

ExpenShare

Generated by Doxygen 1.8.6

Thu Nov 27 2014 13:10:25

Contents

1	Main Page	1
2	Namespace Index	3
2.1	Packages	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Namespace Documentation	11
6.1	forms Namespace Reference	11
6.2	models Namespace Reference	11
6.3	urls Namespace Reference	11
6.3.1	Variable Documentation	12
6.3.1.1	urlpatterns	12
6.4	views Namespace Reference	12
6.4.1	Function Documentation	12
6.4.1.1	add_groupform	12
6.4.1.2	add_payform	13
6.4.1.3	confirmPayment	13
6.4.1.4	history	13
6.4.1.5	home	13
6.4.1.6	index	14
6.4.1.7	joingroup_form	14
6.4.1.8	leavegroup	14
6.4.1.9	register	14
6.4.1.10	removePayForm	15
6.4.1.11	userLogin	15

6.4.1.12	userLogout	15
7	Class Documentation	17
7.1	models.FellowUser Class Reference	17
7.1.1	Detailed Description	17
7.1.2	Member Function Documentation	17
7.1.2.1	__unicode__	17
7.1.3	Member Data Documentation	18
7.1.3.1	owed	18
7.1.3.2	user	18
7.2	forms.MakeGroupForm Class Reference	18
7.2.1	Detailed Description	18
7.2.2	Member Data Documentation	18
7.2.2.1	description	18
7.2.2.2	name	18
7.2.2.3	passcode	18
7.3	models.MemberView Class Reference	19
7.3.1	Detailed Description	19
7.3.2	Member Function Documentation	19
7.3.2.1	__unicode__	19
7.3.3	Member Data Documentation	19
7.3.3.1	fellows	19
7.3.3.2	netOwed	19
7.3.3.3	user	19
7.4	forms.UserForm.Meta Class Reference	20
7.4.1	Detailed Description	20
7.4.2	Member Data Documentation	20
7.4.2.1	fields	20
7.4.2.2	model	20
7.5	forms.PayForm.Meta Class Reference	20
7.5.1	Detailed Description	20
7.5.2	Member Data Documentation	20
7.5.2.1	fields	20
7.5.2.2	model	20
7.6	forms.MakeGroupForm.Meta Class Reference	20
7.6.1	Detailed Description	21
7.6.2	Member Data Documentation	21
7.6.2.1	fields	21
7.6.2.2	model	21
7.7	forms.PayForm Class Reference	21

7.7.1	Detailed Description	21
7.7.2	Member Data Documentation	21
7.7.2.1	amount	21
7.7.2.2	description	22
7.8	models.PayGroup Class Reference	22
7.8.1	Detailed Description	22
7.8.2	Member Function Documentation	22
7.8.2.1	__unicode__	22
7.8.3	Member Data Documentation	23
7.8.3.1	description	23
7.8.3.2	groupSize	23
7.8.3.3	members	23
7.8.3.4	memberViews	23
7.8.3.5	name	23
7.8.3.6	passcode	23
7.8.3.7	paymentLogs	23
7.9	models.PaymentLog Class Reference	23
7.9.1	Detailed Description	23
7.9.2	Member Function Documentation	24
7.9.2.1	__unicode__	24
7.9.3	Member Data Documentation	24
7.9.3.1	amount	24
7.9.3.2	contested	24
7.9.3.3	contestedMessage	24
7.9.3.4	date	24
7.9.3.5	description	24
7.9.3.6	user	24
7.10	models.PayUser Class Reference	24
7.10.1	Detailed Description	25
7.10.2	Member Function Documentation	25
7.10.2.1	__unicode__	25
7.10.3	Member Data Documentation	25
7.10.3.1	payGroups	25
7.10.3.2	userKey	25
7.11	forms.UserForm Class Reference	25
7.11.1	Detailed Description	25
7.11.2	Member Data Documentation	26
7.11.2.1	password	26

8.1	forms.py File Reference	27
8.1.1	Detailed Description	27
8.2	mainpage.dox File Reference	28
8.3	models.py File Reference	28
8.3.1	Detailed Description	28
8.4	urls.py File Reference	28
8.4.1	Detailed Description	29
8.5	views.py File Reference	29
8.5.1	Detailed Description	30
Index		31

Chapter 1

Main Page

Authors

Taylor Andrews, Ian Char, Brennan McConnell

Date

11/26/2014

ExpenShare is an online application designed to help people split costs and share expenses. You can now create one or more group(s) with your friends and log your transactions, then let ExpenShare do the work and calculate how much each group member owes each other. This application can be used for roommates to split living expenses, friends to split travel costs, or any other necessary use!

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

forms	11
models	11
urls	11
views	12

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

forms.UserForm.Meta	20
forms.PayForm.Meta	20
forms.MakeGroupForm.Meta	20
Model	
models.FellowUser	17
models.MemberView	19
models.PayGroup	22
models.PaymentLog	23
models.PayUser	24
ModelForm	
forms.MakeGroupForm	18
forms.PayForm	21
forms.UserForm	25

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

models.FellowUser	Model designed to represent a fellow group member for a single particular MemberView	17
forms.MakeGroupForm	A form for creating a new ExpenShare group	18
models.MemberView	The MemberView model is designed to represent a single group member's view of the expense balances	19
forms.UserForm.Meta	Meta defines corresponding model (User) and fields	20
forms.PayForm.Meta	Meta defines corresponding model (PaymentLog) and fields	20
forms.MakeGroupForm.Meta	Meta defines corresponding model (PayGroup) and fields	20
forms.PayForm	A form for entering an expense amount and description	21
models.PayGroup	Model for an ExpenShare group	22
models.PaymentLog	Model designed to represent a transaction	23
models.PayUser	A model for a user of ExpenShare	24
forms.UserForm	A form for creating a brand new ExpenShare User	25

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

forms.py	Django forms for taking input from ExpenShare users	27
models.py	Contains the database models implemented by ExpenShare	28
urls.py	URL map for ExpenShare	28
views.py	Contains the functions which implement all the logic behind ExpenShare	29

Chapter 6

Namespace Documentation

6.1 forms Namespace Reference

Classes

- class [UserForm](#)
A form for creating a brand new ExpenShare User.
- class [PayForm](#)
A form for entering an expense amount and description.
- class [MakeGroupForm](#)
A form for creating a new ExpenShare group.

6.2 models Namespace Reference

Classes

- class [FellowUser](#)
Model designed to represent a fellow group member for a single particular [MemberView](#).
- class [MemberView](#)
The [MemberView](#) model is designed to represent a single group member's view of the expense balances.
- class [PaymentLog](#)
Model designed to represent a transaction.
- class [PayGroup](#)
Model for an ExpenShare group.
- class [PayUser](#)
A model for a user of ExpenShare.

6.3 urls Namespace Reference

Variables

- tuple [urlpatterns](#)
Contains all the url tuples for the ExpenShare website.

6.3.1 Variable Documentation

6.3.1.1 tuple `urls.urlpatterns`

Initial value:

```
1 = patterns(' ',
2     url(r'^$', views.index, name='index'),
3     url(r'^home/$', views.home, name='home'),
4     url(r'^history/$', views.history, name='history'),
5     url(r'^register/$', views.register, name='register'),
6     url(r'^add_payform/$', views.add_payform, name='add_payform'),
7     url(r'^add_groupform/$', views.add_groupform, name='add_groupform'),
8     url(r'^joining_group_form/$', views.joining_group_form, name='joining_group_form'),
9     url(r'^removePayForm/$', views.removePayForm, name='removePayForm'),
10    url(r'^leavegroup/$', views.leavegroup, name='leavegroup'),
11    url(r'^login/$', views.userLogin, name='login'),
12    url(r'^logout/$', views.userLogout, name='logout'),
13    url(r'^confirmPayment/$', views.confirmPayment, name='confirmPayment'))
```

Contains all the url tuples for the ExpenShare website.

6.4 views Namespace Reference

Functions

- `def index`
Returns the default ExpenShare page.
- `def home`
Displays the home page for the current User.
- `def history`
Displays the history webpage which shows all previous expenses for a specific group.
- `def register`
This function is used for registering new users.
- `def add_groupform`
This function creates a new PayGroup.
- `def add_payform`
Adds an expense to a group.
- `def joining_group_form`
Function which allows a User to join an existing PayGroup.
- `def leavegroup`
Function which makes a user leave a PayGroup upon request.
- `def userLogin`
Logs a user into their ExpenShare profile.
- `def userLogout`
Logs a User out.
- `def confirmPayment`
Confirms that a PayUser has received payment for existing expenses by a fellow member of the group.
- `def removePayForm`
Removes a payment from a group.

6.4.1 Function Documentation

6.4.1.1 `def views.add_groupform (request)`

This function creates a new PayGroup.

This function creates a new PayGroup and adds the user who created it to it.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirects home, passes an error if one occurred

6.4.1.2 `def views.add_payform (request)`

Adds an expense to a group.

Function implements adds a new paymentLog to a PayGroup. The function updates the amount owed amongst the group members as the payment is added.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirects to expense history, if an error occurred redirects home

6.4.1.3 `def views.confirmPayment (request)`

Confirms that a PayUser has received payment for existing expenses by a fellow member of the group.

Allows user to specify which member of the group they are reporting as having paid them back. Updates the amount owed between users and the group.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirects user to their homepage, will pass an error if one occurred

6.4.1.4 `def views.history (request)`

Displays the history webpage which shows all previous expenses for a specific group.

The page checks the current group the user has chosen and shows the expense history of that group. This is on the ExpenseLog.html page.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Directs to expense history unless user cannot access that group

6.4.1.5 `def views.home (request)`

Displays the home page for the current User.

Home displays any paygroups an user is in. It allows a user to report new expenses and create new groups.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirects to home page

6.4.1.6 def views.index (*request*)

Returns the default ExpenShare page.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirects to login

6.4.1.7 def views.joingroup_form (*request*)

Function which allows a User to join an existing PayGroup.

If the user correctly inputs the group name and passcode, the user is added to the PayGroup requested.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirects user back to their homepage with the group joined or passes an error

6.4.1.8 def views.leavegroup (*request*)

Function which makes a user leave a PayGroup upon request.

Pending eligibility, allows a PayUser to leave a Paygroup. If the PayGroup no longer has any members, it is deleted.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Directs user to their homepage with group removed, passes an error if one occurred

6.4.1.9 def views.register (*request*)

This function is used for registering new users.

This function is used to register a new user to the expenshare site. It creates a User Model for an individual when they sign up on the register.html page.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirects to next step in registration unless an error occurred

6.4.1.10 `def views.removePayForm (request)`

Removes a payment from a group.

Removes an expense from a PayGroup and removes it from the expense history log. Then recalculates the balances owed between group members.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirect user to the expense history, passes an error if one occurred

6.4.1.11 `def views.userLogin (request)`

Logs a user into their ExpenShare profile.

Allows an existing ExpenShare user to access their profile.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Redirects user to their homepage, passes an error if one occurred

6.4.1.12 `def views.userLogout (request)`

Logs a User out.

Parameters

<i>request</i>	An http request from ExpenShare
----------------	---------------------------------

Returns

Http redirect to the homepage

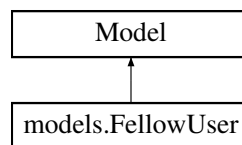
Chapter 7

Class Documentation

7.1 models.FellowUser Class Reference

Model designed to represent a fellow group member for a single particular [MemberView](#).

Inheritance diagram for models.FellowUser:



Public Member Functions

- def [__unicode__](#)
Gets model's information in unicode string format.

Static Public Attributes

- tuple [user](#) = models.ForeignKey(User)
- tuple [owed](#) = models.DecimalField(max_digits=11, decimal_places=2, default=0)

7.1.1 Detailed Description

Model designed to represent a fellow group member for a single particular [MemberView](#).

[FellowUser](#) models interact with a [MemberView](#) model in order to represent the [MemberView](#)'s balances which they owe their fellow group members.

7.1.2 Member Function Documentation

7.1.2.1 def models.FellowUser.__unicode__(self)

Gets model's information in unicode string format.

Returns

The fellow user's username

7.1.3 Member Data Documentation

7.1.3.1 tuple `models.FellowUser.owed = models.DecimalField(max_digits=11, decimal_places=2, default=0)` `[static]`

7.1.3.2 tuple `models.FellowUser.user = models.ForeignKey(User)` `[static]`

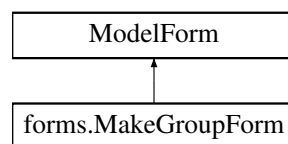
The documentation for this class was generated from the following file:

- [models.py](#)

7.2 forms.MakeGroupForm Class Reference

A form for creating a new ExpenShare group.

Inheritance diagram for `forms.MakeGroupForm`:



Classes

- class [Meta](#)
Meta defines corresponding model (PayGroup) and fields.

Static Public Attributes

- tuple `name = forms.CharField(max_length=20, label = 'Group Name')`
- tuple `description = forms.CharField(max_length=50, label = 'Group Description', widget=forms.TextInput(attrs={'size': '37'}))`
- tuple `passcode = forms.CharField(max_length=16, label = 'Group Passcode', widget=forms.PasswordInput())`

7.2.1 Detailed Description

A form for creating a new ExpenShare group.

[MakeGroupForm](#) requires a group name, description, and passcode.

7.2.2 Member Data Documentation

7.2.2.1 tuple `forms.MakeGroupForm.description = forms.CharField(max_length=50, label = 'Group Description', widget=forms.TextInput(attrs={'size': '37'}))` `[static]`

7.2.2.2 tuple `forms.MakeGroupForm.name = forms.CharField(max_length=20, label = 'Group Name')` `[static]`

7.2.2.3 tuple `forms.MakeGroupForm.passcode = forms.CharField(max_length=16, label = 'Group Passcode', widget=forms.PasswordInput())` `[static]`

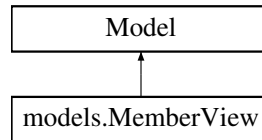
The documentation for this class was generated from the following file:

- [forms.py](#)

7.3 models.MemberView Class Reference

The [MemberView](#) model is designed to represent a single group member's view of the expense balances.

Inheritance diagram for models.MemberView:



Public Member Functions

- `def __unicode__`
Gets model's information in unicode string format.

Static Public Attributes

- tuple `user` = `models.ForeignKey(User)`
- tuple `netOwed` = `models.DecimalField(max_digits=11, decimal_places=2, default=0)`
- tuple `fellows` = `models.ManyToManyField(FellowUser)`

7.3.1 Detailed Description

The [MemberView](#) model is designed to represent a single group member's view of the expense balances.

[MemberView](#) represents a particular viewpoint of the expenses. Many [FellowUsers](#) are associated with a single [MemberView](#). The [MemberView](#) then represents how much the particular member owes his or her fellow group members. It is a look at the current balance owed to all members from the viewpoint of a particular group member.

7.3.2 Member Function Documentation

7.3.2.1 `def models.MemberView.__unicode__(self)`

Gets model's information in unicode string format.

Returns

The member's username

7.3.3 Member Data Documentation

7.3.3.1 `tuple models.MemberView.fellows = models.ManyToManyField(FellowUser)` [static]

7.3.3.2 `tuple models.MemberView.netOwed = models.DecimalField(max_digits=11, decimal_places=2, default=0)` [static]

7.3.3.3 `tuple models.MemberView.user = models.ForeignKey(User)` [static]

The documentation for this class was generated from the following file:

- [models.py](#)

7.4 forms.UserForm.Meta Class Reference

[Meta](#) defines corresponding model (User) and fields.

Static Public Attributes

- [model](#) = User
- tuple [fields](#) = ('email', 'username', 'password')

7.4.1 Detailed Description

[Meta](#) defines corresponding model (User) and fields.

7.4.2 Member Data Documentation

7.4.2.1 tuple `forms.UserForm.Meta.fields` = ('email', 'username', 'password') `[static]`

7.4.2.2 `forms.UserForm.Meta.model` = User `[static]`

The documentation for this class was generated from the following file:

- [forms.py](#)

7.5 forms.PayForm.Meta Class Reference

[Meta](#) defines corresponding model (PaymentLog) and fields.

Static Public Attributes

- [model](#) = [models.PaymentLog](#)
- tuple [fields](#) = ('amount', 'description')

7.5.1 Detailed Description

[Meta](#) defines corresponding model (PaymentLog) and fields.

7.5.2 Member Data Documentation

7.5.2.1 tuple `forms.PayForm.Meta.fields` = ('amount', 'description') `[static]`

7.5.2.2 `forms.PayForm.Meta.model` = [models.PaymentLog](#) `[static]`

The documentation for this class was generated from the following file:

- [forms.py](#)

7.6 forms.MakeGroupForm.Meta Class Reference

[Meta](#) defines corresponding model (PayGroup) and fields.

Static Public Attributes

- `model` = `models.PayGroup`
- tuple `fields` = `('name', 'description', 'passcode')`

7.6.1 Detailed Description

`Meta` defines corresponding model (`PayGroup`) and fields.

7.6.2 Member Data Documentation

7.6.2.1 tuple `forms.MakeGroupForm.Meta.fields` = `('name', 'description', 'passcode')` `[static]`

7.6.2.2 `forms.MakeGroupForm.Meta.model` = `models.PayGroup` `[static]`

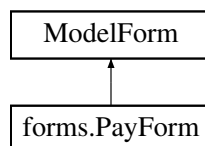
The documentation for this class was generated from the following file:

- `forms.py`

7.7 forms.PayForm Class Reference

A form for entering an expense amount and description.

Inheritance diagram for `forms.PayForm`:



Classes

- class `Meta`
`Meta` defines corresponding model (`PaymentLog`) and fields.

Static Public Attributes

- tuple `amount` = `forms.DecimalField(max_digits=11, decimal_places=2, label = 'Amount')`
- tuple `description` = `forms.CharField(max_length=50, label = 'Description')`

7.7.1 Detailed Description

A form for entering an expense amount and description.

7.7.2 Member Data Documentation

7.7.2.1 tuple `forms.PayForm.amount` = `forms.DecimalField(max_digits=11, decimal_places=2, label = 'Amount')` `[static]`

7.7.2.2 tuple `forms.PayForm.description = forms.CharField(max_length=50, label = 'Description')` `[static]`

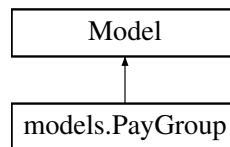
The documentation for this class was generated from the following file:

- [forms.py](#)

7.8 models.PayGroup Class Reference

Model for an ExpenShare group.

Inheritance diagram for models.PayGroup:



Public Member Functions

- `def __unicode__`
Gets model's information in unicode string format.

Static Public Attributes

- tuple `name` = `models.CharField(max_length=20, default="", unique=True)`
- tuple `description` = `models.CharField(max_length=50, default="")`
- tuple `members` = `models.ManyToManyField(User)`
- tuple `passcode` = `models.CharField(max_length=16, default="")`
- tuple `paymentLogs` = `models.ManyToManyField(PaymentLog)`
- tuple `memberViews` = `models.ManyToManyField(MemberView)`
- tuple `groupSize` = `models.IntegerField(default=1)`

7.8.1 Detailed Description

Model for an ExpenShare group.

`PayGroup` is a model designed to represent one expense sharing group. It is associated with multiple members and contains a `MemberView` model per member which represents a members particular view point of the expenses incurred.

7.8.2 Member Function Documentation

7.8.2.1 `def models.PayGroup.__unicode__(self)`

Gets model's information in unicode string format.

Returns

The name of the paygroup

7.8.3 Member Data Documentation

7.8.3.1 tuple `models.PayGroup.description` = `models.CharField(max_length=50, default="")` [static]

7.8.3.2 tuple `models.PayGroup.groupSize` = `models.IntegerField(default=1)` [static]

7.8.3.3 tuple `models.PayGroup.members` = `models.ManyToManyField(User)` [static]

7.8.3.4 tuple `models.PayGroup.memberViews` = `models.ManyToManyField(MemberView)` [static]

7.8.3.5 tuple `models.PayGroup.name` = `models.CharField(max_length=20, default="", unique=True)` [static]

7.8.3.6 tuple `models.PayGroup.passcode` = `models.CharField(max_length=16, default="")` [static]

7.8.3.7 tuple `models.PayGroup.paymentLogs` = `models.ManyToManyField(PaymentLog)` [static]

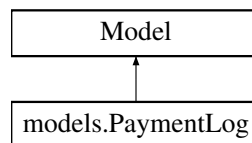
The documentation for this class was generated from the following file:

- [models.py](#)

7.9 models.PaymentLog Class Reference

Model designed to represent a transaction.

Inheritance diagram for `models.PaymentLog`:



Public Member Functions

- def `__unicode__`
Gets model's information in unicode string format.

Static Public Attributes

- tuple `amount` = `models.DecimalField(max_digits=11, decimal_places=2, default=0)`
- tuple `description` = `models.CharField(max_length=50, default="")`
- tuple `date` = `models.DateField(default=time.strftime("%Y-%m-%d"))`
- tuple `user` = `models.ForeignKey(User)`
- tuple `contested` = `models.BooleanField(default=False)`
- tuple `contestedMessage` = `models.CharField(max_length=140, default="")`

7.9.1 Detailed Description

Model designed to represent a transaction.

The [PaymentLog](#) model is used by a [PayUser](#) when he or she wants to report an Expense.

7.9.2 Member Function Documentation

7.9.2.1 `def models.PaymentLog.__unicode__(self)`

Gets model's information in unicode string format.

Returns

Description of the payment log

7.9.3 Member Data Documentation

7.9.3.1 `tuple models.PaymentLog.amount = models.DecimalField(max_digits=11, decimal_places=2, default=0)` `[static]`

7.9.3.2 `tuple models.PaymentLog.contested = models.BooleanField(default=False)` `[static]`

7.9.3.3 `tuple models.PaymentLog.contestedMessage = models.CharField(max_length=140, default="")` `[static]`

7.9.3.4 `tuple models.PaymentLog.date = models.DateField(default=time.strftime("%Y-%m-%d"))` `[static]`

7.9.3.5 `tuple models.PaymentLog.description = models.CharField(max_length=50, default="")` `[static]`

7.9.3.6 `tuple models.PaymentLog.user = models.ForeignKey(User)` `[static]`

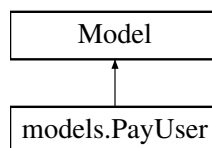
The documentation for this class was generated from the following file:

- [models.py](#)

7.10 `models.PayUser` Class Reference

A model for a user of ExpenShare.

Inheritance diagram for `models.PayUser`:



Public Member Functions

- `def __unicode__`

Gets model's information in unicode string format.

Static Public Attributes

- tuple `userKey` = `models.ForeignKey(User)`
- tuple `payGroups` = `models.ManyToManyField(PayGroup)`

7.10.1 Detailed Description

A model for a user of ExpenShare.

A [PayUser](#) can join PayGroups and make PaymentLogs.

A [PayUser](#) corresponds to both a User and a [PayGroup](#) and links them together.

7.10.2 Member Function Documentation

7.10.2.1 `def models.PayUser.__unicode__(self)`

Gets model's information in unicode string format.

Returns

The user's key

7.10.3 Member Data Documentation

7.10.3.1 `tuple models.PayUser.payGroups = models.ManyToManyField(PayGroup) [static]`

7.10.3.2 `tuple models.PayUser.userKey = models.ForeignKey(User) [static]`

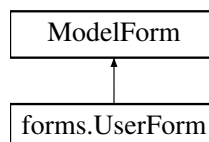
The documentation for this class was generated from the following file:

- [models.py](#)

7.11 forms.UserForm Class Reference

A form for creating a brand new ExpenShare User.

Inheritance diagram for forms.UserForm:



Classes

- class [Meta](#)
Meta defines corresponding model (User) and fields.

Static Public Attributes

- tuple `password = forms.CharField(widget=forms.PasswordInput())`

7.11.1 Detailed Description

A form for creating a brand new ExpenShare User.

[UserForm](#) requires a password, username, and email.

7.11.2 Member Data Documentation

7.11.2.1 `tuple forms.UserForm.password = forms.CharField(widget=forms.PasswordInput())` `[static]`

The documentation for this class was generated from the following file:

- [forms.py](#)

Chapter 8

File Documentation

8.1 forms.py File Reference

Django forms for taking input from ExpenShare users.

Classes

- class [forms.UserForm](#)
A form for creating a brand new ExpenShare User.
- class [forms.UserForm.Meta](#)
Meta defines corresponding model (User) and fields.
- class [forms.PayForm](#)
A form for entering an expense amount and description.
- class [forms.PayForm.Meta](#)
Meta defines corresponding model (PaymentLog) and fields.
- class [forms.MakeGroupForm](#)
A form for creating a new ExpenShare group.
- class [forms.MakeGroupForm.Meta](#)
Meta defines corresponding model (PayGroup) and fields.

Namespaces

- [forms](#)

8.1.1 Detailed Description

Django forms for taking input from ExpenShare users.

Authors

Taylor Andrews, Ian Char, Brennan McConnell

Date

11/26/2014

Note

Django Meta class defines non-field aspects of django forms.

8.2 mainpage.dox File Reference

8.3 models.py File Reference

Contains the database models implemented by ExpenShare.

Classes

- class [models.FellowUser](#)
Model designed to represent a fellow group member for a single particular [MemberView](#).
- class [models.MemberView](#)
The [MemberView](#) model is designed to represent a single group member's view of the expense balances.
- class [models.PaymentLog](#)
Model designed to represent a transaction.
- class [models.PayGroup](#)
Model for an ExpenShare group.
- class [models.PayUser](#)
A model for a user of ExpenShare.

Namespaces

- [models](#)

8.3.1 Detailed Description

Contains the database models implemented by ExpenShare.

Authors

Taylor Andrews, Ian Char, Brennan McConnell

Date

11/26/2014

Note

The **unicode** function allows the models to communicate with Django admin.

8.4 urls.py File Reference

URL map for ExpenShare.

Namespaces

- [urls](#)

Variables

- tuple [urls.urlpatterns](#)
Contains all the url tuples for the ExpenShare website.

8.4.1 Detailed Description

URL map for ExpenShare.

Authors

Taylor Andrews, Ian Char, Brennan McConnell

Date

11/25/2014

The urlpatterns uses regular expressions to match urls and pass them to views.

8.5 views.py File Reference

Contains the functions which implement all the logic behind ExpenShare.

Namespaces

- [views](#)

Functions

- def [views.index](#)
Returns the default ExpenShare page.
- def [views.home](#)
Displays the home page for the current User.
- def [views.history](#)
Displays the history webpage which shows all previous expenses for a specific group.
- def [views.register](#)
This function is used for registering new users.
- def [views.add_groupform](#)
This function creates a new PayGroup.
- def [views.add_payform](#)
Adds an expense to a group.
- def [views.joingroup_form](#)
Function which allows a User to join an existing PayGroup.
- def [views.leavegroup](#)
Function which makes a user leave a PayGroup upon request.
- def [views.userLogin](#)
Logs a user into their ExpenShare profile.
- def [views.userLogout](#)
Logs a User out.
- def [views.confirmPayment](#)
Confirms that a PayUser has received payment for existing expenses by a fellow member of the group.
- def [views.removePayForm](#)
Removes a payment from a group.

8.5.1 Detailed Description

Contains the functions which implement all the logic behind ExpenShare. Each view handles various http requests and returns appropriate responses.

Authors

Taylor Andrews, Ian Char, Brennan McConnell

Date

11/26/2014

Index

- `__unicode__`
 - `models::FellowUser`, [17](#)
 - `models::MemberView`, [19](#)
 - `models::PayGroup`, [22](#)
 - `models::PaymentLog`, [24](#)
 - `models::PayUser`, [25](#)
- `add_groupform`
 - `views`, [12](#)
- `add_payform`
 - `views`, [13](#)
- `amount`
 - `forms::PayForm`, [21](#)
 - `models::PaymentLog`, [24](#)
- `confirmPayment`
 - `views`, [13](#)
- `contested`
 - `models::PaymentLog`, [24](#)
- `contestedMessage`
 - `models::PaymentLog`, [24](#)
- `date`
 - `models::PaymentLog`, [24](#)
- `description`
 - `forms::MakeGroupForm`, [18](#)
 - `forms::PayForm`, [21](#)
 - `models::PayGroup`, [23](#)
 - `models::PaymentLog`, [24](#)
- `fellows`
 - `models::MemberView`, [19](#)
- `fields`
 - `forms::MakeGroupForm::Meta`, [21](#)
 - `forms::PayForm::Meta`, [20](#)
 - `forms::UserForm::Meta`, [20](#)
- `forms`, [11](#)
- `forms.MakeGroupForm`, [18](#)
- `forms.MakeGroupForm.Meta`, [20](#)
- `forms.PayForm`, [21](#)
- `forms.PayForm.Meta`, [20](#)
- `forms.py`, [27](#)
- `forms.UserForm`, [25](#)
- `forms.UserForm.Meta`, [20](#)
- `forms::MakeGroupForm`
 - `description`, [18](#)
 - `name`, [18](#)
 - `passcode`, [18](#)
- `forms::MakeGroupForm::Meta`
 - `fields`, [21](#)
 - `model`, [21](#)
- `forms::PayForm`
 - `amount`, [21](#)
 - `description`, [21](#)
- `forms::PayForm::Meta`
 - `fields`, [20](#)
 - `model`, [20](#)
- `forms::UserForm`
 - `password`, [26](#)
- `forms::UserForm::Meta`
 - `fields`, [20](#)
 - `model`, [20](#)
- `groupSize`
 - `models::PayGroup`, [23](#)
- `history`
 - `views`, [13](#)
- `home`
 - `views`, [13](#)
- `index`
 - `views`, [14](#)
- `joiningroup_form`
 - `views`, [14](#)
- `leavegroup`
 - `views`, [14](#)
- `mainpage.dox`, [28](#)
- `memberViews`
 - `models::PayGroup`, [23](#)
- `members`
 - `models::PayGroup`, [23](#)
- `model`
 - `forms::MakeGroupForm::Meta`, [21](#)
 - `forms::PayForm::Meta`, [20](#)
 - `forms::UserForm::Meta`, [20](#)
- `models`, [11](#)
- `models.FellowUser`, [17](#)
- `models.MemberView`, [19](#)
- `models.PayGroup`, [22](#)
- `models.PayUser`, [24](#)
- `models.PaymentLog`, [23](#)
- `models.py`, [28](#)
- `models::FellowUser`
 - `__unicode__`, [17](#)
 - `owed`, [18](#)
 - `user`, [18](#)
- `models::MemberView`

- [__unicode__](#), 19
 - [fellows](#), 19
 - [netOwed](#), 19
 - [user](#), 19
- [models::PayGroup](#)
 - [__unicode__](#), 22
 - [description](#), 23
 - [groupSize](#), 23
 - [memberViews](#), 23
 - [members](#), 23
 - [name](#), 23
 - [passcode](#), 23
 - [paymentLogs](#), 23
- [models::PayUser](#)
 - [__unicode__](#), 25
 - [payGroups](#), 25
 - [userKey](#), 25
- [models::PaymentLog](#)
 - [__unicode__](#), 24
 - [amount](#), 24
 - [contested](#), 24
 - [contestedMessage](#), 24
 - [date](#), 24
 - [description](#), 24
 - [user](#), 24
- [name](#)
 - [forms::MakeGroupForm](#), 18
 - [models::PayGroup](#), 23
- [netOwed](#)
 - [models::MemberView](#), 19
- [owed](#)
 - [models::FellowUser](#), 18
- [passcode](#)
 - [forms::MakeGroupForm](#), 18
 - [models::PayGroup](#), 23
- [password](#)
 - [forms::UserForm](#), 26
- [payGroups](#)
 - [models::PayUser](#), 25
- [paymentLogs](#)
 - [models::PayGroup](#), 23
- [register](#)
 - [views](#), 14
- [removePayForm](#)
 - [views](#), 15
- [urlpatterns](#)
 - [urls](#), 12
- [urls](#), 11
 - [urlpatterns](#), 12
- [urls.py](#), 28
- [user](#)
 - [models::FellowUser](#), 18
 - [models::MemberView](#), 19
 - [models::PaymentLog](#), 24
 - [userKey](#)
 - [models::PayUser](#), 25
 - [userLogin](#)
 - [views](#), 15
 - [userLogout](#)
 - [views](#), 15
 - [views](#), 12
 - [add_groupform](#), 12
 - [add_payform](#), 13
 - [confirmPayment](#), 13
 - [history](#), 13
 - [home](#), 13
 - [index](#), 14
 - [joining_group_form](#), 14
 - [leavegroup](#), 14
 - [register](#), 14
 - [removePayForm](#), 15
 - [userLogin](#), 15
 - [userLogout](#), 15
- [views.py](#), 29