

Flask-tut tut1.py



tut1



Project Flask-tut C:\Users\Haris\PycharmProjects\Flask-tut



External Libraries

Scratches and Consoles

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 @app.route("/")
5 def hello():
6     return "Hello World!"
7
8 @app.route("/harry")
9 def harry():
10    return "Hello harry bhai4!"
11 app.run(debug=True)
```

Run: tut1

WARNING: Do not use the development server in a production environment.  
Use a production WSGI server instead.  
\* Debug mode: on  
\* Restarting with stat  
\* Debugger is active!  
\* Debugger PIN: 326-619-872  
\* Running on <http://127.0.0.1:5000/> (Press CTRL+C to quit)

You are using the Flask Framework

PyCharm Professional Edition has special support for it.

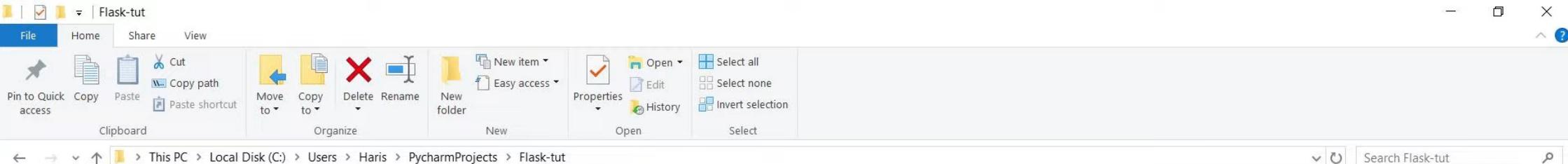
PEP 8: expected 2 blank lines after class or function definition, found 0. PEP 8: no newline at end of file.

11:20 n/a UTF-8 ENG



Type here to search





Quick access

- Desktop
- 2018
- Haris
- PycharmProjects
- 11.0
- Bandicam
- Downloads
- Documents
- Pictures
- Convocation
- Favorite Songs
- Photoshop Templates
- Video Songs
- OneDrive
- This PC



4 items 1 item selected



Type here to search



ENG 3

## [Hindi] Static and Templates folder - Web Development Using Flask and Python #3

FOLDERS

- templates
  - about.html
  - index.html

```
index.html      about.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Title of my page</title>
5  </head>
6  <body>
7
8  <h1>Hi I am {{ name2 }}!</h1>
9
10 <p>This is a way for me to teach you flask nicely</p>
11 </body>
12 </html>
```

Run:  tut1 

Use a production WSGI server instead

- ```
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 326-619-872
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1:5000/2018-06-20-101 "GET / HTTP/1.1" 200
```

 **You are using the Flask Framework**  
PyCharm Professional Edition has special support for it.

## FOLDERS

- templates
  - about.html
  - bootstrap.html
  - index.html

index.html about.html bootstrap.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Title of my page</title>
5 </head>
6 <body>
7 
8 <p>In this video, I have shown you how to create a flask app using templates. I have used
render template function from the Flask Module</p>
9 <p>Hope you will like this video!</p>
10
11 </body>
12 </html>
```



Type here to search



ENG



[Hindi] Flask SQLAlchemy Tutorial x +

codewithharry.com/videos/web-dev-using-flask-and-python-9/

CodeWithHarry

Home Courses Tutorial Blog Notes Contact My Gear Work With Us ☾ Login Signup

HTML CSS JS C C++ JAVA PYTHON PHP REACT JS

MORE VIDEOS

Want to get in touch? Fill out the form below to send me a message and I will get back to you as soon as possible!

23:34 / 24:38 Type here to search

Subscribe

CC HD YouTube

Hindi Basics of Jinja Templating - Web

Overview Q&A Downloads Announcements

# [Hindi] Flask SQLAlchemy Tutorial In Hindi - Web Development Using Flask and Python #9

In this blog we will learn how to send form data to our database table! We will make contact page working. For that we will use Flask SQLAlchemy. It sounds complicated but it's really not. I will try to make it as easy as possible.

First we will install it in our system through pip. Open command prompt and write:

```
pip install flask-sqlalchemy  
or  
pip install flask_sqlalchemy
```

Now import this module in your python file by writing:

```
from flask_sqlalchemy import SQLAlchemy
```

To connect with the database first we have to give database's address. For that the code is:

89°F Haze 10:45 AM 5/29/2023



Login

Signup



HTML CSS JS C C++ JAVA PYTHON PHP REACT JS



```
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://username:password@localhost/db_name'
```

In this if you have not set username or password, you can write "root" as username and nothing in password. In db\_name you will write database's name like in the video it was codingthunder. After editing it would look like this:

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:@localhost/codingthunder'
```

Now our flask app knows to which database it has to connect!

Now we will initialize the object/class which we imported from flask\_sqlalchemy i.e. SQLAlchemy. To do this we will have to make a variable, code is like this:

```
db = SQLAlchemy(app)
```

Once we are connected to database, we need to access our tables. Remember we made tables in our database? To access tables we need to make a class, like this:

```
class Contacts(db.Model):
    sno = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(80), nullable=False)
    phone_num = db.Column(db.String(12), nullable=False)
    msg = db.Column(db.String(120), nullable=False)
```



Now our flask app knows to which database it has to connect!

Now we will initialize the object/class which we imported from flask\_sqlalchemy i.e. SQLAlchemy. To do this we will have to make a variable, code is like this:

```
db = SQLAlchemy(app)
```

Once we are connected to database, we need to access our tables. Remember we made tables in our database? To access tables we need to make a class, like this:

```
class Contacts(db.Model):
    sno = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(80), nullable=False)
    phone_num = db.Column(db.String(12), nullable=False)
    msg = db.Column(db.String(120), nullable=False)
    date = db.Column(db.String(12), nullable=True)
    email = db.Column(db.String(20), nullable=False)
```

With this class we are accessing our "contacts" table. "db.Integer" is like setting data types, means this variable will have this type of data. Similarly "db.String" for name and the integer argument is maximum length of string. "nullable" means can this variable be null? So some things are false, some true. You can decide according to your requirement.

## Post request in contact form:



## Post request in contact form:

When a user fills contact us form and presses submit, we want the page to make a post request with which it will post all that form information in our database's table.

## How to make a post request?

We want contact us page to make post request so in our flask app we will give post method to that function which is returning that page, in this case contact(), like this:

```
@app.route("/contact", methods = ['GET', 'POST'])
def contact():
    return render_template('contact.html')
```

We wrote 'GET' in methods. Get request is used anyways. You are opening a URL that is get request, loading an image is get request, anything showing in a website is through get request. Getting anything, even a simple plain paragraph is get request.

In the function written above, we have just coded that it will return contact.html when visited "/contact" end point. Now we will code and tell what to do when a post request is made on this page.

First we will make an 'if' statement that only do it when post request is made. So for that we will import request from flask and write this:

```
@app.route("/contact", methods = ['GET', 'POST'])
def contact():
```

```
@app.route("/contact", methods = ['GET', 'POST'])
def contact():
    return render_template('contact.html')
```

We wrote 'GET' in methods. Get request is used anyways. You are opening a URL that is get request, loading an image is get request, anything showing in a website is through get request. Getting anything, even a simple plain paragraph is get request.

In the function written above, we have just coded that it will return contact.html when visited “/contact” end point. Now we will code and tell what to do when a post request is made on this page.

First we will make an 'if' statement that only do it when post request is made. So for that we will import request from flask and write this:

```
@app.route("/contact", methods = [ 'GET', 'POST'])
def contact():
    if(request.method=='POST'):
        '''Fetch data and add it to the database'''
    return render_template('contact.html')
```

Now we have programmed it to do whatever is written in that if block when made post request. First we will fetch the data from contact form and then post it to the database table.

How to fetch data from form and send to database?



```
@app.route("/contact", methods = ['GET', 'POST'])
def contact():
    if(request.method=='POST'):
        '''Fetch data and add it to the database'''
    return render_template('contact.html')
```

Now we have programmed it to do whatever is written in that if block when made post request. First we will fetch the data from contact form and then post it to the database table.

## How to fetch data from form and send to database?

First you have to go in HTML and give name attribute to every element that has data which we want to fetch, like this:

```
<input name="email" type="email" placeholder="Email Address">
```

Then through these names we can locate the elements from which we want to fetch data and then send to database, like this:

```
@app.route("/contact", methods = ['GET', 'POST'])
def contact():
    if(request.method=='POST'):
        '''Add entry to the database'''
    name = request.form.get('name')
```

[Hindi] Flask SQLAlchemy Tutorial

codewithharry.com/videos/web-dev-using-flask-and-python-9/

# CodeWithHarry

Home Courses Tutorial Blog Notes Contact My Gear Work With Us ☾ Login Signup

HTML CSS JS C C++ JAVA PYTHON PHP REACT JS

Search icon

```
@app.route("/contact", methods = [ 'GET', 'POST'])
def contact():
    if(request.method=='POST'):
        '''Add entry to the database'''
        name = request.form.get('name')
        email = request.form.get('email')
        phone = request.form.get('phone')
        message = request.form.get('message')
        entry = Contacts(name=name, phone_num = phone, msg = message, date= datetime.now(),email = email)
        db.session.add(entry)
        db.session.commit()
    return render_template('contact.html')
```

In this we are fetching name, email, phone and message. Remember we made a class 'Contacts'? In variable 'entry' we are using that class(variables in that class are the arguments used here), then we are adding it to the database's table.

Everything is done in our flask app but one thing is left in HTML file to make it work. We have to add method and action in our HTML file, like this:

```
<form name="sentMessage" action = "/contact" method="post" novalidate>
```

Now everytime somebody clicks on submit it will make a request on "/contact" end point and we said method is post so request will be post request.



```
<form name="sentMessage" action = "/contact" method="post" novalidate>
```

Now everytime somebody clicks on submit it will make a request on "/contact" end point and we said method is post so request will be post request.

Now everything is done!

## 1. main.py file as described in the video

```
from flask import Flask, render_template, request
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:@localhost/codingthunder'
db = SQLAlchemy(app)

class Contacts(db.Model):
    ...
    sno, name, phone_num, msg, date, email
    ...
    sno = db.Column(db.Integer, primary_key=True)
```



```
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:@localhost/codingthunder'
db = SQLAlchemy(app)
```

```
class Contacts(db.Model):
    ...
    sno, name phone_num, msg, date, email
    ...
    sno = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(80), nullable=False)
    phone_num = db.Column(db.String(12), nullable=False)
    msg = db.Column(db.String(120), nullable=False)
    date = db.Column(db.String(12), nullable=True)
    email = db.Column(db.String(20), nullable=False)
```

```
@app.route("/")
def home():
    return render_template('index.html')
```

```
@app.route("/about")
def about():
```

[Hindi] Flask SQLAlchemy Tutorial

codewithharry.com/videos/web-dev-using-flask-and-python-9/

CodeWithHarry

Home Courses Tutorial Blog Notes Contact My Gear Work With Us ☾ Login Signup

HTML CSS JS C C++ JAVA PYTHON PHP REACT JS

Search icon

```
@app.route("/")
def home():
    return render_template('index.html')

@app.route("/about")
def about():
    return render_template('about.html')

@app.route("/contact", methods = ['GET', 'POST'])
def contact():
    if(request.method=='POST'):
        '''Add entry to the database'''
        name = request.form.get('name')
        email = request.form.get('email')
        phone = request.form.get('phone')
        message = request.form.get('message')
        entry = Contacts(name=name, phone_num = phone, msg = message, date= datetime.now(),email = email)
        db.session.add(entry)
        db.session.commit()
    return render_template('contact.html')
```

89°F Haze

Windows Start Search File Explorer Microsoft Edge Google Chrome LinkedIn Email Spotify Task View

10:46 AM 5/29/2023



```
        db.session.add(entry)
        db.session.commit()
    return render_template('contact.html')

app.run(debug=True)
```

## 2. File contact.html file as described in the video

```
{% extends "layout.html" %}

{% block body %}

    <!-- Page Header -->
    <header class="masthead" style="background-image: url('{{ url_for('static', filename='img/contact.jpg') }}')>
        <div class="overlay"></div>
        <div class="container">
            <div class="row">
                <div class="col-lg-8 col-md-10 mx-auto">
                    <div class="page-heading">
                        <h1>Contact Me</h1>
```

youtube.com

Search

Windows PowerShell

```
PS C:\Users\lenovo> pip install flask-sqlalchemy
Collecting flask-sqlalchemy
  Downloading Flask_SQLAlchemy-3.0.3-py3-none-any.whl (24 kB)
Requirement already satisfied: Flask>=2.2 in c:\python\python311\lib\site-packages (from flask-sqlalchemy) (2.3.2)
Collecting SQLAlchemy>=1.4.18
  Downloading SQLAlchemy-2.0.15-cp311-cp311-win_amd64.whl (2.0 MB)
  2.0/2.0 MB 6.6 MB/s eta 0:00:00
Requirement already satisfied: Werkzeug>=2.3.3 in c:\python\python311\lib\site-packages (from Flask>=2.2->flask-sqlalchemy) (2.3.4)
Requirement already satisfied: Jinja2>=3.1.2 in c:\python\python311\lib\site-packages (from Flask>=2.2->flask-sqlalchemy) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\python\python311\lib\site-packages (from Flask>=2.2->flask-sqlalchemy) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\python\python311\lib\site-packages (from Flask>=2.2->flask-sqlalchemy) (8.1.3)
Requirement already satisfied: blinker>=1.6.2 in c:\python\python311\lib\site-packages (from Flask>=2.2->flask-sqlalchemy) (1.6.2)
Collecting typing-extensions>=4.2.0
  Downloading typing_extensions-4.6.2-py3-none-any.whl (31 kB)
Collecting greenlet!=0.4.17
  Downloading greenlet-2.0.2-cp311-cp311-win_amd64.whl (192 kB)
  192.5/192.5 kB 12.1 MB/s eta 0:00:00
Requirement already satisfied: colorama in c:\python\python311\lib\site-packages (from click>=8.1.3->Flask>=2.2->flask-sqlalchemy) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\python\python311\lib\site-packages (from Jinja2>=3.1.2->Flask>=2.2->flask-sqlalchemy) (2.1.2)
Installing collected packages: typing-extensions, greenlet, SQLAlchemy, flask-sqlalchemy
Successfully installed SQLAlchemy-2.0.15 flask-sqlalchemy-3.0.3 greenlet-2.0.2 typing-extensions-4.6.2
```

[notice] A new release of pip available: 22.3.1 -> 23.1.2



स्वामी सदाफल से बिन्ती हमार बा, Swami  
Sadafal se vinati hamar ba# sadguru...



Leetcode: Longest Consecutive  
Sequence | Yash Jain



What Is Virtual Private Server? | VPS  
explained in Hindi