

Generated Documentation



Contents

Package default Classes	1
Class AbstractModel	1
Var \$ table	1
Constructor construct	1
Method alterar	2
Method deletar	2
Method descreverTabela	3
Method inserir	3
Method recuperaPorParametro	3
Method recuperaTodos	4
Class acaoModel	4
Var \$ table	4
Constructor construct	5
Class anexoModel	5
Var \$ table	5
Constructor construct	5
Class AuthModel	6
Constructor construct	6
Method getInfo	6
Method logar	6
Method logout	7
Method verificaLogin	7
Class grupoModel	8
Var \$ table	8
Constructor construct	8
Class grupoTipoModel	8
Var \$ table	9
Constructor construct	9
Class grupoTipoVinculoModel	9
Var \$ table	9
Constructor construct	9
Class grupoVinculoModel	10
Var \$ table	10
Constructor construct	10
Class home	10
Constructor construct	11
Method index	11
Class logModel	11
Var \$ table	11
Constructor construct	12
Class perfilModel	12
Var \$ table	12
Constructor construct	12

Class permissaoModel	12
Var \$ table	13
Constructor __construct	13
Class postModel	13
Var \$ table	13
Constructor __construct	13
Class projetoModel	14
Var \$ table	14
Constructor __construct	14
Class serviceauth	14
Constructor __construct	15
Method index	15
Method login	15
Method logout	16
Class tarefaModel	16
Var \$ table	16
Constructor __construct	16
Class tituloModel	17
Var \$ table	17
Constructor __construct	17
Class usuario	17
Constructor __construct	17
Method inserir	18
Class usuarioModel	18
Var \$ table	18
Constructor __construct	18
Class usuarioVinculoModel	18
Var \$ table	19
Constructor __construct	19
Class usuarioTipoVinculoModel	19
Var \$ table	19
Constructor __construct	19
Class viewGrupoTipoModel	20
Var \$ table	20
Constructor __construct	20
Class viewPerfilAcaoModel	20
Var \$ table	21
Constructor __construct	21
Method gerarPermissoes	21
Method verificaPermissao	21
Appendices	23
Appendix A - Class Trees	24
default	24

Package default Classes

Class AbstractModel [line 3]

- **Package** default

AbstractModel::\$_table

String = "" [line 9]

- **Var** variável abstrata que será sobrescrita nas classes filhas, representando a tabela na qual a model vai atuar.
- **Access** protected

Constructor *NULL* function AbstractModel::__construct(0) [line 17]

Function Parameters:

- *NULL* 0

Carrega todos os métodos contidos na classe pai.

- **Author** Paulo Viegas < pauloviegas93@gmail.com >
- **Access** public

Boolean function AbstractModel::alterar(\$dados, [\$parametroAlteracao = NULL]) [line 75]

Function Parameters:

- Array **\$dados** contem as informações a serem substitutas.
- Array **\$parametroAlteracao** A chave corresponde a coluna da tabela no banco e o valor da chave corresponde ao valor do filtro a ser usado na busca.

Altera as informações da tabela setada na classe filha, através do id, caso não seja informado o segundo parâmetro, ou através do segundo parâmetro, caso este seja populado.

OBS: No Array \$dados deve estar contido o id a ser alterado, caso se deseje fazer a alteração pelo mesmo.

- **Author** Paulo Viegas < pauloviegas93@gmail.com>
- **Access** public

Boolean function AbstractModel::deletar([\$id = NULL], [\$parametros = NULL]) [line 115]

Function Parameters:

- Int **\$id** corresponde ao id linha a ser excluida;
- Array **\$parametros** A chave corresponde a coluna da tabela no banco e o valor da chave corresponde ao valor do filtro a ser usado na exclusao.

Exclui logicamente (alterando para 0 o campo excluido do registro), ou fisicamente, um ou mais registros da tabela setada na classe filha, a partir de id, caso ele exista, ou a partir do segundo parâmetro, caso o id seja NULL e segundo parâmetro seja populado.

OBS 1: Se houver mais de um registro que correspondam aos valores dos parâmetros de busca passados, todos serão excluidos.

OBS 2: Se os dois parâmetros forem setados os dois serão parâmetros para busca.

- **Author** Paulo Viegas < pauloviegas93@gmail.com>

- **Access** public

Array function AbstractModel::descreverTabela(0) [line 29]

Function Parameters:

- *NULL* 0

Busca no banco todas as colunas da tabela.

- **Author** Paulo Viegas < pauloviegas93@gmail.com>
- **Access** public

Int function AbstractModel::inserir(\$dados) [line 51]

Function Parameters:

- *Array \$dados* contem as informações a serem inseridas;

Insere um registro na tabela

- **Author** Paulo Viegas < pauloviegas93@gmail.com>
- **Access** public

Array function AbstractModel::recuperaPorParametro([\$id = NULL], [\$parametrosBusca = NULL]) [line 165]

Function Parameters:

- *Int \$id* corresponde ao id linha a ser recuperada;
- *Array \$parametrosBusca* a chave corresponde a coluna de busca e o valor corresponde ao valor usado como filtro de busca;

Recupera todas as linhas (da tabela setada na classe filha) por id, caso exista, ou a

partir do segundo parâmetro, caso o id seja NULL e segundo parâmetro seja populado.

OBS: Caso sejam setados os dois parâmetros o id que será o parâmetro para busca.

- **Author** Paulo Viegas < pauloviegas93@gmail.com>
- **Access** public

Array function AbstractModel::recuperaTodos(0) [line 140]

Function Parameters:

- *NULL* 0

Recuperas todas as linhas (da tabela setada na classe filha) que não estão logicamente excluídas.

- **Author** Paulo Viegas < pauloviegas93@gmail.com>
- **Access** public

Class acaoModel

[line 3]

- **Package** default

acaoModel::\$_table

mixed = 'sitelinc_usu_acao' [line 6]

- **Access** protected

Constructor *void* function `acaoModel::__construct()` *[line 8]*

- **Access** public

Class `anexoModel` *[line 3]*

- **Package** default

`anexoModel::$_table`

mixed = `'sitelinc_gen_anexo'` *[line 6]*

- **Access** protected

Constructor *void* function `anexoModel::__construct()` *[line 8]*

- **Access** public

Class AuthModel

[line 3]

- **Package** default

Constructor *NULL* function AuthModel::__construct(0) [line 12]

Function Parameters:

- *NULL* 0

Carrega todos os métodos contidos na classe pai, bem como as Models (usuarioModel e ViewPerfiAcaoModel), necessárias .

- **Author** Paulo Viegas < pauloviegas93@gmail.com >
- **Access** public

void function AuthModel::getInfo(\$item) [line 106]

Function Parameters:

- **\$item**

- **Access** public

Boolean function AuthModel::logar(\$email, \$senha) [line 29]

Function Parameters:

- *String* **\$email** corresponde ao email fornecido pelo usuario;
- *String* **\$senha** corresponde a senha fornecida pelo usuario;

Realiza a autenticação do usuário no sistema, seguindo as seguintes regras:

- 1) A senha e o e-mail estejam corretos;**
- 2) O usuário não esteja logicamente excluído;**
- 3) O cadastro de usuário já tenha sido aprovado.**

São criadas duas sessions, que serão destruídas quando o usuário fizer logout do sistema:

- 1 - usuario: Corresponde a todos os dados do usuário.**
- 2 - permissões: Corresponde a todas as permissões dadas ao perfil deste usuário.**

- **Author** Paulo Viegas < pauloviegas93@gmail.com >
- **Access** public

Boolean function AuthModel::logout(0) [line 74]

Function Parameters:

- *NULL 0*

Realiza o logout do usuário! Destruído as sessões de usuário e permissões.

- **Author** Paulo Viegas < pauloviegas93@gmail.com >
- **Access** public

Boolean function AuthModel::verificaLogin(0) [line 94]

Function Parameters:

- *NULL 0*

Verifica se o usuário está logado, através da session usuário populada.

- **Author** Paulo Viegas < pauloviegas93@gmail.com>
- **Access** public

Class grupoModel

[line 3]

- **Package** default

grupoModel::\$_table

mixed = 'sitelinc_gru_grupo' [line 6]

- **Access** protected

Constructor *void* function grupoModel::__construct() [line 8]

- **Access** public

Class grupoTipoModel

[line 3]

- **Package** default

grupoTipoModel::\$_table

mixed = 'sitelinc_gru_tipo' [line 6]

- **Access** protected

Constructor *void* function grupoTipoModel::__construct() [line 8]

- **Access** public

Class grupoTipoVinculoModel

[line 3]

- **Package** default

grupoTipoVinculoModel::\$_table

mixed = 'sitelinc_gru_tipo_vinculo' [line 6]

- **Access** protected

Constructor *void* function grupoTipoVinculoModel::__construct() [line 8]

- **Access** public

Class grupoVinculoModel

[line 3]

- **Package** default

grupoVinculoModel::\$_table

mixed = 'sitelinc_gru_vinculo' [line 6]

- **Access** protected

Constructor *void* function grupoVinculoModel::__construct() [line 8]

- **Access** public

Class home

[line 3]

- **Package** default

Constructor *NULL* function home::__construct(0) [line 12]

Function Parameters:

- *NULL* 0

Carrega todos os mÃ©todos contidos na classe pai, necessÃ¡rias .

- **Author** Paulo Viegas < pauloviegas93@gmail.com>
- **Access** public

void function home::index() [line 18]

- **Access** public

Class logModel

[line 3]

- **Package** default

logModel::\$_table

mixed = 'sitelinc_log_log' [line 6]

- **Access** protected

Constructor *void* function logModel::__construct() *[line 8]*

- **Access** public

Class perfilModel

[line 3]

- **Package** default

perfilModel::\$_table

mixed = 'sitelinc_usu_perfil' *[line 6]*

- **Access** protected

Constructor *void* function perfilModel::__construct() *[line 8]*

- **Access** public

Class permissaoModel

[line 3]

- **Package** default

permissaoModel::\$_table

mixed = 'sitelinc_usu_permissao' [*line 6*]

- **Access** protected

Constructor *void* function permissaoModel::__construct() [*line 8*]

- **Access** public

Class postModel

[*line 3*]

- **Package** default

postModel::\$_table

mixed = 'sitelinc_gen_post' [*line 6*]

- **Access** protected

Constructor *void* function postModel::__construct() [*line 8*]

- **Access** public

Class projetoModel

[line 3]

- **Package** default

projetoModel::\$_table

mixed = 'sitelinc_pro_projeto' [line 6]

- **Access** protected

Constructor *void* function projetoModel::__construct() [line 8]

- **Access** public

Class serviceauth

[line 3]

- **Package** default

Constructor *NULL* function serviceauth::__construct(0) [line 12]

Function Parameters:

- *NULL* 0

Carrega todos os mÃ©todos contidos na classe pai, bem como as Models (tituloModel, grupoModel e viewGrupoTipoModel), necessÃ¡rias .

- **Author** Paulo Viegas < pauloviegas93@gmail.com>
- **Access** public

redirecionado function serviceauth::index(0) [line 27]

Function Parameters:

- *NULL*; 0

Carrega a pÃ¡gina inicial do sistema, onde o usuÃ¡rio poderÃ¡ autenticar-se ou, caso nÃ£o faÃ§a parte da rede social ainda, realize o seu cadastro.

- **Access** public

NULL; function serviceauth::logar(0) [line 50]

Function Parameters:

- *NULL*; 0

Realiza a requisitÃ£o ao Model authModel o login do usuÃ¡rio, caso valido
redireciona par o Feed de notÃcias do usuÃ¡rio, se nÃ£o para a pagina de login da rede social! Validando, anteriormente, os campos fornecidos!

- **Access** public

void function serviceauth::logout() [*line 69*]

- **Access** public

Class tarefaModel

[*line 3*]

- **Package** default

tarefaModel::\$_table

mixed = "sitelinc_pro_tarefa" [*line 6*]

- **Access** protected

Constructor *void* function tarefaModel::__construct() [*line 9*]

- **Access** public

Class tituloModel

[line 3]

- **Package** default

tituloModel::\$_table

mixed = 'sitelinc_usu_titulo' [line 6]

- **Access** protected

Constructor *void* function tituloModel::__construct() [line 8]

- **Access** public

Class usuario

[line 3]

- **Package** default

Constructor *void* function usuario::__construct() [line 6]

- **Access** public

void function usuario::inserir() [*line 12*]

- **Access** public

Class usuarioModel

[*line 3*]

- **Package** default

usuarioModel::\$_table

mixed = 'sitelinc_usu_usuario' [*line 6*]

- **Access** protected

Constructor *void* function usuarioModel::__construct() [*line 8*]

- **Access** public

Class usuarioVinculoModel

[*line 3*]

- **Package** default

usuarioVinculoModel::\$_table

mixed = 'sitelinc_usu_vinculo' [*line 6*]

- **Access** protected

Constructor *void* function usuarioVinculoModel::__construct() [*line 8*]

- **Access** public

Class usuarrioTipoVinculoModel

[*line 3*]

- **Package** default

usuarrioTipoVinculoModel::\$_table

mixed = 'sitelinc_usu_tipo_vinculo' [*line 6*]

- **Access** protected

Constructor *void* function usuarrioTipoVinculoModel::__construct() [*line 8*]

- **Access** public

Class viewGrupoTipoModel

[line 3]

- **Package** default

viewGrupoTipoModel::\$_table

mixed = 'sitelinc_view_grupo_tipo' [line 6]

- **Access** protected

Constructor *void* function viewGrupoTipoModel::__construct() [line 8]

- **Access** public

Class viewPerfilAcaoModel

[line 3]

- **Package** default

viewPerfilAcaoModel::\$_table

String = 'sitelinc_view_perfil_acao' [line 9]

- **Var** Nome da tabela no banco na qual este Model atua.
- **Access** protected

Constructor *NULL* function viewPerfilAcaoModel::__construct(0) [line 17]

Function Parameters:

- *NULL* 0

Carrega todos os mÃ©todos contidos na classe pai.

- **Author** Paulo Viegas < pauloviegas93@gmail.com >
- **Access** public

Array function viewPerfilAcaoModel::gerarPermissoes(\$id_perfil) [line 56]

Function Parameters:

- *\$id_perfil* **\$id_perfil** Corresponde ao id do perfil desse usuÃ¡rio.

Gera todas as aÃ§Ãões que o perfil de usuÃ¡rio tem acesso.

- **Author** Paulo Viegas < pauloviegas93@gmail.com >
- **Access** public

Boolean function viewPerfilAcaoModel::verificaPermissao([\$acao = NULL]) [line 30]

Function Parameters:

- *String* **\$acao** Corresponde a um endereço de ação ou de página.

Verifica se o usuário tem permissão para realizar uma ação passada por parâmetro ou (caso esta seja NULL) a que se encontra disposta na URL.

- **Author** Paulo Viegas < pauloviegas93@gmail.com >
- **Access** public

Appendices

Appendix A - Class Trees

Package default

AbstractModel

- [AbstractModel](#)
 - [acaoModel](#)
 - [anexoModel](#)
 - [grupoModel](#)
 - [grupoTipoModel](#)
 - [grupoTipoVinculoModel](#)
 - [grupoVinculoModel](#)
 - [logModel](#)
 - [perfilModel](#)
 - [permissaoModel](#)
 - [postModel](#)
 - [projetoModel](#)
 - [tarefaModel](#)
 - [tituloModel](#)
 - [usuarioModel](#)
 - [usuarrrioTipoVinculoModel](#)
 - [viewGrupoTipoModel](#)
 - [viewPerfilAcaoModel](#)

AuthModel

- [CI_Model](#)
 - [AuthModel](#)

home

- SocialController
 - [home](#)

serviceauth

- MY_Controller
 - [serviceauth](#)

usuario

- MY_Controller
 - [usuario](#)

usuarioVinculoModel

- [usuarioVinculoModel](#)

Index

A

AuthModel	6
anexoModel::\$ table	5
anexoModel	5
AuthModel::getInfo()	6
AuthModel::logar()	6
<i>Realiza a autenticação do usuário no sistema, seguindo as seguintes regras:
1) A senha e o e-mail estejam corretos;
2) O usuário não esteja logicamente excluído;
3) O cadastro de usuário já tenha sido aprovado.

São criadas duas sessions, que serão destruídas quando o usuário fizer logout do sistema:
1 - usuario: Corresponde a todos os dados do usuário.
2 - permissões: Corresponde a todas as permissões dadas ao perfil deste usuário.</i>	
AuthModel::verificaLogin()	7
<i>Verifica se o usuário está logado, através da session usuario populada.</i>	
AuthModel::logout()	7
<i>Realiza o logout do usuario! Destruído as sessões de usuario e permissão.</i>	
acaoModel::\$ table	4
acaoModel	4
AbstractModel::deletar()	2
<i>Exclui logicamente (alterando para 0 o campo excluído do registro), ou fisicamente, um ou mais registros da tabela setada na classe filha, a partir de id, caso ele exista, ou a partir do segundo parâmetro, caso o id seja NULL e segundo parâmetro seja populado.
OBS 1: Se houver mais de um registro que correspondam aos valores dos parâmetros de busca passados, todos serão excluídos.</i>	
AbstractModel::alterar()	2
<i>Altera as informações da tabela setada na classe filha, através do id, caso não seja informado o segundo parâmetro, ou através do segundo parâmetro, caso este seja populado.
OBS: No Array \$dados deve estar contido o id a ser alterado, caso se deseje fazer a alteração pelo mesmo.</i>	
AbstractModel::\$ table	1
AbstractModel::descreverTabela()	3
<i>Busca no banco todas as colunas da tabela.</i>	
AbstractModel::inserir()	3
<i>Insere um registro na tabela</i>	
AbstractModel::recuperaTodos()	4
<i>Recupera todas as linhas (da tabela setada na classe filha) que não estão logicamente excluídas.</i>	
AbstractModel::recuperaPorParametro()	3
<i>Recupera todas as linhas (da tabela setada na classe filha) por id, caso exista, ou a partir do segundo parâmetro, caso o id seja NULL e segundo parâmetro seja populado.</i>	
AbstractModel	1

C

constructor tituloModel:: construct()	17
constructor tarefaModel:: construct()	16
constructor serviceauth:: construct()	15
<i>Carrega todos os métodos contidos na classe pai, bem como as Models (tituloModel, grupoModel e viewGrupoTipoModel), necessárias.</i>	
constructor projetoModel:: construct()	14
constructor usuario:: construct()	17
constructor usuarioModel:: construct()	18
constructor viewPerfilAcaoModel:: construct()	21
<i>Carrega todos os métodos contidos na classe pai.</i>	
constructor viewGrupoTipoModel:: construct()	20
constructor usuarioTipoVinculoModel:: construct()	19
constructor usuarioVinculoModel:: construct()	19
constructor postModel:: construct()	13
constructor permissaoModel:: construct()	13
constructor grupoModel:: construct()	8
constructor AuthModel:: construct()	6
<i>Carrega todos os métodos contidos na classe pai, bem como as Models (usuarioModel e ViewPerfiAcaoModel), necessárias.</i>	
constructor anexoModel:: construct()	5
constructor acaoModel:: construct()	5
constructor grupoTipoModel:: construct()	9
constructor grupoTipoVinculoModel:: construct()	9
constructor perfilModel:: construct()	12
constructor logModel:: construct()	12
constructor home:: construct()	11
<i>Carrega todos os métodos contidos na classe pai, necessárias.</i>	
constructor grupoVinculoModel:: construct()	10
constructor AbstractModel:: construct()	1
<i>Carrega todos os métodos contidos na classe pai.</i>	

G

grupoTipoVinculoModel::\$ table	9
grupoVinculoModel	10
grupoVinculoModel::\$ table	10
grupoTipoVinculoModel	9
grupoTipoModel::\$ table	9
grupoModel::\$ table	8
grupoTipoModel	8
grupoModel	8

H

home::index()	11
home	10

L

logModel::\$ table	11
logModel	11

P

postModel::\$ table	13
projetoModel	14
projetoModel::\$ table	14
postModel	13
permissaoModel::\$ table	13
perfilModel::\$ table	12
permissaoModel	12
perfilModel	12

S

serviceauth::logout()	16
serviceauth::logar()	15
<i>Realiza a requisi��o ao Model authModel o login do usu��rio, caso valido</i>	
serviceauth::index()	15
<i>Carrega a p��gina inicial do sistema, onde o usu��rio poder�� autenticar-se ou, caso n��o fa��a parte da rede social ainda, realize o seu cadastro.</i>	
serviceauth	14

T

tituloModel::\$ table	17
tituloModel	17
tarefaModel::\$ table	16
tarefaModel	16

U

usuarioVinculoModel::\$ table	19
usuarrioTipoVinculoModel	19
usuarrioTipoVinculoModel::\$ table	19
usuarioVinculoModel	18
usuarioModel::\$ table	18
usuario::inserir()	18
usuarioModel	18
usuario	17

V

viewPerfilAcaoModel::gerarPermissoes()	21
<i>Gera todas as a���es que o perfil de usu��rio tem acesso.</i>	

viewPerfilAcaoModel::verificaPermissao()	21
<i>Verifica se o usuário tem permissão para realizar uma ação passada por parâmetro ou (caso esta seja NULL) a que se encontra disposta na URL.</i>	
viewPerfilAcaoModel::\$ table	21
viewPerfilAcaoModel	20
viewGrupoTipoModel::\$ table	20
viewGrupoTipoModel	20