

# Web Programming with Python and JavaScript

Lecture 4: ORMs and APIs

July 10, 2018

# This Time

- ORMs
- APIs

id	origin	destination	duration
1	New York	London	415
2	Shanghai	Paris	760
3	Istanbul	Tokyo	700
4	New York	Paris	435
5	Moscow	Paris	245
6	Lima	New York	455

# **Object-Oriented Programming**

**ORM**

# **Object- Relational Mapping**

# Object-Oriented Programming Relational Database Mapping

# Python SQL Mapping



# SQL

~~SQL~~  
Python

# Flask-SQLAlchemy

```
CREATE TABLE flights (  
    id SERIAL PRIMARY KEY,  
    origin VARCHAR NOT NULL,  
    destination VARCHAR NOT NULL,  
    duration INTEGER NOT NULL  
);
```

```
class Flight(db.model):  
    __tablename__ = "flights"  
    id = db.Column(db.Integer, primary_key=True)  
    origin = db.Column(db.String, nullable=False)  
    destination = db.Column(db.String, nullable=False)  
    duration = db.Column(db.Integer, nullable=False)
```

```
CREATE TABLE flights (  
    id SERIAL PRIMARY KEY,  
    origin VARCHAR NOT NULL,  
    destination VARCHAR NOT NULL,  
    duration INTEGER NOT NULL  
);
```

---

```
class Flight(db.model):  
    __tablename__ = "flights"  
    id = db.Column(db.Integer, primary_key=True)  
    origin = db.Column(db.String, nullable=False)  
    destination = db.Column(db.String, nullable=False)  
    duration = db.Column(db.Integer, nullable=False)
```

```
CREATE TABLE flights (  
    id SERIAL PRIMARY KEY,  
    origin VARCHAR NOT NULL,  
    destination VARCHAR NOT NULL,  
    duration INTEGER NOT NULL  
);
```

---

```
class Flight(db.model):  
    __tablename__ = "flights"  
    id = db.Column(db.Integer, primary_key=True)  
    origin = db.Column(db.String, nullable=False)  
    destination = db.Column(db.String, nullable=False)  
    duration = db.Column(db.Integer, nullable=False)
```

```
CREATE TABLE flights (  
    id SERIAL PRIMARY KEY,  
    origin VARCHAR NOT NULL,  
    destination VARCHAR NOT NULL,  
    duration INTEGER NOT NULL  
);
```

---

```
class Flight(db.model):  
    __tablename__ = "flights"  
    id = db.Column(db.Integer, primary_key=True)  
    origin = db.Column(db.String, nullable=False)  
    destination = db.Column(db.String, nullable=False)  
    duration = db.Column(db.Integer, nullable=False)
```



```
CREATE TABLE flights (  
    id SERIAL PRIMARY KEY,  
    origin VARCHAR NOT NULL,  
    destination VARCHAR NOT NULL,  
    duration INTEGER NOT NULL  
);
```

---

```
class Flight(db.model):  
    __tablename__ = "flights"  
    id = db.Column(db.Integer, primary_key=True)  
    origin = db.Column(db.String, nullable=False)  
    destination = db.Column(db.String, nullable=False)  
    duration = db.Column(db.Integer, nullable=False)
```

```
CREATE TABLE flights (  
    id SERIAL PRIMARY KEY,  
    origin VARCHAR NOT NULL,  
    destination VARCHAR NOT NULL,  
    duration INTEGER NOT NULL  
);
```

---

```
class Flight(db.model):  
    __tablename__ = "flights"  
    id = db.Column(db.Integer, primary_key=True)  
    origin = db.Column(db.String, nullable=False)  
    destination = db.Column(db.String, nullable=False)  
    duration = db.Column(db.Integer, nullable=False)
```

```
CREATE TABLE flights (  
    id SERIAL PRIMARY KEY,  
    origin VARCHAR NOT NULL,  
    destination VARCHAR NOT NULL,  
    duration INTEGER NOT NULL  
);
```

---

```
class Flight(db.model):  
    __tablename__ = "flights"  
    id = db.Column(db.Integer, primary_key=True)  
    origin = db.Column(db.String, nullable=False)  
    destination = db.Column(db.String, nullable=False)  
    duration = db.Column(db.Integer, nullable=False)
```

```
CREATE TABLE flights (  
    id SERIAL PRIMARY KEY,  
    origin VARCHAR NOT NULL,  
    destination VARCHAR NOT NULL,  
    duration INTEGER NOT NULL  
);
```

---

```
db.create_all()
```

```
INSERT INTO flights  
(origin, destination, duration)  
VALUES  
('New York', 'Paris', 540);
```

---

```
INSERT INTO flights  
(origin, destination, duration)  
VALUES  
( 'New York', 'Paris', 540);
```

---

```
flight = Flight(origin = 'New York',  
                 destination = 'Paris',  
                 duration = 540)  
db.session.add(flight)
```

```
SELECT * FROM flights;
```

---

```
SELECT * FROM flights;
```

---

```
Flight.query.all()
```



```
SELECT * FROM flights  
WHERE origin = 'Paris';
```

---

```
SELECT * FROM flights  
WHERE origin = 'Paris';
```

---

```
Flight.query.filter_by(origin='Paris').all()
```

```
SELECT * FROM flights  
WHERE origin = 'Paris' LIMIT 1;
```

---

```
SELECT * FROM flights  
WHERE origin = 'Paris' LIMIT 1;
```

---

```
Flight.query.filter_by(origin='Paris').first()
```

```
SELECT COUNT(*) FROM flights  
WHERE origin = 'Paris';
```

---

```
SELECT COUNT(*) FROM flights  
WHERE origin = 'Paris';
```

---

```
Flight.query.filter_by(origin='Paris').count()
```

```
SELECT * FROM flights  
WHERE id = 44;
```

---

```
SELECT * FROM flights  
WHERE id = 44;
```

---

```
Flight.query.filter_by(id=44).first()
```



```
SELECT * FROM flights  
WHERE id = 44;
```

---

```
Flight.query.get(44)
```

```
UPDATE flights  
SET duration = 280  
WHERE id = 6;
```

---

```
UPDATE flights  
SET duration = 280  
WHERE id = 6;
```

---

```
flight = Flight.query.get(6)  
flight.duration = 280
```

```
DELETE FROM flights  
WHERE id = 44;
```

---

```
DELETE FROM flights  
WHERE id = 44;
```

---

```
flight = Flight.query.get(44)  
db.session.delete(flight)
```

# COMMIT

---

# COMMIT

---

```
db.session.commit()
```

```
SELECT * FROM flights  
ORDER BY origin;
```

---



```
SELECT * FROM flights  
ORDER BY origin;
```

---

```
Flight.query.order_by(Flight.origin).all()
```

```
SELECT * FROM flights  
ORDER BY origin DESC;
```

---

```
SELECT * FROM flights  
ORDER BY origin DESC;
```

---

```
Flight.query.order_by(Flight.origin.desc()).all()
```

```
SELECT * FROM flights  
WHERE origin != 'Paris';
```

---

```
SELECT * FROM flights  
WHERE origin != 'Paris';
```

---

```
Flight.query.filter(  
    Flight.origin != 'Paris').all()
```

```
SELECT * FROM flights  
WHERE origin != 'Paris';
```

---

```
Flight.query.filter(  
    Flight.origin != 'Paris').all()
```

```
SELECT * FROM flights  
WHERE origin LIKE '%a%';
```

---

```
SELECT * FROM flights  
WHERE origin LIKE '%a%';
```

---

```
Flight.query.filter(  
    Flight.origin.like('%a%')).all()
```



```
SELECT * FROM flights  
WHERE origin IN ('Tokyo', 'Paris');
```

---

```
SELECT * FROM flights  
WHERE origin IN ('Tokyo', 'Paris');
```

---

```
Flight.query.filter(  
    Flight.origin.in_  
        ['Tokyo', 'Paris'])).all()
```

```
SELECT * FROM flights  
WHERE origin IN ('Tokyo', 'Paris');
```

---

```
Flight.query.filter(  
    Flight.origin.in_  
        ['Tokyo', 'Paris'])).all()
```

```
SELECT * FROM flights  
WHERE origin = 'Paris'  
AND duration > 500;
```

---

```
SELECT * FROM flights  
WHERE origin = 'Paris'  
AND duration > 500;
```

---

```
Flight.query.filter(  
    and_(Flight.origin == 'Paris',  
        Flight.duration > 500)).all()
```

```
SELECT * FROM flights  
WHERE origin = 'Paris'  
AND duration > 500;
```

---

```
Flight.query.filter(  
    and_(Flight.origin == 'Paris',  
        Flight.duration > 500)).all()
```

```
SELECT * FROM flights  
WHERE origin = 'Paris'  
OR duration > 500;
```

---

```
SELECT * FROM flights  
WHERE origin = 'Paris'  
OR duration > 500;
```

---

```
Flight.query.filter(  
    or_(Flight.origin == 'Paris',  
        Flight.duration > 500)).all()
```



```
SELECT * FROM flights  
WHERE origin = 'Paris'  
OR duration > 500;
```

---

```
Flight.query.filter(  
    or_(Flight.origin == 'Paris',  
        Flight.duration > 500)).all()
```

```
SELECT * FROM flights JOIN passengers  
ON flights.id = passengers.flight_id;
```

---

```
SELECT * FROM flights JOIN passengers  
ON flights.id = passengers.flight_id;
```

---

```
db.session.query(Flight, Passenger).filter(  
    Flight.id == Passenger.flight_id).all()
```

# Relationships

```
SELECT * FROM passengers  
WHERE flight_id = 1;
```

---

```
SELECT * FROM passengers  
WHERE flight_id = 1;
```

---

```
Flight.query.get(1).passengers
```

```
SELECT * FROM flights JOIN passengers  
ON flights.id = passengers.flight_id  
WHERE passengers.name = 'Alice';
```

---

```
SELECT * FROM flights JOIN passengers  
ON flights.id = passengers.flight_id  
WHERE passengers.name = 'Alice';
```

---

```
Passenger.query.filter_by(name='Alice').first().flight
```



```
SELECT * FROM flights JOIN passengers  
ON flights.id = passengers.flight_id  
WHERE passengers.name = 'Alice';
```

---

```
Passenger.query.filter_by(name='Alice').first().flight
```

**Break**

# APIs

# JSON

```
{  
  "origin": "Tokyo",  
  "destination": "Shanghai",  
  "duration": 185  
}
```

```
{  
  "origin": "Tokyo",  
  "destination": "Shanghai",  
  "duration": 185,  
  "passengers": ["Alice", "Bob"]  
}
```

```
{  
  "origin": {  
    "city": "Tokyo",  
    "code": "HND"  
  },  
  "destination": {  
    "city": "Shanghai",  
    "code": "PVG"  
  },  
  "duration": 185,  
  "passengers": ["Alice", "Bob"]  
}
```

/flights

/flights/44

/flights/44/passengers

/flights/44/passengers/6



# HTTP Methods

- GET
- POST
- PUT
- PATCH
- DELETE

# HTTP Methods

- **GET:** retrieve a resource (SELECT)
- **POST:** create a new resource (INSERT)
- **PUT:** replace a resource
- **PATCH:** update a resource (UPDATE)
- **DELETE:** delete a resource (DELETE)

**requests module**

# Using the requests module

- `requests.get(url)`
- `requests.post(url)`
- `requests.put(url)`
- `requests.patch(url)`
- `requests.delete(url)`

# HTTP Status Codes

- 200 OK
- 201 Created
- 400 Bad Request
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed
- 422 Unprocessable Entity
- ...

# HTTP Status Codes

- 200 OK
- 201 Created
- 400 Bad Request
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed
- 422 Unprocessable Entity
- ...

# HTTP Status Codes

- 200 OK
- 201 Created
- 400 Bad Request
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed
- 422 Unprocessable Entity
- ...

# API Keys