دانشگاه لرستان

به نام خدا

تمرین اول درس پایگاه داده پیشرفته

دانشجو:

پریسا مبارک

شماره دانشجویی:

۴۰۲۱۱۴۱۵۰۰۶

استاد مربوطه:

دکتر عبدالرضا رشنو

آذر ۱۴۰۲

شرح کار:

سه جدولstudent وteacher وlesson داریم. میخواهیم با استفاده از پایگاه داده sqlie و fastapi مدل ها را پیاده سازی کنیم

در فایل models.py:

مطابق با فایل های ضمیمه شده مدل ها را پیاده سازی کردم.دراین فایل ۴مدل studentوteacher وlesson داریم که هرکدام فیلدهای مخصوص خود را دارند.

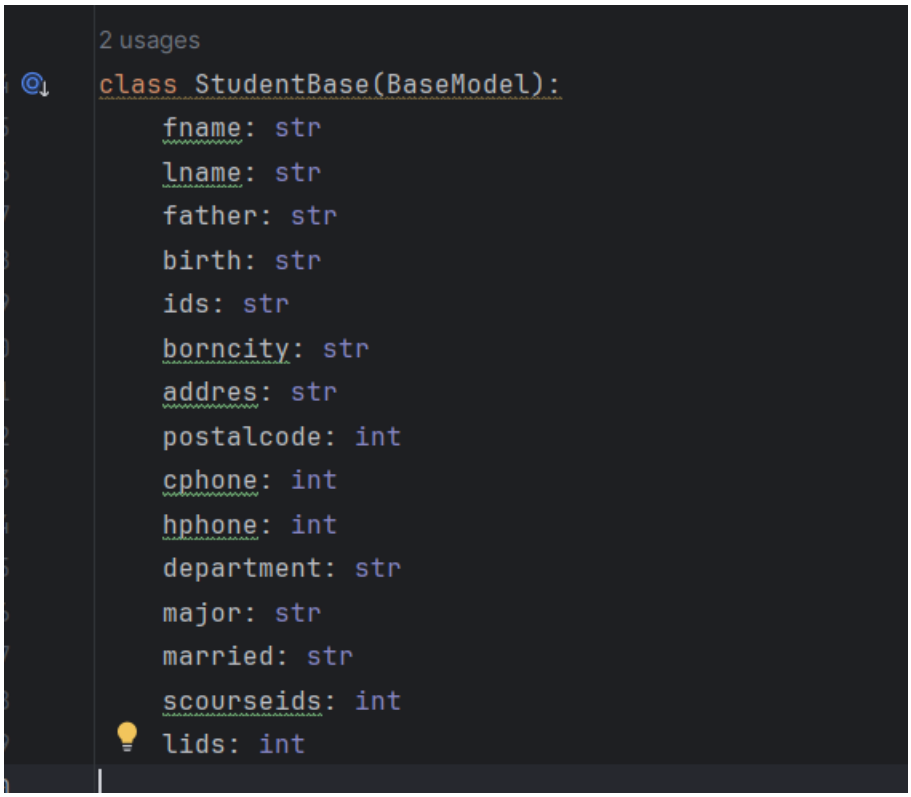
API test_main.http | main.py | database.py | schemas.py | models.py

```python
from database import Base
from sqlalchemy import Column, Integer, String

9 usages
class Student(Base):
    __tablename__ = 'student'

    stid = Column(Integer, primary_key=True, index=True)
    fname = Column(String)
    lname = Column(String)
    father = Column(String)
    birth = Column(String)
    ids = Column(String, unique=True)
    borncity = Column(String)
    addres = Column(String)
    postalcode = Column(Integer)
    cphone = Column(Integer)
    hphone = Column(Integer)
    department = Column(String)
    major = Column(String)
    married = Column(String)
    id = Column(Integer, unique=True)
    scourseids = Column(Integer)
    lids = Column(Integer)
9 usages
class Teacher(Base):
    __tablename__ = 'teacher'

    lid = Column(Integer, primary_key=True, index=True)
    fname = Column(String)
```

شکل ۱. یک نمونه از پیاده سازی مدلstudent در فایل models.py

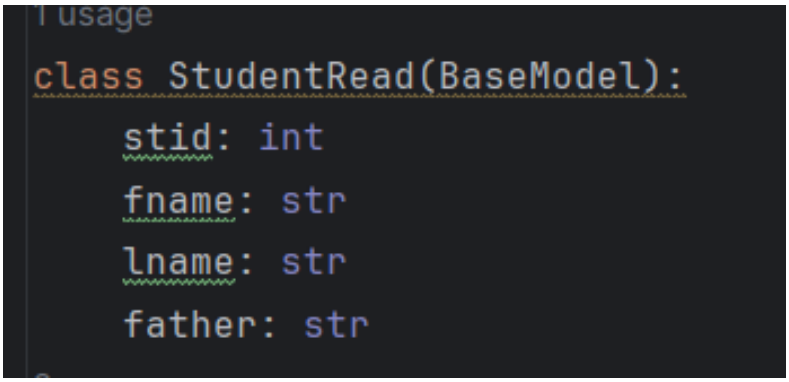
در فایلschemas.py:

کلاس های مربوط به هر مدل را نیز پیاده سازی کردم.



شکل۲. یک کلاس برای جدول دانشجو

در ادامه برای استفاده از responsive model ها یک سری فیلد هارا در کلاس های جداگانه نیز گذاشتم(مانند کلاس ReadStu) که در ادامه کاربرد آن ها را بررسی میکنیم.



شکل۳. ایجاد یک کلاس برای استفاده از responsv model ها در جدول دانشجو

همچنین در این فایل اعتبارسنجی های مربوط به فیلدهای هرجدول را انجام دادم.

به عنوان مثال برای فیلد birth در جدولstudent میخواهم مطابق با الگوی منظم تاریخ شمسی باشد:

```python
@validator('birth')
def validate_birth(cls, birth):
    date_pattern = "^1[3-4][0-9]{2}/((0[0-9])|(1[0-2]))/(([0-2][0-9])|3[0-1])$"
    match = re.match(date_pattern, birth)
    if not match:
        raise ValueError("تاریخ تولد نامعتبر است.")
    return birth
```

شکل۴. یک نمونه اعتبارسنجی در جدول دانشجو

در فایلdatabase.py:

دراین فایل به پایگاه دادهsqlite متصل میشویم.

test_main.http | main.py | database.py | schemas.py | models.py

```python
from sqlalchemy import create_engine
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker
SQLALCHEMY_DATABASE_URL = 'sqlite:///./test.db'
engine = create_engine(SQLALCHEMY_DATABASE_URL, connect_args={'check_same_thread': False})
SessionLocal = sessionmaker(bind=engine)
Base = declarative_base()
```

شکل۵ . محتوای فایل database.py

در فایلmain.py:

در این فایل برای عملیات درج و حذف و آپدیت و خواندن در هر جدول یکapi طراحی میکنیم.

به عنوان مثال برای جدولstudent:

t_main.http | main.py | database.py | schemas.py | models.py

```python
from fastapi import FastAPI,Depends, HTTPException
from sqlalchemy.orm import Session
from database import engine,SessionLocal
import schemas,models

models.Base.metadata.create_all(bind=engine)
app = FastAPI()

12 usages
def get_db():
    db = SessionLocal()
    try:
        yield db
    finally:
        db.close()

@app.post( path: '/RegStu/',response_model=schemas.Student)
def create_student(student: schemas.StudentCreate, db:Session=Depends(get_db)):
    db_student=db.query(models.Student).filter(models.Student.ids ==student.ids).first()
    if db_student:
        raise HTTPException(status_code=400,detail='already exists')
    student=models.Student(fname=student.fname,lname=student.lname,father=student.father,birth
    db.add(student)
    db.commit()
    db.refresh(student)
    return student
```

شکل ۶. یکapi برای عمل درج رکورد در جدول دانشجو

```python

@app.get( path: '/RegStu/{student_stid}',response_model=schemas.StudentRead)
def read_student(student_stid:int,db:Session=Depends(get_db)):
    db_student=db.query(models.Student).filter(models.Student.stid ==student_stid).first()
    if db_student is None:
        raise HTTPException(status_code=404, detail='STUDENT NOT FOUND')
    return db_student



@app.delete('/DelStu/{student_stid}')
def delete_student(student_stid: int, db: Session = Depends(get_db)):
    db_student = db.query(models.Student).filter(models.Student.stid == student_stid).first()
    if db_student is None:
        raise HTTPException(status_code=404, detail='STUDENT NOT FOUND')
    db.delete(db_student)
    db.commit()
    return {"message": f"Student with stid {student_stid} has been deleted."}

```

شکل۷. یکapi برای عمل خواندن رکورد و یک api برای عمل حذف رکورد از جدول دانشجو

```python
@app.post( path: '/UpStu/{student_stid}', response_model=schemas.Student)
def update_student(student_stid: int, student: schemas.StudentCreate, db: Session = Depends(get_db)):
    db_student = db.query(models.Student).filter(models.Student.stid == student_stid).first()
    if db_student is None:
        raise HTTPException(status_code=404, detail="Student not found")

    # تنظیم مستقیم فیلدها از اطلاعات جدید دانشجو
    db_student.fname = student.fname
    db_student.lname = student.lname
    db_student.father = student.father
    db_student.birth = student.birth
    db_student.ids = student.ids
    db_student.borncity = student.borncity
    db_student.addres = student.addres
    db_student.postalcode = student.postalcode
    db_student.cphone = student.cphone
    db_student.hphone = student.hphone
    db_student.department = student.department
    db_student.major = student.major
    db_student.married = student.married
    db_student.scourseids = student.scourseids
    db_student.lids = student.lids
    db_student.id = student.id

    # ذخیره تغییرات در پایگاه داده
    db.commit()

    # بازخوانی دانشجو برای به‌روزرسانی اطلاعات
    db.refresh(db_student)
```

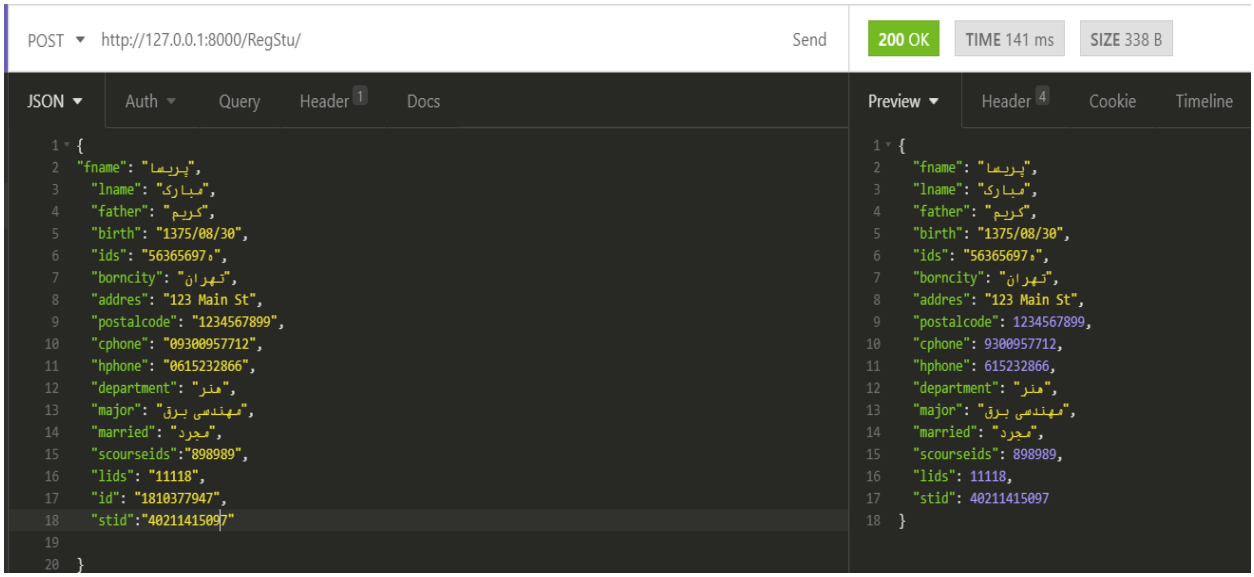
شکل۸. یک api برای عمل آپدیت رکورد در جدول دانشجو

در فایل های ضمیمه شده تمام کدها به طور کامل برای هر جدول وجود دارد.که به دلیل طولانی بودن تنها خلاصه ای از آن ها را در بالا نشان دادم.

تست api ها:

در اینجا برای تست api ها از ابزار insomnia استفاده میکنیم:

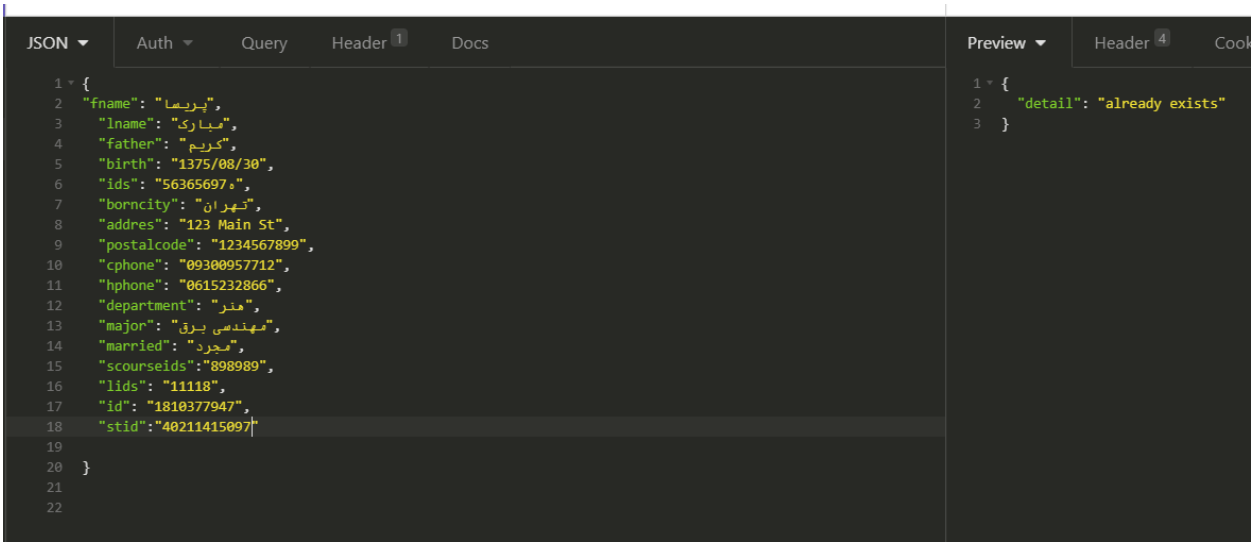
درج رکورد در جدول دانشجو:



شکل۹. درج رکورد در جدول دانشجو

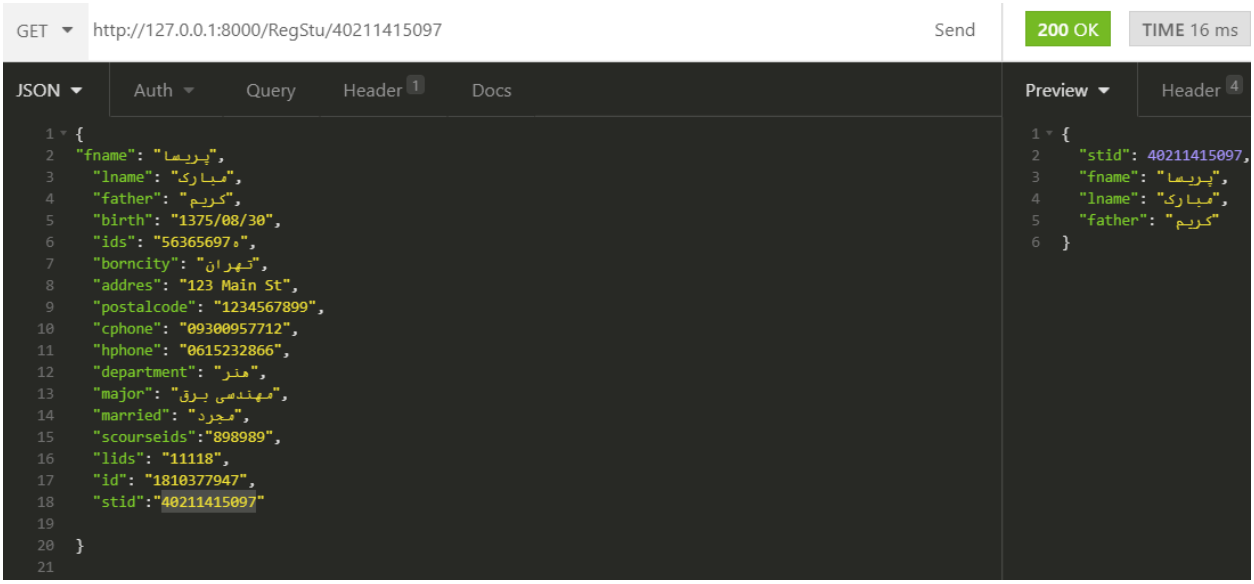
همانطور که مشاهده میکنیم با متدpost واز طریقurl نوشته شده توانستم یک رکورد به جدول دانشجو اضافه کنم.

حال اگر بخواهیم دوباره این رکورد را وارد کنیم به ما میگوید رکورد وجود دارد:



شکل۱۰. درج رکورد تکراری در جدول دانشجو

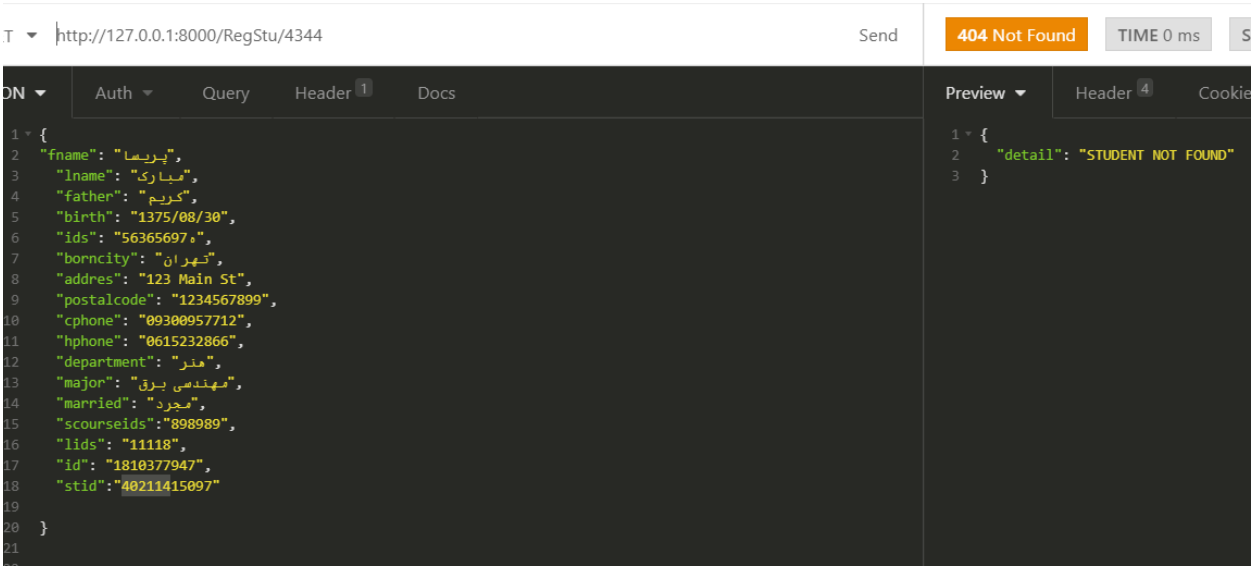
خواندن رکورد از جدول دانشجو:



شکل۱۱. خواندن رکورد از جدول دانشجو

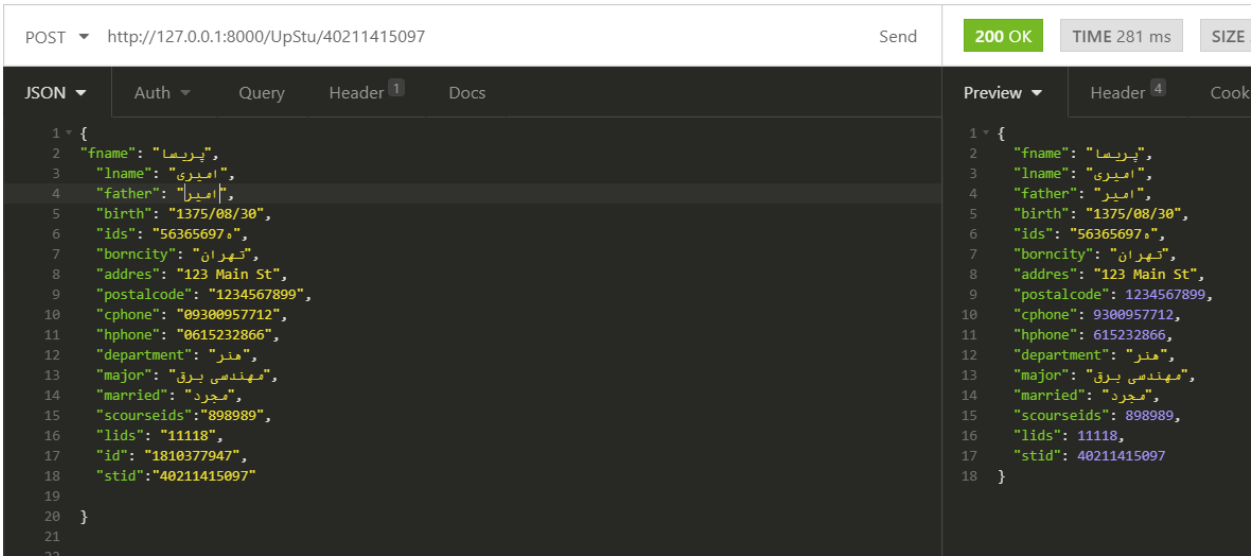
مطابق تصویر بالا با استفاده از متدget واز طریقurl بالا و از طریق شماره stid که در آخرurl نوشتم، توانستم یک رکورد را بخوانم و ۴ فیلد اول جدول دانشجو را به عنوان خروجی گرفتم.

اگر بخواهیم رکوردی که وجود ندارد را بخوانیم به ما میگوید دانشجو پیدا نمیشود:



شکل۱۲. خواندن رکوردی که در جدول دانشجو وجود ندارد

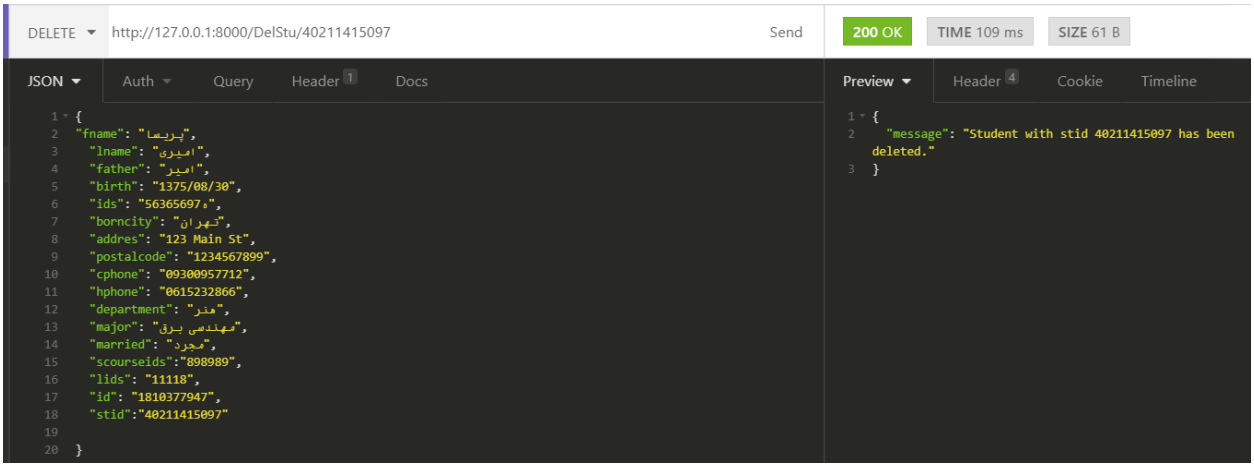
آپدیت رکورد در جدول دانشجو:



شکل۱۳. آپدیت یک رکورد در جدول دانشجو

طبق تصویر بالا فیلدهایlname وfather رکورد مدنظر را آپدیت کردم.

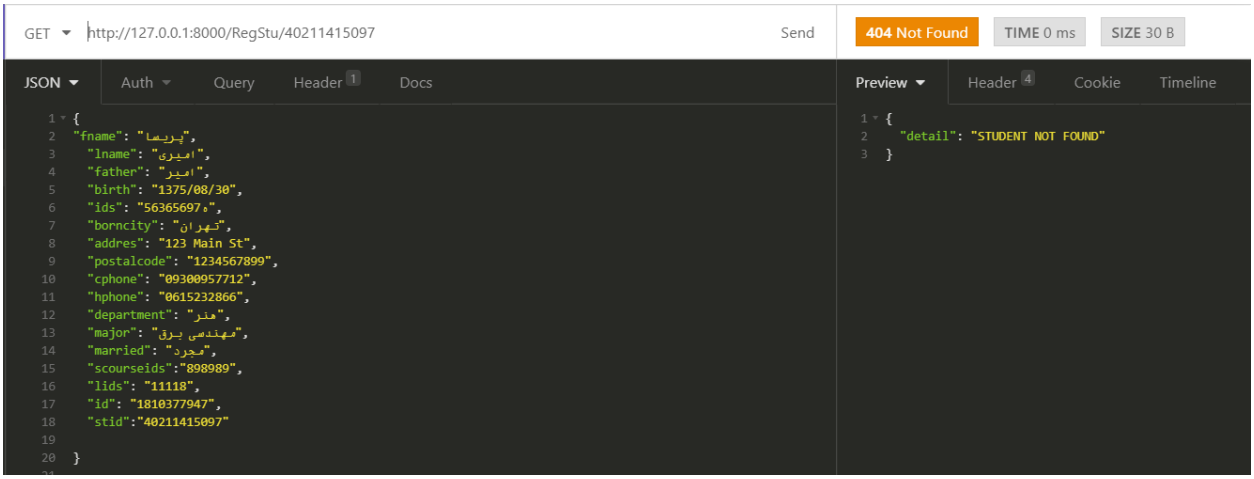
حذف رکورد از جدول دانشجو:



شکل۱۴. حذف یک رکورد از جدول دانشجو

طبق تصویر بالا رکورد مدنظر را از طریق متدdelete واز طریقurl بالا حذف کردم.

حال اگر بخواهم رکورد را بخوانم چون رکورد حذف شده است خطا میدهد.مطابق تصویر زیر:

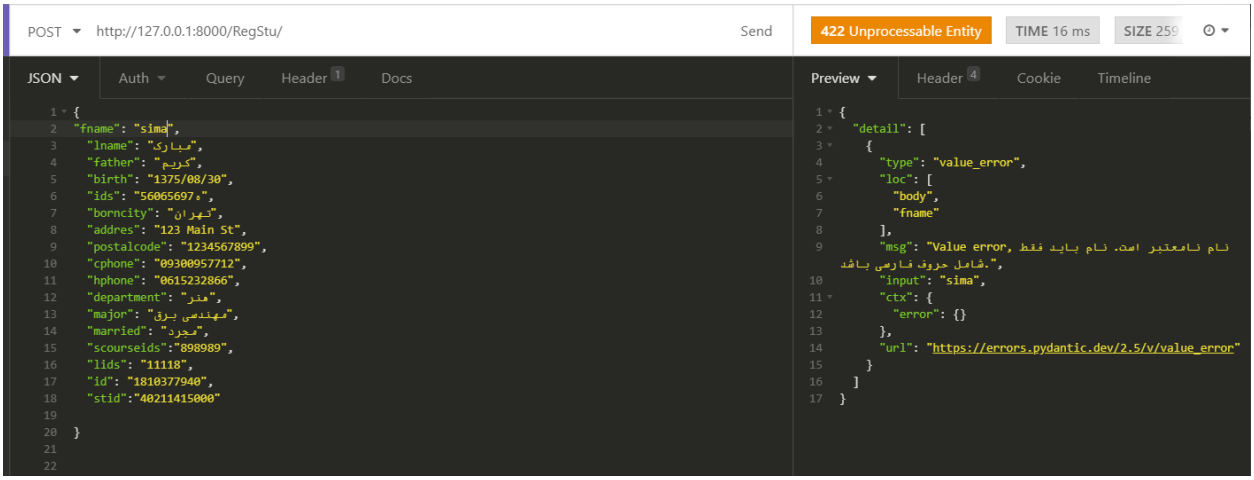


شکل۱۵. خواندن رکوردی که حذف شده است از جدول دانشجو

همانطور که مشاده میکنیم در خروجی به ما میگوید که دانشجو وجود ندارد.

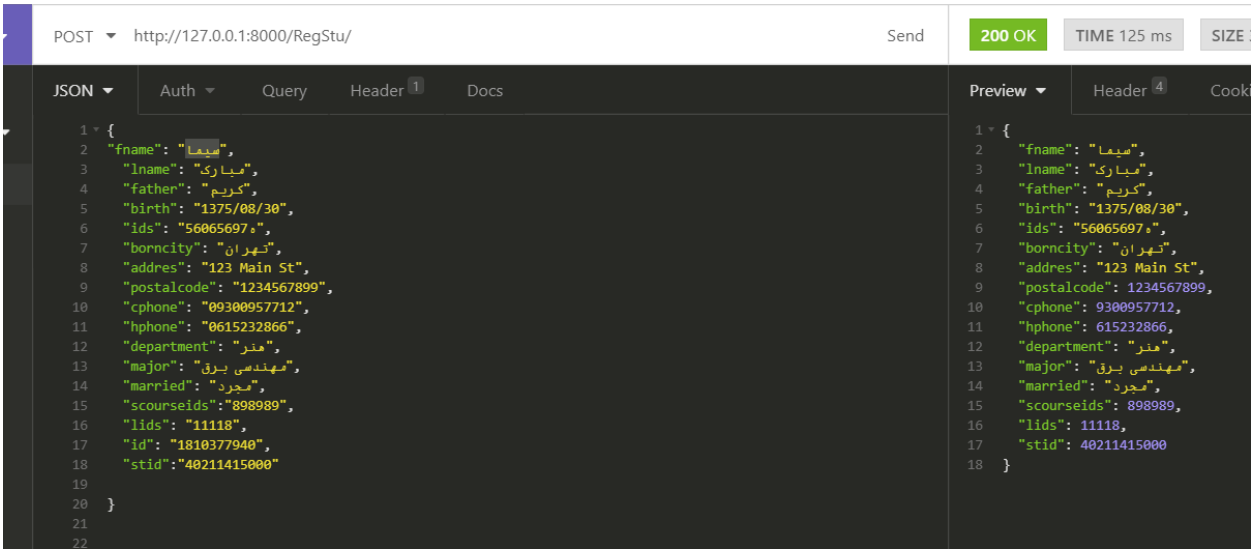
اعتبارسنجی فیلدها در جدول دانشجو:

همانطور که گفتم در فایلschemas.py فیلدها اعتبارسنجی شده اند که نمونه ای ازآن ها را در ادامه مشاهده میکنید. به عنوان مثال برای فیلدهای fname,lname,father:



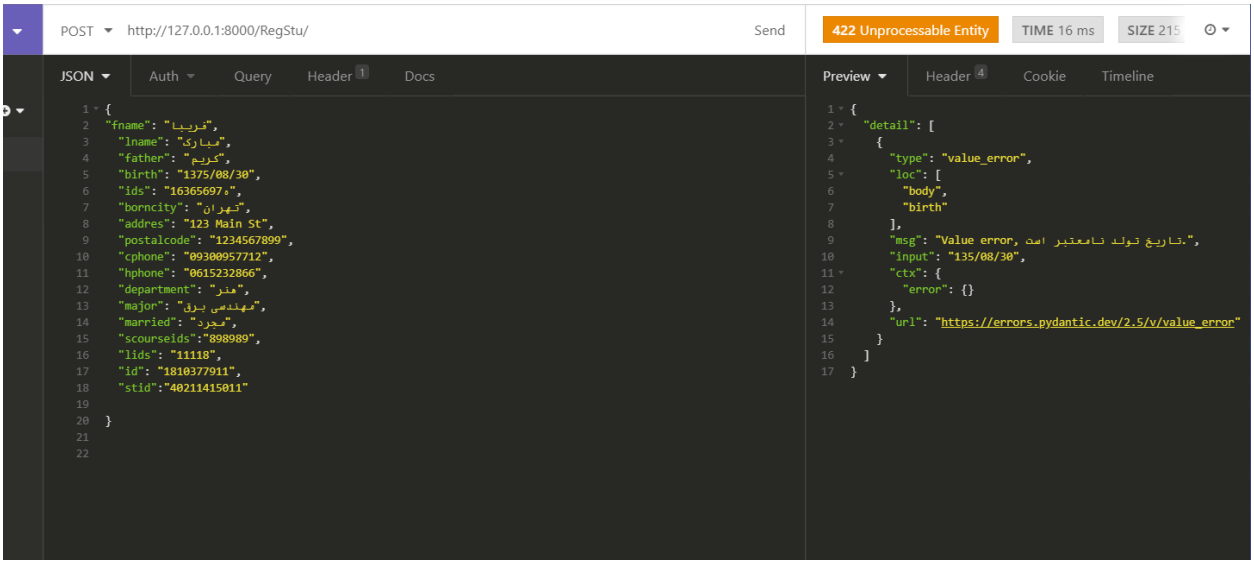
شکل۱۶.درج رکورد با مقادیر نامعتبر برای فیلدهای lname,fname,father

طبق تصویر بالا، نام را انگلیسی نوشتم و خطا داد. وقتی خطا را رفع کردم و نام را فارسی نوشتم رکورد بدون خطا اضافه شد:

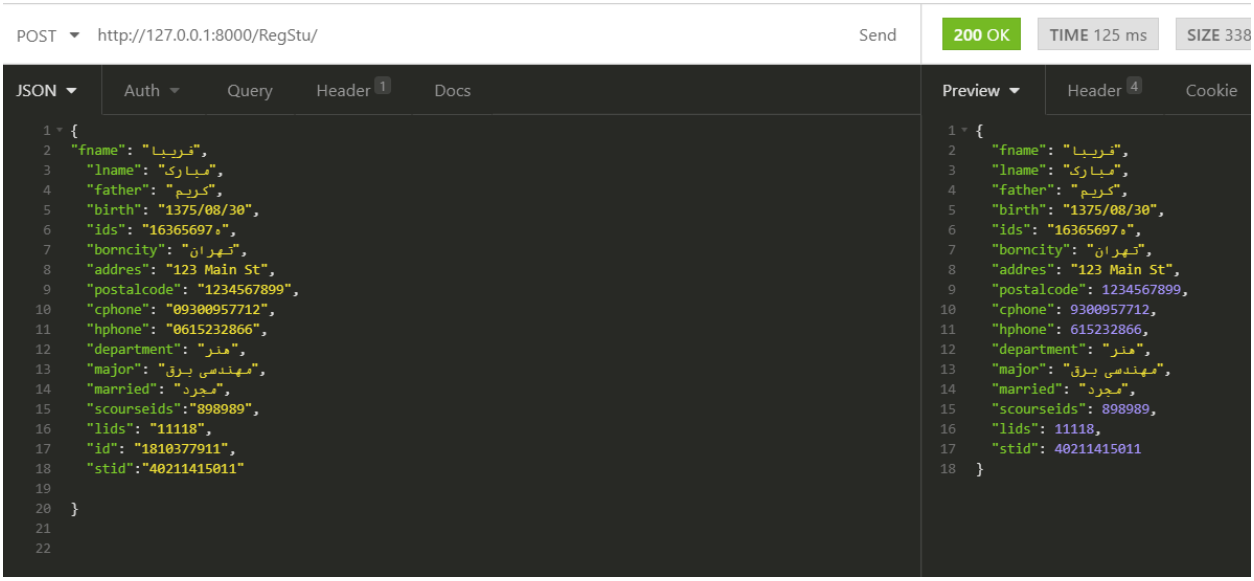


شکل۱۷.رفع خطای مربوط به نامعتبر بودن فیلدهای lname,fname,father

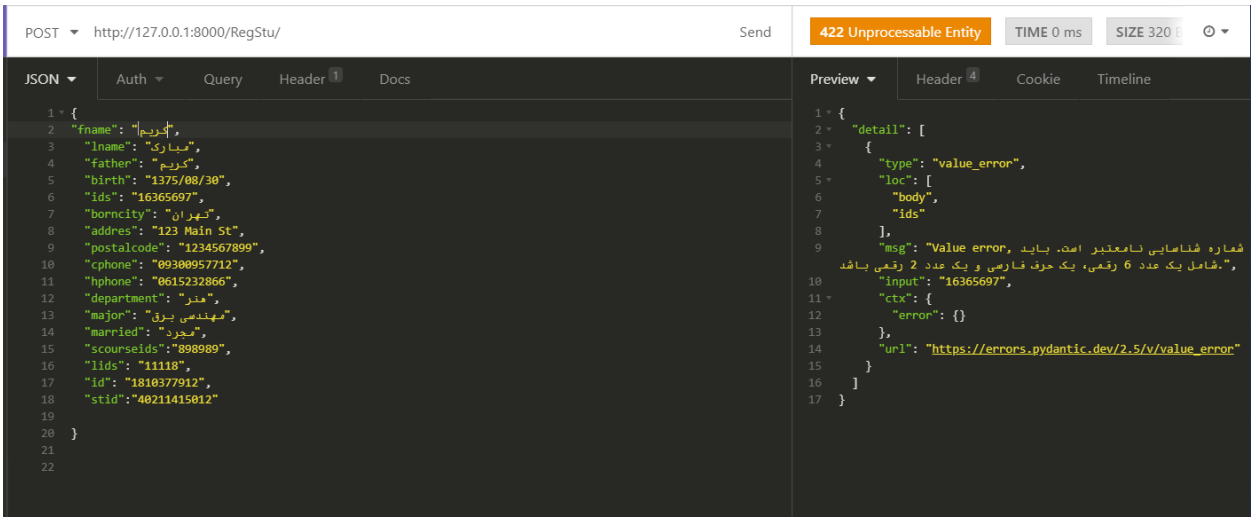
برای فیلدهای دیگر مثل تاریخ تولد و محل تولد و آدرس و شناسه کدملی و آدرس وکدپستی و نام دانشکده ورشته و وضعیت تاءهل و کد درس و کد اساتیدوکد ملی و شماره دانشجویی و سایر فیلدها نیز اعتبارسنجی کردم.در ادامه برای بررسی اعتبارسنجی هرکدام از فیلدها یک تصویر از اطلاعات نادرست و خطای آن و همچنین یک تصویر از اطلاعات درست و درج رکورد بدون خطا مشاهده میکنید:



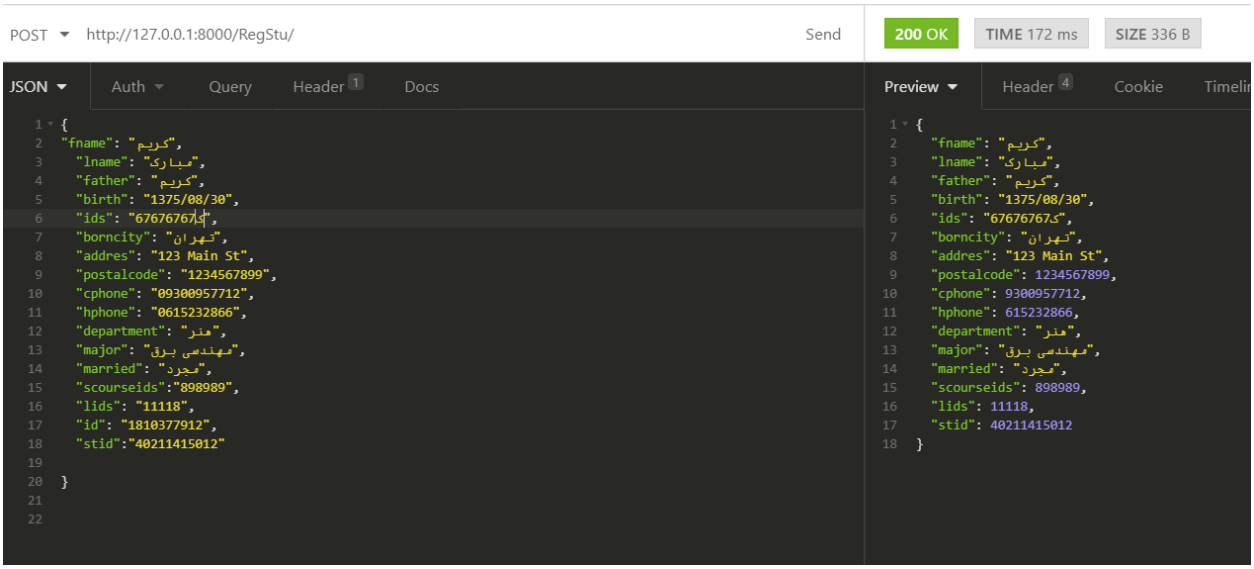
شکل۱۸.درج رکورد با مقادیر نامعتبر برای فیلد birth



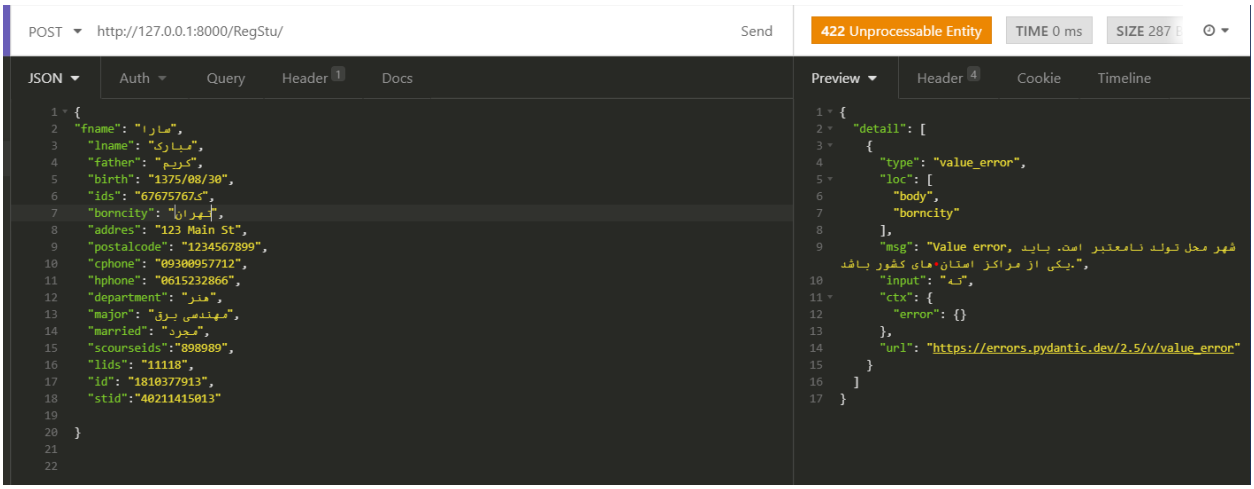
شکل۱۹. رفع خطای مربوط به نامعتبر بودن فیلد birth



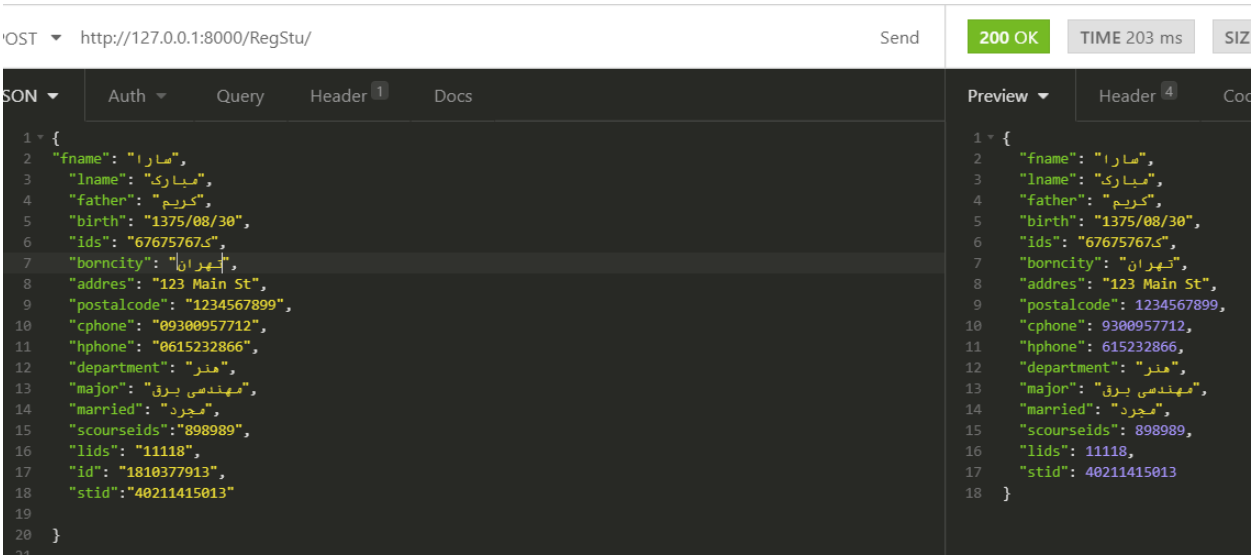
شکل۲۰. درج رکورد با مقادیر نامعتبر برای فیلد ids



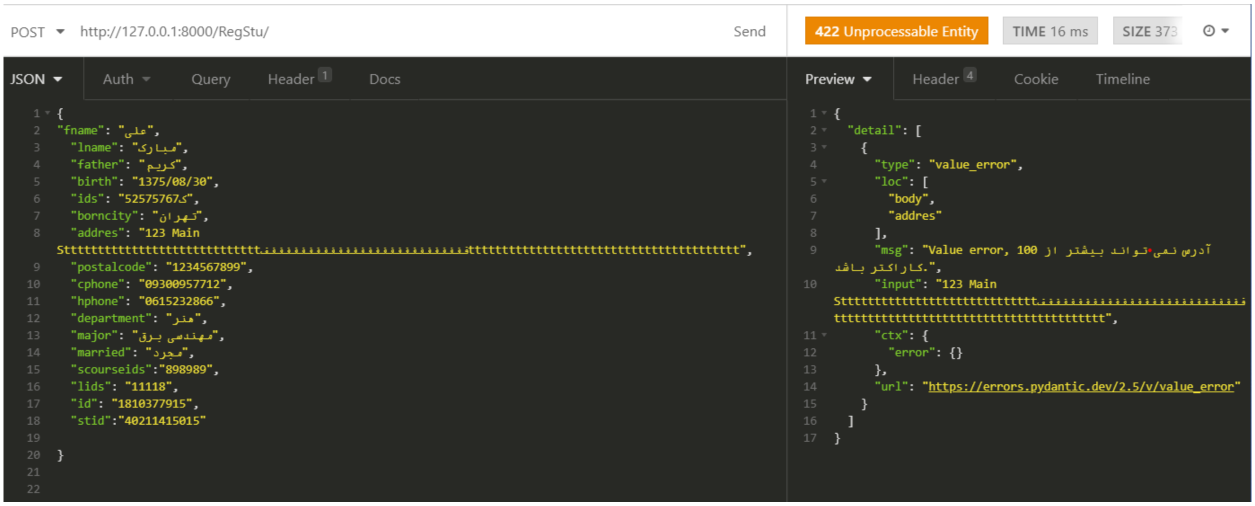
شکل۲۱. رفع خطای مربوط به نامعتبر بودن فیلد ids



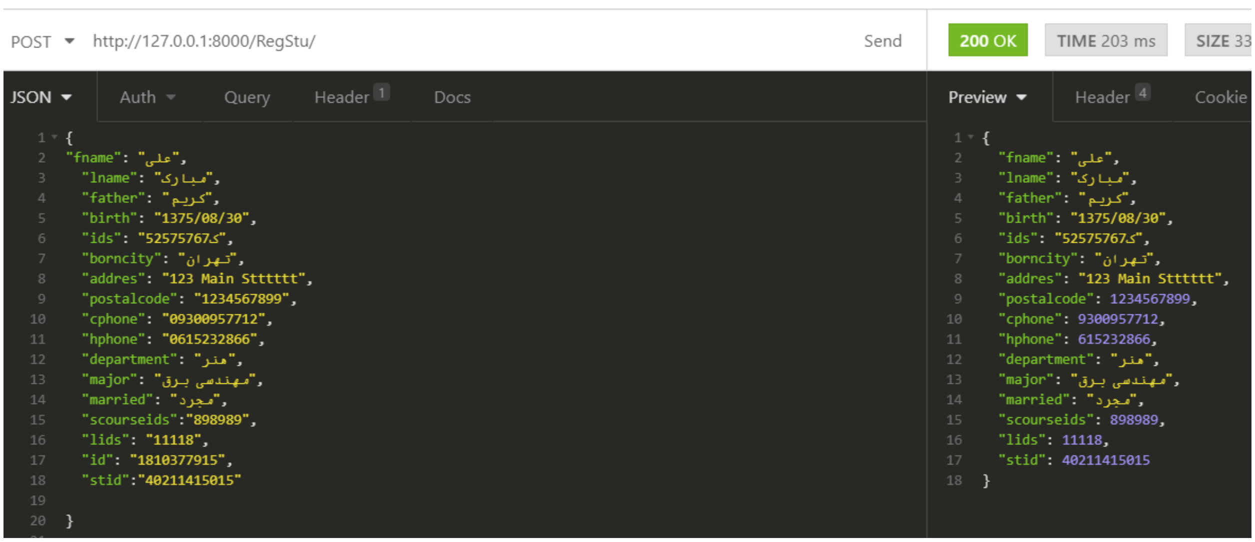
شکل۲۲. درج رکورد با مقادیر نامعتبر برای فیلد borncity



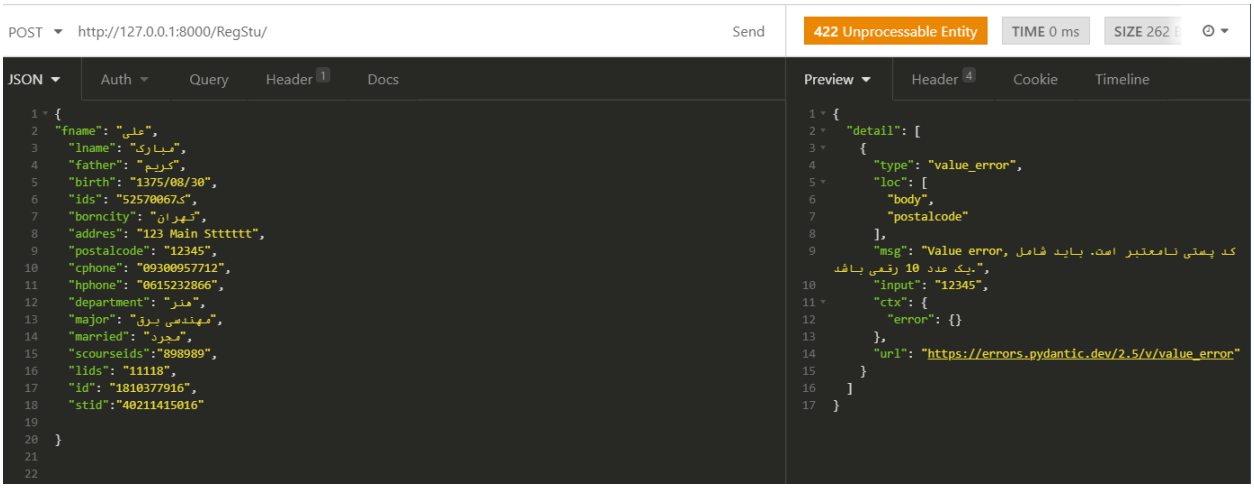
شکل۲۳. رفع خطای مربوط به نامعتبر بودن فیلد borncity



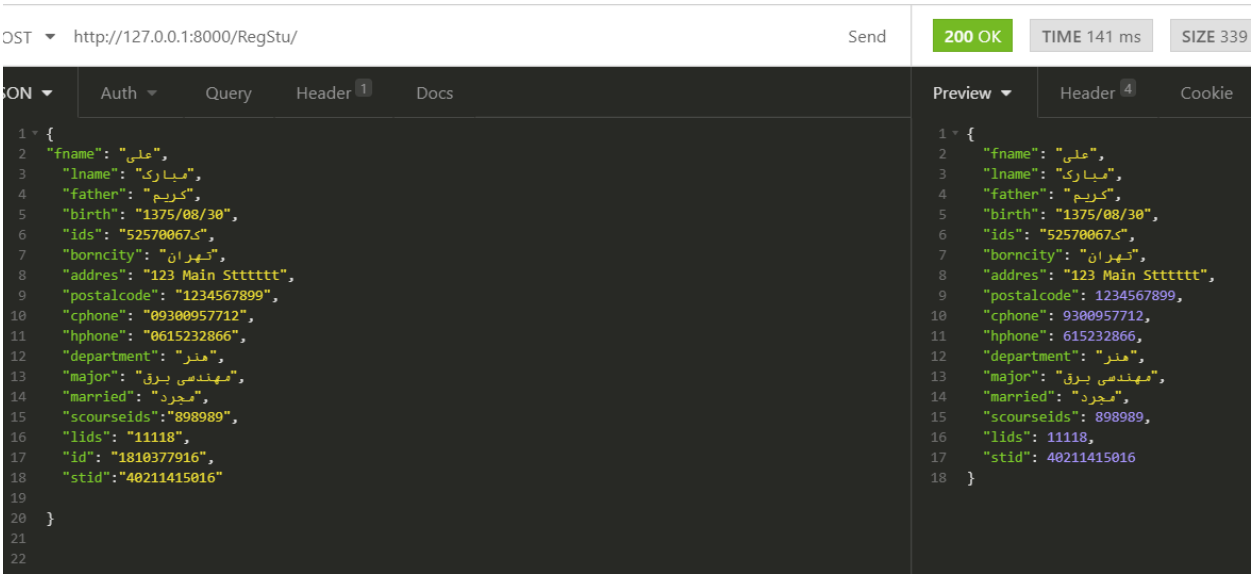
شکل۲۴. درج رکورد با مقادیر نامعتبر برای فیلد addres



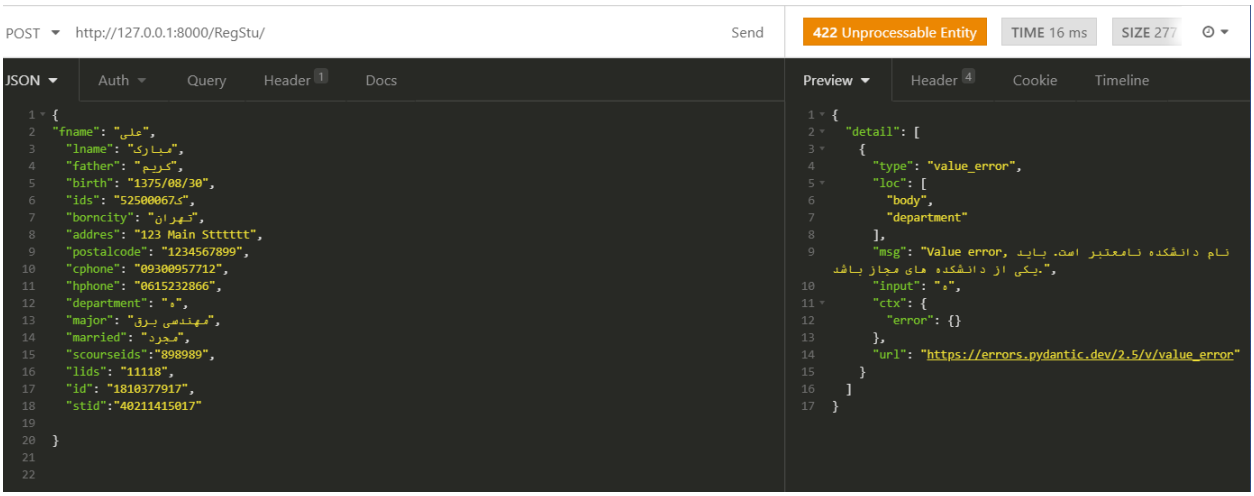
شکل۲۵. رفع خطای مربوط به نامعتبر بودن فیلد addres



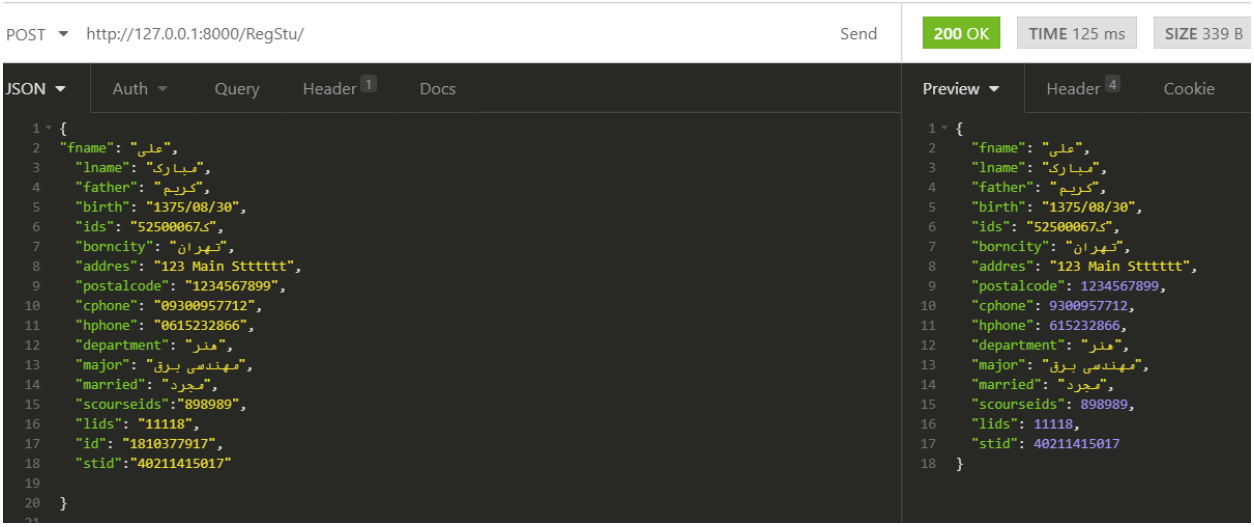
شکل۲۶. درج رکورد با مقادیر نامعتبر برای فیلد postalcode



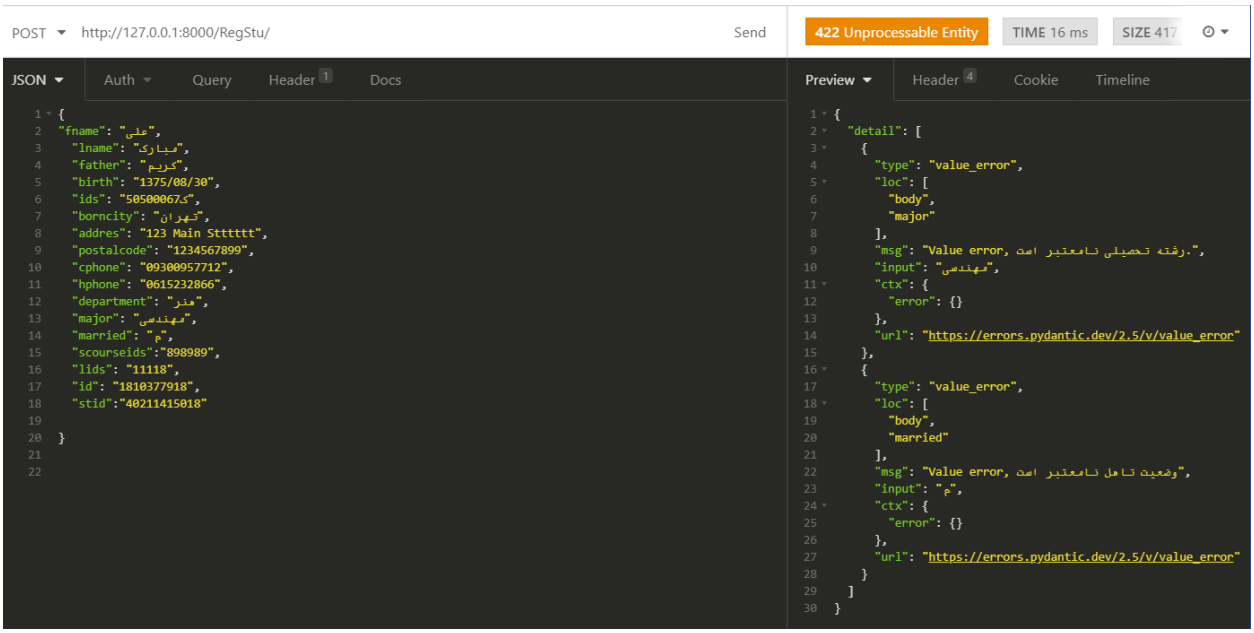
شکل۲۷. رفع خطای مربوط به نامعتبر بودن فیلد postalcode



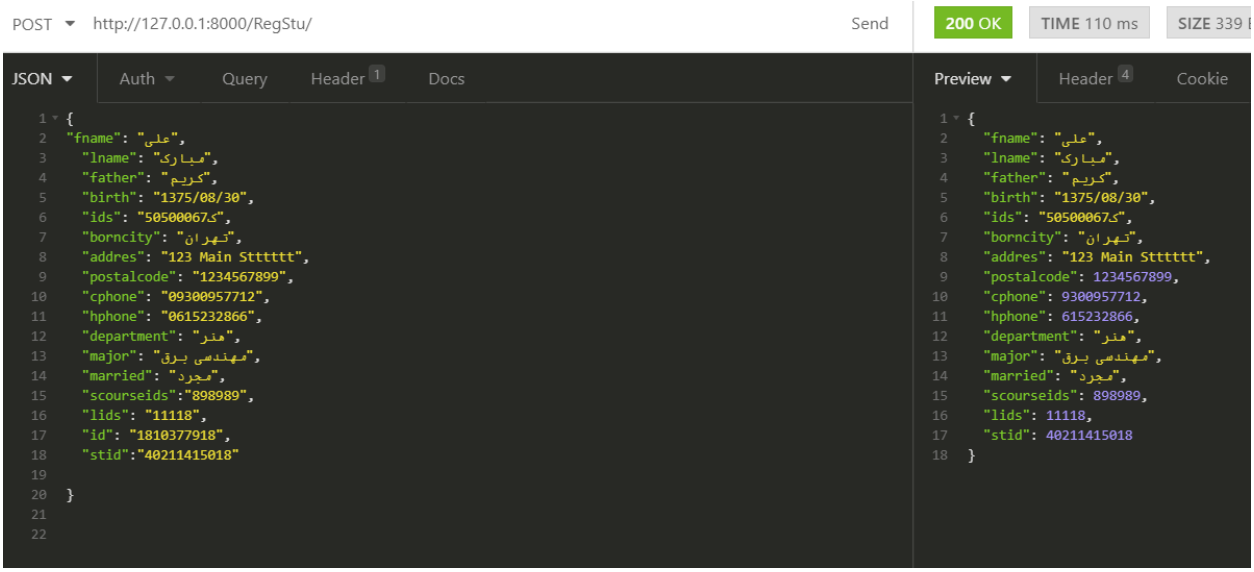
شکل۲۸. درج رکورد با مقادیر نامعتبر برای فیلد department



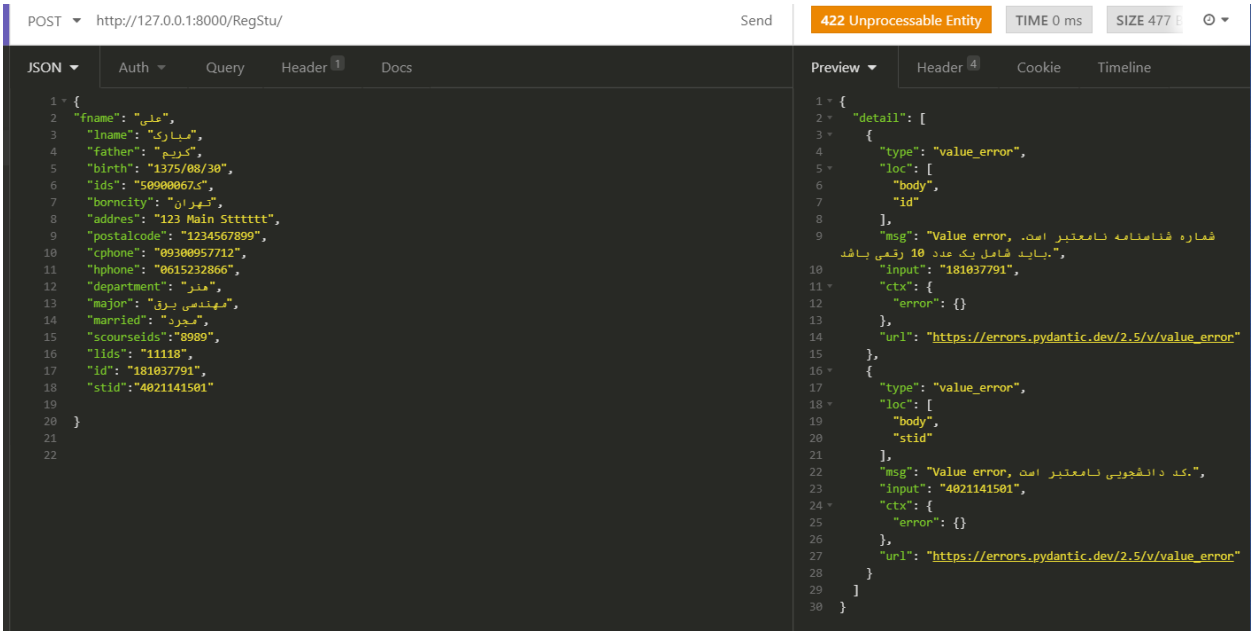
شکل۲۹. رفع خطای مربوط به نامعتبر بودن فیلد department



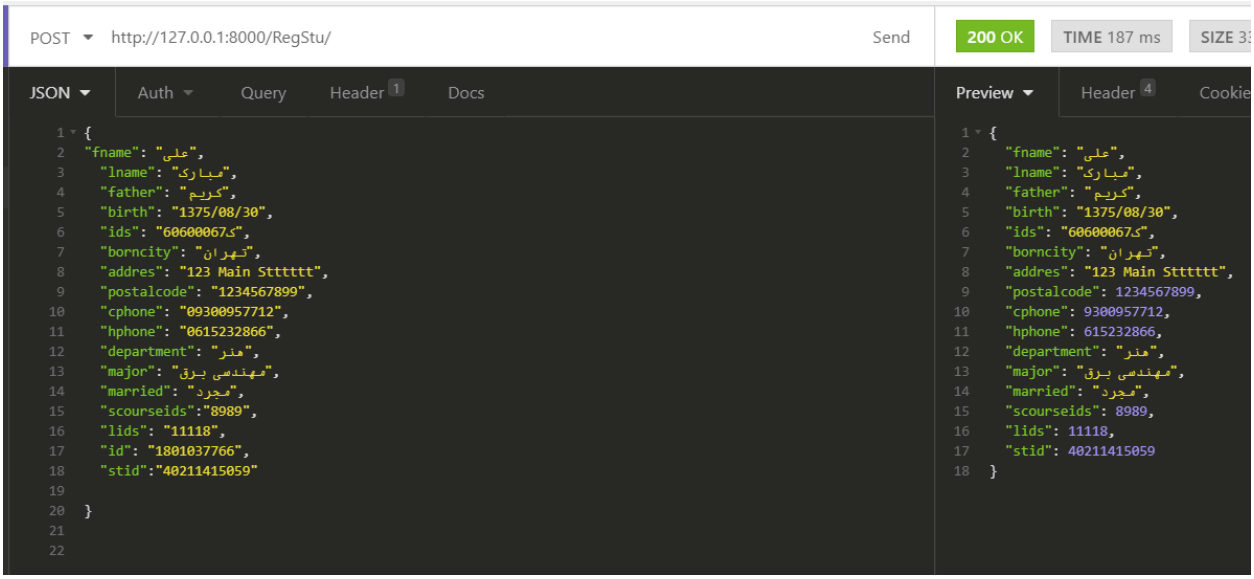
شکل ۳۰. درج رکورد با مقادیر نامعتبر برای فیلدهای major , married



شکل ۳۱. رفع خطای مربوط به نامعتبر بودن فیلد های major , married



شکل۳۲. درج رکورد با مقادیر نامعتبر برای فیلدهای id,stid



شکل۳۳. رفع خطای مربوط به نامعتبر بودن فیلد های id,stid

در ادامه برای فیلدهای جدوال teacher و lesson نیز به همین صورت اعتبارسنجی میکنیم.که در فایل های ضمیمه شده کدهای مربوط یه آن ها نیز وجود دارد و میتوان تست کرد.

پایان