Flask - Flask-Admin - One-To-Many - Cascade

I'm currently starting a project with Flask. I'm currently experiencing Flask-Admin.

I try to setup a 2 levels relationship. For each "Candidat" I like to associate a language and a level (LanguageLevel) for this language.

Here is my models.py

```
from app import db

class Candidat(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    firstname = db.Column(db.String(128))
    lastname = db.Column(db.String(128))
    birthdate = db.Column(db.DateTime)
    languages = db.relationship("CandidatLanguage", backref="candidat")

    def __repr__(self):
        return '<Nom %r>' % self.lastname


class Language(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(128))

    def __repr__(self):
        return '<Langues %r>' % self.name

class Languagelevel(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(128))

    def __repr__(self):
        return '<Niveau : %r>' % self.name

class CandidatLanguage(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    candidat_id = db.Column(db.Integer, db.ForeignKey('candidat.id'))
    language_id = db.Column(db.Integer, db.ForeignKey('language.id'))
    langguage_level_id = db.Column(db.Integer, db.ForeignKey('languagelevel.id'))
```

Here is my views.py

```
from models import Candidat, Languagelevel, Language, CandidatLanguage

from flask.ext.admin import Admin, BaseView, expose
class MyView(BaseView):
    @expose('/')
    def index(self):
        return self.render('adm-index.html')

admin = Admin(app)
admin.add_view(MyView(name='Hello'))

from flask.ext.admin.contrib.sqla import ModelView

admin.add_view(ModelView(Candidat, db.session))
admin.add_view(ModelView(CandidatLanguage, db.session))
admin.add_view(ModelView(Language, db.session))
admin.add_view(ModelView(Languagelevel, db.session))
```

What I would like to have is a Candidat form wich looks like this:

Firstname : ____

Lastname : _____

Language :

- DropDown for Language :
  - Language 1
  - Language 2
- DropDown for LanguageLevel :
  - Level 1
  - Level 2

For one "Candidat", one Language can only be associated with one LanguageLevel.

I've tried `column_auto_select_related` , `column_display_all_relations` , and of course none of them works like I thought. Of course this is my mistake.

If someone can show me the right way for reaching my goal it would be really helpfull.

Thanks in advance.

Regards

python flask flask-sqlalchemy flask-wtforms flask-admin

edited Feb 23 '14 at 18:11

asked Feb 23 '14 at 12:26

Youpsla
29 9

1 Answer

Finally, I've found my solution in the following post

My mistakes were:

- putting a `db.relationship` (CandidatLanguage) in my `Candidat` class. Instead of that, I put relations in my "pivot" table (CandidatLanguage).
- as IfLoop said in the post, I have to "switch from a plain many-to-many relationship to an "Association Object". I don" understood why but it works using Association object.

Then my models.py now looks like this:

```
class Candidat(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    firstname = db.Column(db.String(128))
    lastname = db.Column(db.String(128))
    birthdate = db.Column(db.Date)
    categories = db.relationship('Category', secondary=category_candidat,
                                 backref=db.backref('candidat', lazy='dynamic'))

    def __repr__(self):
        return '<Nom %r>' % self.lastname


class Language(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(128))

    def __repr__(self):
        return '%s' % unicode(self.name)


class Languagelevel(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(128))
    def __repr__(self):
        return '%s' % self.name

class CandidatLanguage(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    candidat_id = db.Column(db.Integer, db.ForeignKey('candidat.id'))
    language_id = db.Column(db.Integer, db.ForeignKey('language.id'))
    language_level_id = db.Column(db.Integer, db.ForeignKey('languagelevel.id'))

    language = db.relationship(Language, backref="Candidat")
    candidat = db.relationship(Candidat, backref="Langue")
    languagelevel = db.relationship(Languagelevel, backref="Candidat")
```

And my views.py like that:

```
from flask.ext.admin import Admin, BaseView, expose
class MyView(BaseView):
    @expose('/')
    def index(self):
        return self.render('adm-index.html')

admin = Admin(app)
admin.add_view(MyView(name='Hello'))

from flask.ext.admin.contrib.sqla import ModelView

class CandidatView(ModelView):
    column_auto_select_related = True
    inline_models = (CandidatLanguage,)

admin.add_view(CandidatView(Candidat, db.session))
admin.add_view(ModelView(Language, db.session))
admin.add_view(ModelView(Languagelevel, db.session))
```

With that, everything works fine and I have an inline form like I want.

I haven't understood everything but it works. if somebody can explain, it's of coruse welcome.

Regards

answered Feb 24 '14 at 12:38

Youpsla
29 9