

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

Join the Stack Overflow community to:



Ask programming questions



Answer and help your peers



Get recognized for your expertise

Flask-Migrate “No section: 'alembic'” on “db migrate” command

Maßgeschneidertes Agentur-Hosting?
Du bestimmst, wo es langgeht!



I'm using Flask-Migrate, and I'm trying to add a column to my database by using the db migrate command. Here is the setup:

```
from flask import Flask, render_template, request, session, flash, redirect, url_for
from flask.ext.babel import Babel
from flask.ext.mail import Mail
from flask.ext.script import Manager
from flask.ext.migrate import Migrate, MigrateCommand
from flask.ext.sqlalchemy import SQLAlchemy
from flask.ext.security import Security, SQLAlchemyUserDatastore, \
    UserMixin, RoleMixin, login_required, roles_required
from flask.ext.login import current_user
from forms import UserEditForm
import os, sys, datetime, random, math
basedir = os.path.abspath(os.path.dirname(__file__))

# Create app
application = Flask(__name__)
application.config['DEBUG'] = True
application.config['SECRET_KEY'] = 'super-secret'
application.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:/// ' + os.path.join(basedir, 'app.db')

application.config['DEFAULT_MAIL_SENDER'] = 'info@site.com'
application.config['SECURITY_REGISTERABLE'] = True
application.config['SECURITY_CONFIRMABLE'] = True
application.config['SECURITY_RECOVERABLE'] = True

application.config.from_object('config.email')

application.config['DBPATH'] = os.path.join(basedir, 'app.db')

# Setup mail extension
mail = Mail(application)

# Setup babel
babel = Babel(application)

@babel.localeselector
def get_locale():
    override = request.args.get('lang')

    if override:
        session['lang'] = override

    rv = session.get('lang', 'en')
    return rv

# Create database connection object
db = SQLAlchemy(application)
migrate = Migrate(application, db)

manager = Manager(application)
manager.add_command('db', MigrateCommand)

# Define models
roles_users = db.Table('roles_users',
    db.Column('user_id', db.Integer(), db.ForeignKey('user.id')),
    db.Column('role_id', db.Integer(), db.ForeignKey('role.id')))

class Role(db.Model, RoleMixin):
    id = db.Column(db.Integer(), primary_key=True)
    name = db.Column(db.String(80), unique=True)
    description = db.Column(db.String(255))

class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(255), unique=True)
    password = db.Column(db.String(255))
    active = db.Column(db.Boolean())
```

```

confirmed_at = db.Column(db.DateTime())
nickname = db.Column(db.String(255))
favcolor = db.Column(db.String(255))
favshape = db.Column(db.String(255))
favflower = db.Column(db.String(255))
favband = db.Column(db.String(255)) # <<<--- NEW COLUMN

roles = db.relationship('Role', secondary=roles_users,
                        backref=db.backref('users', lazy='dynamic'))

def __str__(self):
    return '<User id=%s email=%s>' % (self.id, self.email)

user_datastore = SQLAlchemyUserDatastore(db, User, Role)
security = Security(application, user_datastore)

# Views
@app.route('/')
def home():
    return render_template('index.html')

if __name__ == '__main__':
    if str(sys.argv[1]) == 'run':
        application.run(debug=True) #(host='0.0.0.0', debug=True)
    else:
        manager.run()

```

I am trying to add the new column `favband` using the following commands:

```
flask\Scripts\python.exe application.py db migrate flask\Scripts\python.exe application.py db upgrade
```

However, when trying `db migrate`, I get the following error:

```

Traceback (most recent call last):
  File "application.py", line 398, in <module>
    manager.run() #1) db migrate, 2) db upgrade
  File "C:\website\flask\lib\site-packages\flask_script\__init__.py", line 412, in
run
    result = self.handle(sys.argv[0], sys.argv[1:])
  File "C:\website\flask\lib\site-packages\flask_script\__init__.py", line 383, in
handle
    res = handle(*args, **config)
  File "C:\website\flask\lib\site-packages\flask_script\commands.py", line 216, in
__call__
    return self.run(*args, **kwargs)
  File "C:\website\flask\lib\site-packages\flask_migrate\__init__.py", line 132, in
migrate
    config = _get_config(directory)
  File "C:\website\flask\lib\site-packages\flask_migrate\__init__.py", line 46, in
_get_config
    config.set_main_option('script_location', directory)
  File "C:\website\flask\lib\site-packages\alembic\config.py", line 198, in
set_main_option
    self.file_config.set(self.config_ini_section, name, value)
  File "C:\Python27\Lib\ConfigParser.py", line 701, in set
    ConfigParser.set(self, section, option, value)
  File "C:\Python27\Lib\ConfigParser.py", line 388, in set
    raise NoSectionError(section)
ConfigParser.NoSectionError: No section: 'alembic'

```

I have also tried this from within `flask\Scripts`, but get the same error:

```
python.exe ../../application.py db migrate
```

I have also tried upgrading flask-migrate to the latest version.

`python` `flask` `flask-sqlalchemy` `alembic` `flask-migrate`

asked May 5 '15 at 15:13

 [b_g](#)
68 ● 7

Do you have an alembic.ini file? – [dirm](#) May 5 '15 at 16:16

2 Answers

Just in case anyone has this problem in the future--

I didn't have the migrations folder because I had transferred the DB file from a different project. So, this is what happens when you have no functioning migrations folder!

answered May 5 '15 at 16:55

 [b_g](#)
68 ● 7

I was also having this problem. I had the migrations folder in a different than expected folder. Normally, you can just do the following and it will appropriately find the directory:

```
migrate = Migrate(app, db, directory="/path/to/migrations_folder")
```

However if you instantiate the Migrate object and then later on initialize with the db and application it will not work:

```
# This won't work  
migrate = Migrate(directory='/path/to/migrations_folder')  
migrate.init_app(app, db)
```

`init_app` overrides the directory back to the default. Therefore you will need to always pass the directory in the `init_app` call.

```
# This will  
migrate = Migrate()  
migrate.init_app(app, db, directory='/path/to/migrations_folder')
```

EDIT I submitted a [fix](#)

edited Aug 28 '15 at 3:31

answered Aug 27 '15 at 21:19



[Tim Martin](#)
178 ● 9