Django-ResaX Documentation

Version 1.4.1

Alexandre Syenchuk, Antoine Bordeau

mars 21, 2017

Table des matières

Contents :

CHAPITRE 1

resax package

Submodules

resax.models module

class resax.models.**AbstractActivity**(**args*, ***kwargs*)

Activité de l'organisation.

deleted

Drapeau indiquant que l'activité est supprimée

get_events_of_the_day(*date=None*)

Retourne tous les évènements correspondants à cette activité pour la date donnée, ou la date actuelle.

Paramètres **date** (*datetime*) – date du jour

Type retourné QuerySet

name

Nom de l'activité

organisation

L'organisation à laquelle appartient l'activité

resources

Liste des ressources nécessaires pour l'activité

stock

Nombre de places disponibles pour cette activité

class resax.models.**AbstractActivityResource**(**args*, ***kwargs*)

Ressource d'activité.

activity

Activité correspondante

quantity

Quantité de la ressource requise pour l'activité ; la valeur spéciale -1 consomme la totalité de la ressource

resource

Ressource requise

class resax.models.**AbstractEvent**(**args*, ***kwargs*)
Représentation d'un évènement dans le calendrier.

activity
Activité associée à cet évènement (faculatif)

book(*user*, *quantity=1*)
Réserve cet évènement pour un utilisateur.

Paramètres
—**user** (User) – utilisateur à associer à la réservation
—**quantity** (*int*) – nombre de places à réserver

Type retourné*Reservation*

date_start
Date et heure de début de l'évènement

date_stop
Date et heure de fin de l'évènement

planning
Planning associé à cet évènement (facultatif)

set_stock(*new_stock*)
Redéfinit le nombre de places disponibles pour l'évènement.

Paramètres**new_stock** (*int*) – nombre de places disponibles pour l'évènement ; 0 si illimité

stock
Nombre de réservations possibles pour cet évènement. 0 signifie réservations illimitées

class resax.models.**AbstractFlexiReservation**(**args*, ***kwargs*)
Réservation flexible. Une réservation est dite "flexible" lorsqu'elle ne vient pas se greffer sur un évènement existant, mais lorsqu'elle crée elle-même un évènement. Cet évènement n'est alors pas associé à une activité ; les ressouces réservées sont directement spécifiées par l'utilisateur.

event
L'évènement créé pour honorer cette réservation

reservation_type
Type de réservation

resources
Ressources réservées, avec les quantités requises

user
L'utilisateur ayant fait la réservation

class resax.models.**AbstractFlexiReservationResource**(**args*, ***kwargs*)
Ressource réservée par une réservation dite "flexible".

flexi_reservation
Réservation "flexible" associée

quantity
Quantité de la ressource requises

resource
Ressource réservée

class resax.models.**AbstractOrganisation**(**args*, ***kwargs*)
Represents an organisation using the API.

add_activity(*name*, *stock=1*, *resources=None*)
Ajoute une activité à l'organisation.

Paramètres
—**name** (*str*) – nom de l'activité
—**stock** (*int*) – nombre de places disponibles pour l'activité

—**resources** (*dict*) – dictionnaire facultatif contenant les ressources à ajouter à cette activité. Chaque clé doit correspondre à une ressource, et chaque valeur doit indiquer la quantité requise pour l'activité

Type retourné*Activity*

add_reservation_type(*name*, *resources=None*)
Ajoute un type de réservation à l'organisation. Si l'argument *resources* est spécifié, il doit être une liste contenant les ressources autorisées pour ce type de réservation.

Paramètres
—**name** (*str*) – nom du type de réservation
—**resources** (*list*) – liste facultative contenant les ressources autorisées pour ce type de réservation

Type retourné*ReservationType*

add_resource_type(*name*, *resources=None*)
Ajoute un nouveau type de ressource à l'organisation. Si l'argument *resources* est spécifié, il doit être un dictionnaire ayant pour clé le nom de la ressource et pour valeur sa quantité en stock.

Paramètres
—**name** (*str*) – nom du type de ressource
—**resources** (*dict*) – dictionnaire facultatif contenant les ressources à créer de ce type. Chaque clé doit correspondre au nom de la ressource, et chaque valeur doit indiquer la quantité disponible en stock

Type retourné*ResourceType*

add_user()
Adds a new member (user) to the organisation

Type retourné*User*

deleted
Flag indicating if the organisation was deleted

name
Name of the organisation

class resax.models.**AbstractPlanning**(**args*, ***kwargs*)
Planning de l'activité.

activity
Activité planifiée

date_stop
Date de fin de l'activité

on_day0
Jours de la semaine programmés pour l'activité

time_start
Date et heure de début du premier évènement planifié

time_stop
Date et heure de fin du premier évènement planifié

class resax.models.**AbstractReservation**(**args*, ***kwargs*)
Représente une réservation pour un évènement.

event
L'évènement réservé

quantity
Nombre de places réservées

user
L'utilisateur ayant réservé l'évènement

class resax.models.**AbstractReservationType**(**args*, ***kwargs*)
Type de réservation. Un type de réservation permet à un utilisateur de réserver les ressources autorisées.

name
Nom du type de réservation

organisation
L'organisation à laquelle appartient ce type de réservation

resources
Liste des ressources autorisées pour ce type de réservation

class resax.models.**AbstractResource**(**args*, ***kwargs*)
Représente une ressource.

deleted
Drapeau indiquant que la ressource est supprimée

get_available_stock(*date_start*, *date_stop*, *exclude_event=None*)
Retour la quantité disponible de la ressource sur la période entre les dates *date_start* et *date_stop* spécifiées.
Si *exclude_event* est spécifié, cet évènement n'est pas pris en compte dans les résultats.

Paramètres
—**date_start** – date de début de disponibilité recherchée pour la ressource
—**date_stop** – date de fin de disponibilité recherchée pour la ressource
—**exclude_event** (Event) – évènement facultatif à ne pas prendre en compte pour le caclul des résultats

Type retournéint

name
Nom de la ressource

resource_type
Type de la ressource

set_stock(*new_stock*)
Redéfinit la quantité en stock de la ressource.

Paramètres**new_stock** (*int*) – quantité en stock de la ressource ; 0 si illimitée

stock
Sotck disponible pour la ressource. 0 signifie que la stock est illimité.

class resax.models.**AbstractResourceType**(**args*, ***kwargs*)
Représente un type de ressource.

deleted
Drapeau indiquant que le type de ressource est supprimé

name
Nom du type de ressource

organisation
Organisation à laquelle appartient le type de ressource

class resax.models.**AbstractUser**(**args*, ***kwargs*)
Représente un utilisateur (membre) d'une organisation.

book_event(*event*, *quantity=1*)
Réserve *quantity* places pour l'évènement *event*.

Paramètres
—**event** (Event) – évènement à réserver
—**quantity** (*int*) – nombre de places à réserver

Type retourné*Reservation*

book_resources(*reservation_type*, *date_start*, *date_stop*, *resources=None*)
Crée une réservation flexible.

Paramètres

—**reservation_type** (ReservationType) – type de réservation
—**date_start** – date de début de la réservation
—**date_stop** – date de fin de la réservation
—**resources** (*dict*) – dictionnaire facultatif contenant les ressources à réserver. Ce dictionnaire doit être composé d'un ensemble de clés et de valeurs, où chaque clé réprésente la ressource à réserver, et chaque valeur la quantité demandée

Type retourné*FlexiReservation*

events
Liste des évènements réservés par l'utilisateur

organisation
L'organisation à laquelle appartient l'utilisateur

class resax.models.**Activity**(*id*, *organisation*, *name*, *stock*, *deleted*)

class resax.models.**ActivityResource**(*id*, *resource*, *activity*, *quantity*)

class resax.models.**Event**(*id*, *activity*, *planning*, *date_start*, *date_stop*, *stock*)

class resax.models.**FlexiReservation**(*id*, *user*, *reservation_type*, *event*)

class resax.models.**FlexiReservationResource**(*id*, *flexi_reservation*, *resource*, *quantity*)

class resax.models.**Organisation**(*id*, *name*, *deleted*)

class resax.models.**Planning**(*id*, *activity*, *on_day0*, *on_day1*, *on_day2*, *on_day3*, *on_day4*, *on_day5*, *on_day6*, *time_start*, *time_stop*, *date_stop*)

class resax.models.**Reservation**(*id*, *event*, *user*, *quantity*)

class resax.models.**ReservationType**(*id*, *organisation*, *name*)

class resax.models.**Resource**(*id*, *resource_type*, *name*, *stock*, *deleted*)

class resax.models.**ResourceType**(*id*, *organisation*, *name*, *deleted*)

class resax.models.**User**(*id*, *organisation*)

CHAPITRE 2

Indices and tables

— genindex
— modindex
— search

Index des modules Python

r

Index