">OTS_SQLite_Results.php</a>	22
<a href="#">OTS_Group.php</a>	18
<a href="#">OTS_DB_SQLite.php</a>	17
<a href="#">OTS_Account.php</a>	14
<a href="#">OTS_Accounts_List.php</a>	15
<a href="#">OTS_DB_MySQL.php</a>	16
<a href="#">OTS_Account</a>	27
<i>OTServ account abstraction.</i>	
<a href="#">OTS_Account::block()</a>	28
<i>Blocks account.</i>	
<a href="#">OTS_Account::getPassword()</a>	30
<i>Account's password.</i>	
<a href="#">OTS_Account::getPlayers()</a>	31
<i>List of characters on account.</i>	
<a href="#">OTS_Account::isBlocked()</a>	31
<i>Checks if account is blocked.</i>	
<a href="#">OTS_Account::getPACCDays()</a>	30
<i>PACC days.</i>	
<a href="#">OTS_Account::getId()</a>	30
<i>Account number.</i>	
<a href="#">OTS_Account::create()</a>	28
<i>Creates new account.</i>	
<a href="#">OTS_Account::find()</a>	29
<i>Loads account by it's e-mail address.</i>	
<a href="#">OTS_Account::getEmail()</a>	30
<i>E-mail address.</i>	
<a href="#">OTS_DB_MySQL::tableName()</a>	40
<i>Query-quoted table name.</i>	

<a href="#">OTS_DB_SQLite</a>	41
<i>SQLite connection interface.</i>	
<a href="#">OTS_Groups_List::key()</a>	52
<i>Current cursor position.</i>	
<a href="#">OTS_Groups_List::next()</a>	53
<i>Moves to next row.</i>	
<a href="#">OTS_Groups_List::rewind()</a>	53
<i>Select groups from database.</i>	
<a href="#">OTS_Groups_List::setLimit()</a>	53
<i>Sets LIMIT.</i>	
<a href="#">OTS_Groups_List::deleteGroup()</a>	52
<i>Deletes group.</i>	
<a href="#">OTS_Groups_List::current()</a>	52
<i>Returns current row.</i>	
<a href="#">OTS_Group::setName()</a>	50
<i>Sets group's name.</i>	
<a href="#">OTS_Groups_List</a>	51
<i>List of groups.</i>	
<a href="#">OTS_Groups_List::count()</a>	51
<i>Returns number of groups on list in current criterium.</i>	
<a href="#">OTS_Groups_List::setOffset()</a>	54
<i>Sets OFFSET.</i>	
<a href="#">OTS_Groups_List::valid()</a>	54
<i>Checks if there are any rows left.</i>	
<a href="#">OTS_Player::getDirection()</a>	56
<i>Looking direction.</i>	
<a href="#">OTS_Player::getExperience()</a>	57
<i>Experience points.</i>	
<a href="#">OTS_Player::getGroup()</a>	57
<i>Returns group of this player.</i>	
<a href="#">OTS_Player::getGuildNick()</a>	57
<i>Guild nick.</i>	
<a href="#">OTS_Player::getConditions()</a>	56
<i>Conditions.</i>	
<a href="#">OTS_Player::getCap()</a>	56
<i>Capacity.</i>	
<a href="#">OTS_Player</a>	54
<i>OTServ character abstraction.</i>	
<a href="#">OTS_Player::find()</a>	55
<i>Loads player by it's name.</i>	
<a href="#">OTS_Player::getAccount()</a>	56
<i>Returns account of this player.</i>	
<a href="#">OTS_Group::setMaxVIPList()</a>	50
<i>Sets maximum count of players in VIP list.</i>	
<a href="#">OTS_Group::setMaxDepotItems()</a>	49
<i>Sets maximum count of items in depot.</i>	
<a href="#">OTS_DB_SQLite::tableName()</a>	44
<i>Query-quoted table name.</i>	
<a href="#">OTS_Group</a>	45
<i>OTServ user group abstraction.</i>	
<a href="#">OTS_Group::getAccess()</a>	45
<i>Access level.</i>	
<a href="#">OTS_Group::getFlags()</a>	46

<i>Rights flags.</i>	
<a href="#">OTS_DB SQLite::SQLquote()</a>	44
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB SQLite::SQLquery()</a>	43
<i>IOTS_DB method.</i>	
<a href="#">OTS_DB SQLite::fieldName()</a>	42
<i>Query-quoted field name.</i>	
<a href="#">OTS_DB SQLite::limit()</a>	42
<i>LIMIT/OFFSET clause for queries.</i>	
<a href="#">OTS_DB SQLite::regexp()</a>	43
<i>REGEXP operator for SQLite</i>	
<a href="#">OTS_Group::getId()</a>	46
<i>Group ID.</i>	
<a href="#">OTS_Group::getMaxDepotItems()</a>	46
<i>Maximum count of items in depot.</i>	
<a href="#">OTS_Group::save()</a>	48
<i>Saves account in database.</i>	
<a href="#">OTS_Group::setAccess()</a>	48
<i>Sets access level.</i>	
<a href="#">OTS_Group::setFlags()</a>	49
<i>Sets rights flags.</i>	
<a href="#">OTS_Group::load()</a>	48
<i>Loads group with given id.</i>	
<a href="#">OTS_Group::isLoading()</a>	47
<i>Checks if object is loaded.</i>	
<a href="#">OTS_Group::getMaxVIPList()</a>	46
<i>Maximum count of players in VIP list.</i>	
<a href="#">OTS_Group::getName()</a>	47
<i>Group name.</i>	
<a href="#">OTS_Group::getPlayers()</a>	47
<i>List of characters in given group.</i>	
<a href="#">OTS.php</a>	13
<i>This file contains main toolkit class.</i>	

## P

<a href="#">POT::VOCATION_KNIGHT</a>	95
<i>Knight.</i>	
<a href="#">POT::VOCATION_NONE</a>	95
<i>None vocation.</i>	
<a href="#">POT::VOCATION_DRUID</a>	94
<i>Druid.</i>	
<a href="#">POT::SKILL_SWORD</a>	94
<i>Sword fighting.</i>	
<a href="#">POT::SKILL_FIST</a>	93
<i>Fist fighting.</i>	
<a href="#">POT::SKILL_SHIELDING</a>	94
<i>Shielding.</i>	
<a href="#">POT::VOCATION_PALADIN</a>	95
<i>Paladin.</i>	
<a href="#">POT::VOCATION_SORCERER</a>	96
<i>Sorcerer.</i>	



<a href="#">POT::loadClass()</a>	98
<i>Loads POT class file.</i>	
<a href="#">POT::setPOTPath()</a>	99
<i>Set POT directory.</i>	
<a href="#">POT::getInstance()</a>	98
<i>Singleton.</i>	
<a href="#">POT::createObject()</a>	97
<i>Creates OTServ DAO class instance.</i>	
<a href="#">POT::connect()</a>	96
<i>Connects to database.</i>	
<a href="#">POT::SKILL_FISHING</a>	93
<i>Fishing.</i>	
<a href="#">POT::SKILL_DISTANCE</a>	93
<i>Distance fighting.</i>	
<a href="#">POT::DB_SQLITE</a>	90
<i>SQLite driver.</i>	
<a href="#">POT::DIRECTION_EAST</a>	90
<i>East.</i>	
<a href="#">POT::DB_MYSQL</a>	90
<i>MySQL driver.</i>	
<a href="#">POT</a>	89
<i>Main POT class.</i>	
<a href="#">POT class preview</a>	3
<a href="#">POT::DIRECTION_NORTH</a>	91
<i>North.</i>	
<a href="#">POT::DIRECTION_SOUTH</a>	91
<i>South.</i>	
<a href="#">POT::SKILL_AXE</a>	92
<i>Axe fighting.</i>	
<a href="#">POT::SKILL_CLUB</a>	92
<i>Club fighting.</i>	
<a href="#">POT::SEX_MALE</a>	92
<i>Male gender.</i>	
<a href="#">POT::SEX_FEMALE</a>	91
<i>Female gender.</i>	
<a href="#">POT::DIRECTION_WEST</a>	91
<i>West.</i>	
<a href="#">POT</a>	1

## Q

<a href="#">Quick start</a>	5
-----------------------------	---

## R

<a href="#">README</a>	105
------------------------	-----