SJJPadel

1.0

Generado por Doxygen 1.8.13

Índice general

1	SJJI	Padel																						1
2	Indic	ce jerár	quico																					3
	2.1	Jerarq	uía de la c	clas	se												 				 •	 		3
3	Índio	ce de es	structura (de	dat	os																		5
	3.1	Estruc	tura de da	atos	.												 				 •	 	•	5
4	Doc	umenta	ción de la	as e	estr	uctur	ras	de	e da	atos	S													7
	4.1	Refere	ncia de la	ı Cl	ase	Base	∍Со	ontr	rolle	er.							 					 		7
		4.1.1	Descripc	ciór	n de	tallad	la .										 					 		8
	4.2	Refere	ncia de la	ı Cl	ase	Cate	gor	ry									 					 		8
		4.2.1	Docume	enta	ciór	ı de la	as f	fun	ncio	nes	s mi	em	bro				 					 		8
			4.2.1.1	С	hec	klsVa	alidF	For	rUp	date	e()						 					 		8
	4.3	Refere	ncia de la	ı Cl	ase	Cate	gor	ryC	Char	mpi	ions	ship					 					 		9
	4.4	Refere	ncia de la	ı Cl	ase	Cate	gor	ryC	Char	mpi	ions	ship	Ма	ppe	er		 					 		9
		4.4.1	Docume	enta	ciór	ı de la	as f	fun	ncio	nes	s mi	em	bro				 					 		9
			4.4.1.1	g	jetC	atego	orie	sFı	rom	1Ch	am	pio	nsh	ip()			 					 		9
			4.4.1.2	g	jetC	ouple	es()										 					 		10
	4.5	Refere	ncia de la	ı Cl	ase	Cate	gor	ryC	Cont	troll	ler						 					 		10
		4.5.1	Docume	enta	ciór	ı de la	as f	fun	ncio	nes	s mi	em	bro				 					 		10
			4.5.1.1	a	add())											 					 		10
			4.5.1.2	d	lelet	e() .											 					 		11
			4.5.1.3	е	edit()												 					 		11
			4.5.1.4	S	how	/all()											 			 				11

ÍNDICE GENERAL

4.6	Refere	ncia de la	Clase CategoryMapper	12
	4.6.1	Docume	ntación de las funciones miembro	12
		4.6.1.1	save()	12
4.7	Refere	ncia de la	Clase Championship	12
	4.7.1	Docume	ntación de las funciones miembro	13
		4.7.1.1	checkIsValidForUpdate()	13
		4.7.1.2	needGenerateCalendar()	13
4.8	Refere	ncia de la	Clase ChampionshipController	14
	4.8.1	Descripc	ión detallada	14
	4.8.2	Docume	ntación de las funciones miembro	14
		4.8.2.1	add()	14
		4.8.2.2	delete()	15
		4.8.2.3	deleteInscription()	15
		4.8.2.4	edit()	15
		4.8.2.5	selectToCalendar()	15
		4.8.2.6	showall()	16
		4.8.2.7	showallInscriptionAllUsers()	16
		4.8.2.8	showallInscriptionCurrentUser()	16
4.9	Refere	ncia de la	Clase ChampionshipMapper	16
	4.9.1	Docume	ntación de las funciones miembro	17
		4.9.1.1	getCampeonato()	17
		4.9.1.2	updateFase()	17
4.10	Refere	ncia de la	Clase Confrontation	18
4.11	Refere	ncia de la	Clase ConfrontationController	18
4.12	Refere	ncia de la	Clase ConfrontationMapper	19
	4.12.1	Docume	ntación de las funciones miembro	19
		4.12.1.1	hasAllConfrontationsResult()	19
		4.12.1.2	save()	19
4.13	Refere	ncia de la	Clase ConfrontationOffer	20
	4.13.1	Descripo	ión detallada	20

ÍNDICE GENERAL III

4.14	Refere	ncia de la Clase ConfrontationOfferController	20
	4.14.1	Descripción detallada	21
4.15	Refere	ncia de la Clase ConfrontationOfferMapper	21
	4.15.1	Descripción detallada	21
	4.15.2	Documentación de las funciones miembro	21
		4.15.2.1 delete()	21
		4.15.2.2 save()	22
4.16	Refere	ncia de la Clase Fase	22
4.17	Refere	ncia de la Clase Group	23
4.18	Refere	ncia de la Clase GroupMapper	23
	4.18.1	Documentación de las funciones miembro	23
		4.18.1.1 getGroupsFromChampionship()	23
		4.18.1.2 save()	23
4.19	Refere	ncia de la Clase I18n	24
	4.19.1	Descripción detallada	24
	4.19.2	Documentación de las funciones miembro	25
		4.19.2.1 getAllMessages()	25
		4.19.2.2 getInstance()	25
		4.19.2.3 i18n()	25
		4.19.2.4 setLanguage()	26
4.20	Refere	ncia de la Clase InscriptionUserChampionship	26
	4.20.1	Documentación de las funciones miembro	26
		4.20.1.1 checklsValidForUpdate()	26
4.21	Refere	ncia de la Clase InscriptionUserChampionshipMapper	27
4.22	Refere	ncia de la Clase LanguageController	27
	4.22.1	Descripción detallada	27
		Documentación de las funciones miembro	28
		4.22.2.1 change()	28
		4.22.2.2 i18njs()	28
4.23	Refere	ncia de la Clase OrganizedMatch	28
0			_0

IV ÍNDICE GENERAL

4.24 Referencia de la Clase OrganizedMatchMapper	29
4.25 Referencia de la Clase OrganizeMatch	29
4.26 Referencia de la Clase OrganizeMatchController	29
4.26.1 Descripción detallada	30
4.27 Referencia de la Clase OrganizeMatchMapper	30
4.27.1 Documentación de las funciones miembro	30
4.27.1.1 delete()	30
4.27.1.2 findAll()	30
4.27.1.3 save()	31
4.28 Referencia de la Clase ParticipantsMatch	31
4.29 Referencia de la Clase ParticipantsMatchMapper	32
4.29.1 Documentación de las funciones miembro	32
4.29.1.1 cancel()	32
4.29.1.2 save()	32
4.30 Referencia de la Clase Partner	33
4.31 Referencia de la Clase PartnerController	33
4.32 Referencia de la Clase PartnerGroup	34
4.33 Referencia de la Clase PartnergroupMapper	34
4.34 Referencia de la Clase PartnerMapper	34
4.35 Referencia de la Clase PDOConnection	35
4.36 Referencia de la Clase Reservation	35
4.37 Referencia de la Clase ReservationController	35
4.38 Referencia de la Clase ReservationMapper	36
4.39 Referencia de la Clase ScheduleController	36
4.40 Referencia de la Clase StatisticsController	37
4.41 Referencia de la Clase User	37
4.42 Referencia de la Clase UserMapper	38
4.42.1 Documentación de las funciones miembro	38
4.42.1.1 borrar()	38
4.42.1.2 isValidUser()	38

ÍNDICE GENERAL V

	4.42.1.3 save()	39
4.43 Refere	rencia de la Clase UsersController	39
4.44 Refere	rencia de la Clase ValidationException	40
4.44.1	1 Descripción detallada	40
4.44.2	2 Documentación de las funciones miembro	40
	4.44.2.1 getErrors()	40
4.45 Refere	rencia de la Clase ViewManager	41
4.45.1	1 Descripción detallada	41
4.45.2	2 Documentación de las funciones miembro	42
	4.45.2.1 getFragment()	42
	4.45.2.2 getVariable()	42
	4.45.2.3 moveToDefaultFragment()	42
	4.45.2.4 moveToFragment()	43
	4.45.2.5 popFlash()	43
	4.45.2.6 redirect()	43
	4.45.2.7 redirectToReferer()	44
	4.45.2.8 render()	44
	4.45.2.9 setFlash()	45
	4.45.2.10 setLayout()	45
	4.45.2.11 setVariable()	45
Índice		47

Capítulo 1

SJJPadel

Santiago Gómez Vilar

Julio Quinteiro Soto

Jonatan Couto Riádigos

2 SJJPadel

Capítulo 2

Indice jerárquico

2.1. Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

BaseController
CategoryController
ChampionshipController
ConfrontationController
ConfrontationOfferController
OrganizeMatchController
PartnerController
ReservationController
ScheduleController
StatisticsController
UsersController
Category
CategoryChampionship
CategoryChampionshipMapper
CategoryMapper
Championship
ChampionshipMapper
Confrontation
ConfrontationMapper
ConfrontationOffer
ConfrontationOfferMapper
Exception
ValidationException
Fase
Group
GroupMapper
l18n
InscriptionUserChampionship
InscriptionUserChampionshipMapper
LanguageController
OrganizedMatch
OrganizedMatchMapper
OrganizeMatch
OrganizeMatchMapper
ParticinantsMatch 3

Indice jerárquico

ParticipantsMatchMapper	 	 32
Partner	 	 33
PartnerGroup	 	 34
PartnergroupMapper	 	 34
PartnerMapper	 	 34
PDOConnection	 	 35
Reservation	 	 35
ReservationMapper	 	 36
User	 	 37
UserMapper	 	 38
ViowManager		41

Capítulo 3

Índice de estructura de datos

3.1. Estructura de datos

Lista de estructuras con una breve descripción:

BaseController	7
Category	8
CategoryChampionship	9
CategoryChampionshipMapper	9
CategoryController	10
CategoryMapper	12
Championship	12
ChampionshipController	14
ChampionshipMapper	16
Confrontation	18
ConfrontationController	18
ConfrontationMapper	19
ConfrontationOffer	20
ConfrontationOfferController	20
ConfrontationOfferMapper	21
Fase	22
Group	23
GroupMapper	23
l18n	24
InscriptionUserChampionship	26
InscriptionUserChampionshipMapper	27
LanguageController	27
OrganizedMatch	28
OrganizedMatchMapper	29
OrganizeMatch	29
OrganizeMatchController	29
OrganizeMatchMapper	30
ParticipantsMatch	31
ParticipantsMatchMapper	32
Partner	33
PartnerController	33
PartnerGroup	34
PartnergroupMapper	34
PartnerMapper	34
PDOConnection	35

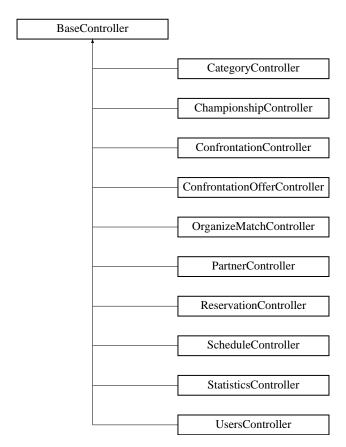
Reservation				 								 										35
ReservationController	٠,			 								 										35
ReservationMapper .				 								 										36
ScheduleController .				 								 										36
StatisticsController .				 								 										37
User				 								 										37
UserMapper				 								 										38
UsersController				 								 										39
ValidationException .				 								 										40
ViewManager																						//1

Capítulo 4

Documentación de las estructuras de datos

4.1. Referencia de la Clase BaseController

Diagrama de herencias de BaseController



Atributos protegidos

- \$view
- \$currentUser

4.1.1. Descripción detallada

Class BaseController

Implements a basic super constructor for the controllers in the Blog App. Basically, it provides some protected attributes and view variables.

Autor

```
lipido lipido@gmail.com
```

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/BaseController.php

4.2. Referencia de la Clase Category

Métodos públicos

- __construct (\$id=NULL, \$nivel=NULL, \$sexo=NULL)
- getId ()
- getNivel ()
- setNivel (\$nivel)
- getSexo ()
- setSexo (\$sexo)
- checklsValidForCreate ()
- checklsValidForUpdate ()

4.2.1. Documentación de las funciones miembro

4.2.1.1. checklsValidForUpdate()

```
checkIsValidForUpdate ( )
```

Checks if the current instance is valid for being updated in the database.

Excepciones

ValidationException	if the instance is not valid
validation Exception	in the instance is not valid

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

model/Category.php

4.3. Referencia de la Clase CategoryChampionship

Métodos públicos

- __construct (\$id=NULL, \$idChampionship=NULL, \$idCategory=NULL)
- getId ()
- getIdChampionship ()
- getIdCategory ()
- setId (\$id)
- setIdChampionship (\$idChampionship)
- setIdCategory (\$idCategory)

La documentación para esta clase fue generada a partir del siguiente fichero:

model/CategoryChampionship.php

4.4. Referencia de la Clase CategoryChampionshipMapper

Métodos públicos

- save (\$categoryChampionship)
- delete (\$idChampionship, \$idCategory)
- getCategoriesFromChampionship (\$idChampionship)
- getCouples (\$idCategoryChampionship)
- getCategoryFromChampionship (\$idChampionship, \$idCategory)
- getChampionshipFromIdCategory (\$idCategoriaCampeonato)
- existsCategory (\$idChampionship, \$idCategory)

4.4.1. Documentación de las funciones miembro

4.4.1.1. getCategoriesFromChampionship()

Obtiene las categorias de un tederminado campeonato

Parámetros

int \$idChampionship

Devuelve

Category[] categorias asociadas al campeonato

4.4.1.2. getCouples()

Obtiene todas las parejas apuntadas a esa categoria

Parámetros

```
int $idCategoryChampionship
```

Devuelve

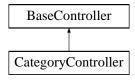
Partner[] parejas de la categoria

La documentación para esta clase fue generada a partir del siguiente fichero:

model/CategoryChampionshipMapper.php

4.5. Referencia de la Clase CategoryController

Diagrama de herencias de CategoryController



Métodos públicos

- showall ()
- delete ()
- edit ()
- add ()

Otros miembros heredados

4.5.1. Documentación de las funciones miembro

4.5.1.1. add()

add ()

Inserta una nueva categoría

Excepciones

Exception	Para insertar una categoria es necesario ser administrador	1

4.5.1.2. delete()

delete ()

Borra una categoría

Excepciones

Exception	Para borrar categorias es	necesario ser administrador
-----------	---------------------------	-----------------------------

4.5.1.3. edit()

edit ()

Muestra la vista de ediccion de categorías. Si la peticion proviene de esa vista se modifica la categoria con los datos aportados

Excepciones

Exception	Para ver esta vista es necesario ser administrador

4.5.1.4. showall()

showall ()

Muestra la vista con todas las categorías

Excepciones

Exception	Para ver esta vista es necesario ser administrador

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/CategoryController.php

4.6. Referencia de la Clase CategoryMapper

Métodos públicos

- save (\$category)
- delete (\$id)
- edit (\$level, \$sex, \$id)
- getDatos (\$id)
- esGeneroAceptado (\$idCategoria, \$genero)
- getCategorias ()
- getCategory (\$idCategory)
- categoryExists (Category \$category)
- getCouplesPerCategory ()

4.6.1. Documentación de las funciones miembro

```
4.6.1.1. save()
save (
$category )
```

Saves a User into the database

Parámetros

User	\$user	The user to be saved
------	--------	----------------------

Excepciones

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

model/CategoryMapper.php

4.7. Referencia de la Clase Championship

Métodos públicos

■ __construct (\$id=NULL, \$fechalnicioInscripcion=NULL, \$fechaFinInscripcion=NULL, \$fechalnicio← Campeonato=NULL, \$fechaFinCampeonato=NULL, \$nombreCampeonato=NULL, \$fase=NULL)

- getId ()
- getFechalnicioInscripcion ()
- setFechalnicioInscripcion (\$fecha)
- getFechaFinInscripcion ()
- setFechaFinInscripcion (\$fecha)
- getFechalnicioCampeonato ()
- setFechalnicioCampeonato (\$fecha)
- getFechaFinCampeonato ()
- setFechaFinCampeonato (\$fecha)
- getNombreCampeonato ()
- setNombreCampeonato (\$nombre)
- getFase ()
- setFase (\$fase)
- checklsValidForCreate ()
- checklsValidForUpdate ()
- needGenerateCalendar ()

4.7.1. Documentación de las funciones miembro

4.7.1.1. checklsValidForUpdate()

```
checkIsValidForUpdate ( )
```

Checks if the current instance is valid for being updated in the database.

Excepciones

ValidationEvacation	if the instance is not valid
validation⊏xception	I II the instance is not valid

Devuelve

void

4.7.1.2. needGenerateCalendar()

```
needGenerateCalendar ( )
```

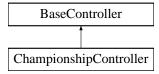
Retorna true en caso de que se deban generar enfrentamientos y false en caso de que aun no sea posible

La documentación para esta clase fue generada a partir del siguiente fichero:

model/Championship.php

4.8. Referencia de la Clase ChampionshipController

Diagrama de herencias de ChampionshipController



Métodos públicos

- showall ()
- add ()
- delete ()
- edit ()
- selectToCalendar ()
- showallInscriptionCurrentUser ()
- deleteInscription ()
- showallInscriptionAllUsers ()
- generateCalendar ()

Otros miembros heredados

4.8.1. Descripción detallada

Controlador de campeonatos

4.8.2. Documentación de las funciones miembro

4.8.2.1. add()

add ()

Inserta un campeonato

Excepciones

Exception Realizar acciones de campeonato requise ser admin

4.8.2.2. delete()

delete ()

Elimina un campeonato

Excepciones

Exception	Realizar acciones de campeonato requise ser admin
-----------	---

4.8.2.3. deleteInscription()

```
deleteInscription ( )
```

Borra la inscripcion a un campeonato

Excepciones

4.8.2.4. edit()

edit ()

Lleva a la vista de editar un capeonato

Excepciones

Exception Realizar acciones de campeonato requise ser admin

4.8.2.5. selectToCalendar()

selectToCalendar ()

Obtiene todos los champeonatos para los que se necesita generar grupos

Excepciones

Exception Realizar acciones de campeonato requise ser admin

4.8.2.6. showall()

showall ()

Lleva a la vista de ver campeonatos

Excepciones

Exception

4.8.2.7. showallInscriptionAllUsers()

```
showallInscriptionAllUsers ( )
```

Muestra los campeonatos que estan inscritos los usuarios

Excepciones

Exception Realizar acciones de campeonato requise ser admin

4.8.2.8. showallInscriptionCurrentUser()

```
showallInscriptionCurrentUser ( )
```

Muestra los campeonatos que esta inscrito el usuario que esta logeado

Excepciones

Exception Realizar acciones de campeonato requise ser admin

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/ChampionshipController.php

4.9. Referencia de la Clase ChampionshipMapper

Métodos públicos

save (\$championship)

- delete (\$id)
- edit (\$championship)
- getCampeonato (\$idCampeonato)
- getCampeonatos ()
- updateFase (\$idChampionship, \$fase)
- getCampeonatosInProgress ()
- getCampeonatosToGenerateGroups ()
- getCampeonatosParaIncripcion ()
- getCategorias (\$idCampeonato)
- getGrupos (\$idCampeonato, \$idCategoria)
- getNombreCampeonato (\$idCampeonato)
- validateHour (\$idChampionship, \$fechaOffer)
- getDatos (\$id)
- **checkPhase** (\$idCampeonato, \$fase)
- getCouplesPerChampionship ()

4.9.1. Documentación de las funciones miembro

4.9.1.1. getCampeonato()

```
\begin{tabular}{ll} \tt getCampeonato & ( & & \\ $idCampeonato \end{tabular} ) \end{tabular}
```

Obtiene los datos de un campeonato

Parámetros

int	\$idCampeonato	identificador del campeonato
-----	----------------	------------------------------

Devuelve

Championship Objeto con todos los datos del campeonato

4.9.1.2. updateFase()

Actualiza el campeonato indicado a la fase indicada

Parámetros

Championship	\$idChampionship	campeonato a actualizar
string	\$fase	nueva fase

La documentación para esta clase fue generada a partir del siguiente fichero:

model/ChampionshipMapper.php

4.10. Referencia de la Clase Confrontation

Métodos públicos

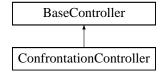
- __construct (\$idConfrontation=NULL, \$idPartner1=NULL, \$idPartner2=NULL, \$idGroup=NULL, \$phase=N ∪ ULL, \$date=NULL, \$time=NULL, \$pointsPartner1=NULL, \$pointsPartner2=NULL, \$setsPartner1=NULL, \$setsPartner2=NULL)
- getIdConfrontation ()
- getIdPartner1 ()
- setIdPartner1 (\$id)
- getIdPartner2 ()
- setIdPartner2 (\$id)
- getIdGroup ()
- setIdGroup (\$id)
- getPhase ()
- setPhase (\$id)
- getDate ()
- setDate (\$date)
- getTime ()
- setTime (\$time)
- getPointsPartner1 ()
- setPointsPartner1 (\$points)
- getPointsPartner2 ()
- setPointsPartner2 (\$points)
- getSetsPartner1 ()
- setSetPartner1 (\$sets)
- getSetsPartner2 ()
- setSetPartner2 (\$sets)
- checklsValidForRegister ()
- getPartner1Names()
- getPartner2Names ()

La documentación para esta clase fue generada a partir del siguiente fichero:

model/Confrontation.php

4.11. Referencia de la Clase ConfrontationController

Diagrama de herencias de ConfrontationController



Métodos públicos

- selectClasification ()
- clasification ()
- select ()
- setresults ()
- selectGroup ()
- showConfrontations ()

Otros miembros heredados

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/ConfrontationController.php

4.12. Referencia de la Clase ConfrontationMapper

Métodos públicos

- save (\$partnergroup)
- actualizarResultados (\$idEnfrentamiento, \$puntosPareja1, \$puntosPareja2, \$setsPareja1, \$setsPareja2)
- getPartidosResultadoNull (\$idGrupo, \$fase)
- getPartidos (\$idGrupo)
- getPartidosPorFase (\$idGrupo, \$fase)
- getPartidosCuartos (\$idGrupo)
- getPartidosSemifinal (\$idGrupo)
- getPartidosFinal (\$idGrupo)
- hadPlayed (\$idPareja1, \$idPareja2, \$idGrupo)
- getIdConfrontation (\$idPareja, \$idParejaOffer)
- actualizarHorario (\$idEnfrentamiento, \$fecha, \$hora)
- hasAllConfrontationsResult (\$idGroup)
- countLocalVictories (\$idPareja)
- countVisitantVictories (\$idPareja)
- countLocalDefeats (\$idPareja)
- countVisitantDefeats (\$idPareja)
- getBestResult (\$idPareja)

4.12.1. Documentación de las funciones miembro

4.12.1.1. hasAllConfrontationsResult()

```
\label{eq:loss_loss} \begin{split} \text{hasAllConfrontationsResult (} \\ & \quad \quad \quad \\ \text{$$\sharp$idGroup )} \end{split}
```

Devuelve true si todos los enfrentamientos de un grupo tienen resultado guardado en la base de datos y false en caso contrario

Inserta en la tabla enfrentamientos los id de pareja1, pareja2 y grupo del parametro enviado

Parámetros

Confrontation	\$partnergroup	enfrentamiento a registrar
---------------	----------------	----------------------------

La documentación para esta clase fue generada a partir del siguiente fichero:

model/ConfrontationMapper.php

4.13. Referencia de la Clase ConfrontationOffer

Métodos públicos

- __construct (\$idOfertaEnfrentamiento=NULL, \$idPareja=NULL, \$idGrupo=NULL, \$hora=NULL, \$fecha=N ∪ ULL, array \$pareja=NULL)
- getIdOfertaEnfrentamiento ()
- setIdOfertaEnfrentamiento (\$idOfertaEnfrentamiento)
- getIdPareja ()
- setIdPareja (\$idPareja)
- getIdGrupo ()
- setIdGrupo (\$idGrupo)
- getHora ()
- setHora (\$hora)
- getFecha ()
- setFecha (\$fecha)
- getPareja ()
- setPareja (\$pareja)
- checklsValidForCreate ()

4.13.1. Descripción detallada

Class ConfrontationOffer

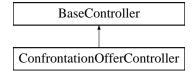
Represents a ConfrontationOffer in the app

La documentación para esta clase fue generada a partir del siguiente fichero:

model/ConfrontationOffer.php

4.14. Referencia de la Clase ConfrontationOfferController

Diagrama de herencias de ConfrontationOfferController



Métodos públicos

- view ()
- select ()
- join ()
- offer ()
- validateMatchs (\$idGrupo)

Otros miembros heredados

4.14.1. Descripción detallada

Class ConfrontationOfferController

Controller to offer a match and join to a tournament match

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/ConfrontationOfferController.php

4.15. Referencia de la Clase ConfrontationOfferMapper

Métodos públicos

- getConfrontationOffersForGroup (\$idGrupo)
- getOffer (\$idOfertaEnfrentamiento)
- save (ConfrontationOffer \$confrontationOffer)
- delete (\$idOfertaEnfrentamiento)

4.15.1. Descripción detallada

Class ConfrontationOfferMapper

Database interface for ConfrontationOffer entities

4.15.2. Documentación de las funciones miembro

Delete a ConfrontationOffer

Excepciones

PDOException	if a database error occurs
--------------	----------------------------

Saves a ConfrontationOffer into the database

Parámetros

Excepciones

PDOException	if a database error occurs
--------------	----------------------------

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

model/ConfrontationOfferMapper.php

4.16. Referencia de la Clase Fase

Campos de datos

- const **INSCRIPCION** = "Inscripcion"
- const **GRUPOS** = "Grupos"
- const **CUARTOS** = "Cuartos"
- const **SEMIFINAL** = "Semifinal"
- const **FINAL** = "Final"

La documentación para esta clase fue generada a partir del siguiente fichero:

model/Fase.php

4.17. Referencia de la Clase Group

Métodos públicos

- _construct (\$idGroup=NULL, \$idCategory=NULL, \$idChampionship=NULL, \$groupName=NULL)
- getIdGroup ()
- getIdCategory ()
- setIdCategory (\$id)
- getIdChampionship ()
- setIdChampionship (\$id)
- getGroupName ()
- setGroupName (\$name)
- checklsValidForRegister ()

La documentación para esta clase fue generada a partir del siguiente fichero:

■ model/Group.php

4.18. Referencia de la Clase GroupMapper

Métodos públicos

- save (\$group)
- getGrupoDefault (\$idCampeonato, \$idCategoria)
- getGroup (\$idCampeonato, \$idCategoria)
- getGroups (\$idCampeonato, \$idCategoria)
- getGroupsFromChampionship (\$idChampionship)

4.18.1. Documentación de las funciones miembro

4.18.1.1. getGroupsFromChampionship()

```
{\tt getGroupsFromChampionship (}\\ {\tt \$idChampionship )}
```

Retorna todos los grupos asociados a un campeonato

```
4.18.1.2. save() save ( $group )
```

Saves a User into the database

Parámetros

User \$user The user to be saved

Excepciones

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

model/GroupMapper.php

4.19. Referencia de la Clase I18n

Métodos públicos

- setLanguage (\$language)
- i18n (\$key)
- getAllMessages ()

Métodos públicos estáticos

■ static getInstance ()

Campos de datos

- const **DEFAULT_LANGUAGE** = "es"
- const CURRENT_LANGUAGE_SESSION_VAR = "__currentlang__"

4.19.1. Descripción detallada

Class I18n

This class implements a helper class for Internationalization (I18n). Basically this Singleton class manages a set of translation files (located in /view/messages/language_[lang].php) and provides a translation function: i18n(string) You can also change the current language with the setLanguage function. The last selected language is saved in the user session so it is the language retrieved each time this class is instantiated. In addition this file creates a global function, i18n(), as a shortcut to the function.

Autor

lipido lipido@gmail.com

4.19.2. Documentación de las funciones miembro

```
4.19.2.1. getAllMessages()
```

```
getAllMessages ( )
```

Gets all the messages in the current language

Devuelve

mixed Array of translations

```
4.19.2.2. getInstance()
```

```
static getInstance ( ) [static]
```

Gets the singleton instance of this class

Devuelve

I18n The singleton instance

```
4.19.2.3. i18n()
```

```
i18n ( $key )
```

Finds the current language translation of a given key

Parámetros

string	\$key	The key to tranlate

Devuelve

string The translation of the given key

4.19.2.4. setLanguage()

Sets the language (and keeps it in the user session)

Parámetros

stri	ng <i>\$language</i>
------	----------------------

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

core/l18n.php

4.20. Referencia de la Clase InscriptionUserChampionship

Métodos públicos

- __construct (\$idPartner=NULL, \$idCaptain=NULL, \$idFellow=NULL, \$level=NULL, \$sex=NULL, \$name Championship=NULL)
- setIdPartner (\$id)
- getIdPartner ()
- setIdCaptain (\$id)
- getIdCaptain ()
- setIdFellow (\$id)
- getIdFellow ()
- setLevel (\$level)
- getLevel ()
- setSex (\$sex)
- getSex ()
- setNameChampionship (\$name)
- getNameChampionship ()
- checklsValidForCreate ()
- checklsValidForUpdate ()

4.20.1. Documentación de las funciones miembro

4.20.1.1. checklsValidForUpdate()

```
checkIsValidForUpdate ( )
```

Checks if the current instance is valid for being updated in the database.

Excepciones

ValidationException	if the instance is not valid
---------------------	------------------------------

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

model/InscriptionUserChampionship.php

4.21. Referencia de la Clase InscriptionUserChampionshipMapper

Métodos públicos

- delete (\$id)
- getDatos (\$id)
- getAllInscriptionsInChampionship ()
- getInscriptionsCurrentUserInChampionship (\$login)

La documentación para esta clase fue generada a partir del siguiente fichero:

model/InscriptionUserChampionshipMapper.php

4.22. Referencia de la Clase LanguageController

Métodos públicos

- change ()
- i18njs ()

Campos de datos

const LANGUAGE_SETTING = "__language__"

4.22.1. Descripción detallada

Class LanguageController

Controller to manage the session language. Allows you to change the current language by establishing it in the I18n singleton instance

Autor

```
lipido lipido@gmail.com
```

4.22.2. Documentación de las funciones miembro

4.22.2.1. change() change () Action to change the current language The expected HTTP parameters are:

■ lang: lange to change to (via HTTP GET)

Devuelve

void

4.22.2.2. i18njs()

```
i18njs ( )
```

Action get a Javascript script with a ji18n(key) function

This is useful to translate strings inside your javascript (.js) files You have to include a <script src="index.comphp?controller=languages&action=i18njs"> </script> tag before all your scripts (in the layout, for example).

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/LanguageController.php

4.23. Referencia de la Clase OrganizedMatch

La documentación para esta clase fue generada a partir del siguiente fichero:

model/OrganizedMatch.php

4.24. Referencia de la Clase OrganizedMatchMapper

La documentación para esta clase fue generada a partir del siguiente fichero:

model/OrganizedMatchMapper.php

4.25. Referencia de la Clase OrganizeMatch

Métodos públicos

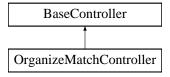
- __construct (\$idOrganizeMatch=NULL, \$fecha=NULL, \$hora=NULL, array \$participants=NULL, \$num↔ Participants=NULL)
- getldOrganizarPartido ()
- setIdOrganizarPartido (\$idOrganizarPartido)
- getFecha ()
- setFecha (\$fecha)
- getHora ()
- setHora (\$hora)
- getParticipants ()
- setParticipants (\$participants)
- getNumParticipants ()
- setNumParticipants (\$numParticipants)
- checklsValidForCreate ()

La documentación para esta clase fue generada a partir del siguiente fichero:

model/OrganizeMatch.php

4.26. Referencia de la Clase OrganizeMatchController

Diagrama de herencias de OrganizeMatchController



Métodos públicos

- add ()
- viewAll ()
- join ()
- delete ()
- cancel ()
- validateAllOrganizeMatches ()

Otros miembros heredados

4.26.1. Descripción detallada

Class OrganizeMatchController

Controller to organize a match

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/OrganizeMatchController.php

4.27. Referencia de la Clase OrganizeMatchMapper

Métodos públicos

- save (OrganizeMatch \$organizeMatch)
- **find** (\$idOrganizarPartido)
- findAll ()
- exist (\$id)
- findMatchWithParticipants (\$id)
- delete (\$idOrganizarPartido)

4.27.1. Documentación de las funciones miembro

Delete an organize Match

Excepciones

```
4.27.1.2. findAll()
```

```
findAll ( )
```

Find all the matches organized by an admin

Excepciones

Devuelve

Array \$organizedMatchesArray

Saves a OrganizeMatch into the database

Parámetros

rganizeMatch	\$organizeMarch	The match to be saved
--------------	-----------------	-----------------------

Excepciones

PDOException	if a database error occurs
--------------	----------------------------

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

model/OrganizeMatchMapper.php

4.28. Referencia de la Clase ParticipantsMatch

Métodos públicos

- __construct (\$idParticipantesPartido=NULL, \$idOrganizarPartido=NULL, \$loginUsuario=NULL)
- getIdParticipantesPartido ()
- setIdParticipantesPartido (\$idParticipantesPartido)
- getIdOrganizarPartido ()
- setIdOrganizarPartido (\$idOrganizarPartido)
- getLoginUsuario ()
- setLoginUsuario (\$loginUsuario)
- checklsValidForCreate ()

La documentación para esta clase fue generada a partir del siguiente fichero:

model/ParticipantsMatch.php

4.29. Referencia de la Clase ParticipantsMatchMapper

Métodos públicos

- play (\$idOrganizeMatch, \$userLogin)
- save (ParticipantsMatch \$paticipantMatch)
- count (\$idOrganizeMatch)
- cancel (\$idOrganizeMatch, \$userLogin)

4.29.1. Documentación de las funciones miembro

Delete an organize Match

Excepciones

PDOException	if a database error occurs
--------------	----------------------------

```
4.29.1.2. save()
```

```
save (
```

ParticipantsMatch \$paticipantMatch)

Saves a OrganizeMatch into the database

Parámetros

ParticipantsMatch	\$paticipantMatch	The participant to be saved
-------------------	-------------------	-----------------------------

Excepciones

PDOException	if a database error occurs

Devuelve

void

La documentación para esta clase fue generada a partir del siguiente fichero:

model/ParticipantsMatchMapper.php

4.30. Referencia de la Clase Partner

Métodos públicos

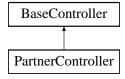
- __construct (\$idPartner=NULL, \$idCaptain=NULL, \$idFellow=NULL, \$idCategoryChampionship=NULL)
- getIdPartner ()
- getIdCaptain ()
- setIdCaptain (\$id)
- getIdFellow ()
- setIdFellow (\$id)
- getIdCategoryChampionship ()
- setIdCategoryChampionship (\$id)
- getVictories ()
- getDefeats ()
- setVictories (\$victories)
- setDefeats (\$defeats)
- getVictoryRate ()
- getTotalMatches ()
- checklsValidForRegister ()

La documentación para esta clase fue generada a partir del siguiente fichero:

model/Partner.php

4.31. Referencia de la Clase PartnerController

Diagrama de herencias de PartnerController



Métodos públicos

- selectChampionship ()
- inscription ()

Otros miembros heredados

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/PartnerController.php

4.32. Referencia de la Clase PartnerGroup

Métodos públicos

- __construct (\$idPartner=NULL, \$idGroup=NULL)
- getIdPartner ()
- setIdPartner (\$id)
- getIdGroup ()
- setIdGroup (\$id)
- checklsValidForRegister ()

La documentación para esta clase fue generada a partir del siguiente fichero:

model/Partnergroup.php

4.33. Referencia de la Clase PartnergroupMapper

Métodos públicos

- **save** (\$partnergroup)
- getIdParejasGrupo (\$idGrupo)
- getIdGrupo (\$idPareja)
- hasGroup (\$idPareja)

La documentación para esta clase fue generada a partir del siguiente fichero:

model/PartnergroupMapper.php

4.34. Referencia de la Clase PartnerMapper

Métodos públicos

- save (\$partner)
- existeParejaCategoria (\$login, \$idCategoriaCampeonato)
- devolverPareja (\$idCapitan, \$idCompañero, \$idCategoria)
- getParejas ()
- getPartnerNames (\$idPartner)
- getMyParners (\$user)
- getIdCapitan (\$idPareja)
- getMembers (\$idPareja)

La documentación para esta clase fue generada a partir del siguiente fichero:

model/PartnerMapper.php

4.35. Referencia de la Clase PDOConnection

Métodos públicos estáticos

static getInstance ()

La documentación para esta clase fue generada a partir del siguiente fichero:

core/PDOConnection.php

4.36. Referencia de la Clase Reservation

Métodos públicos

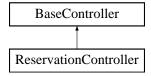
- __construct (\$idReservation=NULL, \$idUserReservation=NULL, \$dateReservation=NULL, \$hour ← Reservation=NULL)
- getIdReservation ()
- getIdUserReservation ()
- getDateReservation ()
- getHourReservation ()
- setIdReservation (\$idReservation)
- setIdUserReservation (\$idUserReservation)
- setDateReservation (\$dateReservation)
- setHourReservation (\$hourReservation)

La documentación para esta clase fue generada a partir del siguiente fichero:

model/Reservation.php

4.37. Referencia de la Clase ReservationController

Diagrama de herencias de ReservationController



Métodos públicos

- showAvaliableSchedules ()
- add ()
- showInfo ()

Otros miembros heredados

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/ReservationController.php

4.38. Referencia de la Clase ReservationMapper

Métodos públicos

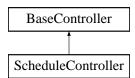
- getReservations ()
- makeReservation (\$reservation)
- getNumReservations (\$date, \$hour)
- getReservationCount ()
- **getReservationId** (\$idUsuarioReserva, \$fechaReserva, \$horaReserva)
- getReservation (\$idReservation)
- getReservationDayStatistics ()
- getReservationHourStatistics ()
- getReservationComingStatistics ()
- getUserReservations (\$login)

La documentación para esta clase fue generada a partir del siguiente fichero:

model/ReservationMapper.php

4.39. Referencia de la Clase ScheduleController

Diagrama de herencias de ScheduleController



Métodos públicos

- viewChampionshipMatches ()
- viewReservations ()
- viewOrganizedMatches ()

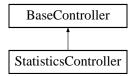
Otros miembros heredados

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/ScheduleController.php

4.40. Referencia de la Clase StatisticsController

Diagrama de herencias de StatisticsController



Métodos públicos

- reservationStatistics ()
- championshipStatistics ()
- personalStatistics ()

Otros miembros heredados

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/StatisticsController.php

4.41. Referencia de la Clase User

Métodos públicos

- __construct (\$login=NULL, \$username=NULL, \$surname=NULL, \$pass=NULL, \$rol=NULL, \$gender=N ← ULL)
- getLogin ()
- setLogin (\$login)
- getUsername ()
- setUsername (\$username)
- getSurname ()
- setSurname (\$surname)
- getPass ()
- setPass (\$pass)
- getRol ()
- setRol (\$rol)
- getGender ()
- setGender (\$gender)
- checklsValidForRegister ()

La documentación para esta clase fue generada a partir del siguiente fichero:

model/User.php

4.42. Referencia de la Clase UserMapper

Métodos públicos

- save (\$user)
- borrar (\$login)
- editar (\$login, \$nombre, \$apellidos, \$pass, \$rol, \$genero)
- loginExists (\$login)
- isValidUser (\$login, \$pass)
- getDatos (\$login)
- getUsuarios ()

4.42.1. Documentación de las funciones miembro

```
4.42.1.1. borrar()
borrar (
$login )
```

Checks if a given username is already in the database

Parámetros

- 4	Φ	Alexander and Alexander alice alice
string	\$username	the username to check

Devuelve

boolean true if the username exists, false otherwise

4.42.1.2. isValidUser()

```
isValidUser ( $login, $pass )
```

Checks if a given pair of username/password exists in the database

Parámetros

string	\$username	the username
string	\$passwd	the password

Devuelve

boolean true the username/passwrod exists, false otherwise.

4.42.1.3. save() save (\$user)

Saves a User into the database

Parámetros

User	\$user	The user to be saved
------	--------	----------------------

Excepciones

PDOException	if a database error occurs
--------------	----------------------------

Devuelve

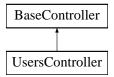
void

La documentación para esta clase fue generada a partir del siguiente fichero:

model/UserMapper.php

4.43. Referencia de la Clase UsersController

Diagrama de herencias de UsersController



Métodos públicos

- index ()
- login ()
- register ()
- showall ()
- delete ()
- edit ()
- add ()
- logout ()

Otros miembros heredados

La documentación para esta clase fue generada a partir del siguiente fichero:

controller/UsersController.php

4.44. Referencia de la Clase Validation Exception

Diagrama de herencias de ValidationException



Métodos públicos

- __construct (array \$errors, \$msg=NULL)
- getErrors ()

4.44.1. Descripción detallada

Class ValidationException

A simple Exception including an array of errors useful for form validation. The errors array contains validation errors, normally indexed by form named parameters.

Autor

```
lipido lipido@gmail.com
```

4.44.2. Documentación de las funciones miembro

```
4.44.2.1. getErrors()
```

```
getErrors ( )
```

Gets the validation errors

Devuelve

mixed The validation errors

La documentación para esta clase fue generada a partir del siguiente fichero:

core/ValidationException.php

4.45. Referencia de la Clase ViewManager

Métodos públicos

- moveToFragment (\$name)
- moveToDefaultFragment ()
- getFragment (\$fragment, \$default="")
- setVariable (\$varname, \$value, \$flash=false)
- getVariable (\$varname, \$default=NULL)
- setFlash (\$flashMessage)
- popFlash ()
- setLayout (\$layout)
- render (\$controller, \$viewname)
- redirect (\$controller, \$action, \$queryString=NULL)
- redirectToReferer (\$queryString=NULL)

Métodos públicos estáticos

static getInstance ()

Campos de datos

const DEFAULT_FRAGMENT = "__default__"

4.45.1. Descripción detallada

Class ViewManager

This class implements the glue between the controller and the view.

This class is a singleton. You should use getInstance() to get the view manager instance.

The main responsibilities are:

- 1. Save variables from the controller and make them available for views. This includes "flash" variables, which are variables kept in session that are removed just after they are retrieved. Use methods like setVariable, getVariable and setFlash.
- 2.Render views. This basically performs an 'include' of the view file, but with more MVC-oriented parameters (controller name and view name).
- 3.Layout (or templating) system. Based on PHP output buffers (ob_functions). Once the view manager is initialized, the output buffer is enabled. By default, all contents that are generated inside your views will be saved in a DE ← FAULT_FRAGMENT. The DEFAULT_FRAGMENT is normally used as the "main" content of the resulting layout. However, you can generate contents for other fragments that will go into the layout. For example, inside your views, you have to call moveToFragment(fragmentName) before generating content for a desired fragment. This fragment normally will be after retrieved by the layout (via calls to getFragment). Typical fragments are 'css', 'javascript', so you can specify additional css and javascripts from your specific views.

Autor

lipido lipido@gmail.com

4.45.2. Documentación de las funciones miembro

4.45.2.1. getFragment()

Gets the contents occumulated in an specified fragment

Parámetros

string	\$fragment	The fragment to retrieve the contents from
string	\$default	The default content if the \$fragment does not exist

Devuelve

string The fragment contents

4.45.2.2. getVariable()

Retrieves a previously established variable.

If the variable is a flash variable, it removes it from the session after being retrieved

Parámetros

string \$varname The name of the variable \$default Object value of the variable		\$varname	The name of the variable
		\$default	Object value of the variable to return if the variable does not exists

Devuelve

Object value of the variable

4.45.2.3. moveToDefaultFragment()

```
moveToDefaultFragment ( )
```

Changes to the default fragment.

The current output is saved before changing. The subsequent outputs will be accumulated in the default fragment

Devuelve

void

4.45.2.4. moveToFragment()

Changes the current fragment where output is accumulating

The current output is saved before changing. The subsequent outputs will be accumulted in the specified fragment.

Parámetros

	string	\$name	The name of the fragment to move to]
--	--------	--------	-------------------------------------	---

Devuelve

void

4.45.2.5. popFlash()

```
popFlash ( )
```

Retrieves the flash message (and pops it)

Devuelve

string The flash message

4.45.2.6. redirect()

```
redirect (
$controller,
$action,
$queryString = NULL )
```

Sends an HTTP 302 redirection to a given action inside a controller

Parámetros

string	\$controller	The name of the controller
string	\$action	The name of the action
Geography por Squae p String		An optional query string

Devuelve

void

4.45.2.7. redirectToReferer()

```
redirectToReferer (
    $queryString = NULL )
```

Sends an HTTP 302 redirection to the refererring page, which is the page where the user was, just before making the current request.

Parámetros

	string	\$queryString	An optional query string
--	--------	---------------	--------------------------

Devuelve

void

4.45.2.8. render()

```
render (
$controller,
$viewname)
```

Renders an specified view of a specified controller

If the \$controller=mycontroller and \$view=myview, the selected php file will be: view/mycontroller/myview.php

It uses the the selected layout (via setLayout) or the default layout if it was not specified before calling the setLayout method

Parámetros

string	\$controller	Name of the controller (in URL format e.g: "posts")
string	\$viewname	Name of the view

Devuelve

void

4.45.2.9. setFlash()

```
{\tt setFlash \ (} \\ {\tt \$flashMessage \ )}
```

Establishes a flash message

Flash messages are useful to pass text from one page to other via HTTP redirects, sinde they are kept in session.

Parámetros

	string	\$flashMessage	The message to save into session	
--	--------	----------------	----------------------------------	--

Devuelve

void

4.45.2.10. setLayout()

```
setLayout (
     $layout )
```

Sets the layout to be used when renderLayout will be called

Parámetros

	string	\$layout	The layout to use
--	--------	----------	-------------------

Devuelve

void

4.45.2.11. setVariable()

Establishes a variable for the view

Variables could be also kept in session (via \$flash parameter)

Parámetros

string	\$varname	The name of the variable
Object	\$value	The value of the variable
boolean	\$flash	If the variable value shoud be kept in session

La documentación para esta clase fue generada a partir del siguiente fichero:

core/ViewManager.php

Índice alfabético

add	ConfrontationOfferController, 20
CategoryController, 10	ConfrontationOfferMapper, 21
ChampionshipController, 14	delete, 21
	save, 22
BaseController, 7	
borrar	delete
UserMapper, 38	CategoryController, 11
aanaal	ChampionshipController, 14
cancel Participants Match Mannay 80	ConfrontationOfferMapper, 21
ParticipantsMatchMapper, 32	OrganizeMatchMapper, 30
Category, 8	deleteInscription
checklsValidForUpdate, 8	ChampionshipController, 15
CategoryChampionship, 9	
CategoryChampionshipMapper, 9	edit
getCategoriesFromChampionship, 9	CategoryController, 11
getCouples, 10	ChampionshipController, 15
CategoryController, 10	
add, 10	Fase, 22
delete, 11	findAll
edit, 11	OrganizeMatchMapper, 30
showall, 11	
CategoryMapper, 12	getAllMessages
save, 12	l18n, <mark>25</mark>
Championship, 12	getCampeonato
checkIsValidForUpdate, 13	ChampionshipMapper, 17
needGenerateCalendar, 13	getCategoriesFromChampionship
ChampionshipController, 14	CategoryChampionshipMapper, 9
add, 14	getCouples
delete, 14	CategoryChampionshipMapper, 1
deleteInscription, 15	getErrors
edit, 15	ValidationException, 40
selectToCalendar, 15	getFragment
showall, 16	ViewManager, 42
showallInscriptionAllUsers, 16	getGroupsFromChampionship
showallInscriptionCurrentUser, 16	GroupMapper, 23
ChampionshipMapper, 16	getInstance
getCampeonato, 17	I18n, 25
updateFase, 17	getVariable
change	ViewManager, 42
LanguageController, 28	Group, 23
checkIsValidForUpdate	GroupMapper, 23
Category, 8	getGroupsFromChampionship, 23
Championship, 13	save, 23
InscriptionUserChampionship, 26	3dv6, 20
Confrontation, 18	hasAllConfrontationsResult
ConfrontationController, 18	ConfrontationMapper, 19
ConfrontationMapper, 19	Comonationwappor, 10
hasAllConfrontationsResult, 19	l18n, 24
save, 19	getAllMessages, 25
ConfrontationOffer, 20	getInstance, 25
 	g,

48 ÍNDICE ALFABÉTICO

i18n, 25	GroupMapper, 23
setLanguage, 25	OrganizeMatchMapper, 31
i18n	ParticipantsMatchMapper, 32
I18n, 25	UserMapper, 39
i18njs	ScheduleController, 36
LanguageController, 28	selectToCalendar
InscriptionUserChampionship, 26	ChampionshipController, 15
checklsValidForUpdate, 26	setFlash
InscriptionUserChampionshipMapper, 27	ViewManager, 44
isValidUser	setLanguage
UserMapper, 38	I18n, <mark>25</mark>
	setLayout
LanguageController, 27	ViewManager, 45
change, 28	setVariable
i18njs, <mark>28</mark>	ViewManager, 45
	showall
moveToDefaultFragment	CategoryController, 11
ViewManager, 42	ChampionshipController, 16
moveToFragment	showallInscriptionAllUsers
ViewManager, 43	ChampionshipController, 16
	showallInscriptionCurrentUser
needGenerateCalendar	ChampionshipController, 16
Championship, 13	StatisticsController, 37
	StatisticsController, 37
OrganizeMatch, 29	updateFase
OrganizeMatchController, 29	ChampionshipMapper, 17
OrganizeMatchMapper, 30	User, 37
delete, 30	
findAll, 30	UserMapper, 38
save, 31	borrar, 38
OrganizedMatch, 28	isValidUser, 38
OrganizedMatchMapper, 29	save, 39
	UsersController, 39
PDOConnection, 35	Velidation Eventual 40
ParticipantsMatch, 31	ValidationException, 40
ParticipantsMatchMapper, 32	getErrors, 40
cancel, 32	ViewManager, 41
save, 32	getFragment, 42
Partner, 33	getVariable, 42
PartnerController, 33	moveToDefaultFragment, 42
PartnerGroup, 34	moveToFragment, 43
PartnerMapper, 34	popFlash, 43
PartnergroupMapper, 34	redirect, 43
popFlash	redirectToReferer, 44
ViewManager, 43	render, 44
Training or,	setFlash, 44
redirect	setLayout, 45
ViewManager, 43	setVariable, 45
redirectToReferer	
ViewManager, 44	
render	
ViewManager, 44	
Reservation, 35	
ReservationController, 35	
ReservationMapper, 36	
i iesei valioi iiviappei, su	
save	
CategoryMapper, 12	
ConfrontationMapper, 19	
ConfrontationOfferMapper, 22	