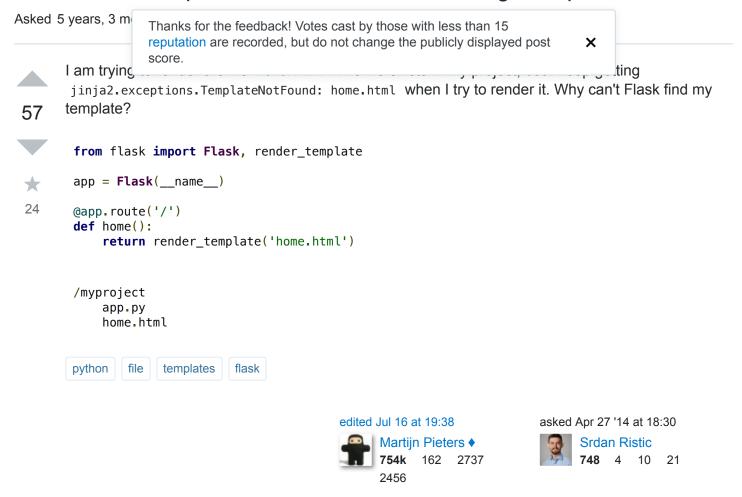
Flask raises TemplateNotFound error even though template file exists



9 Answers



You must create your template files in the correct location; in the templates subdirectory next to your python module.

129



The error indicates that there is no home.html file in the templates/ directory. Make sure you created that directory in the same directory as your python module, and that you did in fact put a home.html file in that subdirectory. If your app is a package, the templates folder should be created inside the package.



```
myproject/
    app.py
    templates/
        home.html

myproject/
    mypackage/
    __init__.py
```

```
templates/
home.html
```

You can ask Flask to explain how it tried to find a given template, by setting the EXPLAIN TEMPLATE LOADING option to True. For every template loaded, you'll get a report logged to the Flask app.logger, at level INFO.

This is what it looks like when a search is successful; in this example the foo/bar.html template extends the base.html template, so there are two searches:

```
[2019-06-15 16:03:39,197] INFO in debughelpers: Locating template
"foo/bar.html":
    1: trying loader of application "flaskpackagename"
       class: jinja2.loaders.FileSystemLoader
       encoding: 'utf-8'
       followlinks: False
       searchpath:
         - /.../project/flaskpackagename/templates
       -> found ('/.../project/flaskpackagename/templates/foo/bar.html')
[2019-06-15 16:03:39,203] INFO in debughelpers: Locating template "base.html":
    1: trying loader of application "flaskpackagename"
       class: jinja2.loaders.FileSystemLoader
       encoding: 'utf-8'
       followlinks: False
       searchpath:
         - /.../project/flaskpackagename/templates
       -> found ('/.../project/flaskpackagename/templates/base.html')
```

Blueprints can <u>register their own template directories</u> too, but this is not a requirement if you are using blueprints to make it easier to split a larger project across logical units. The main Flask app template directory is always searched first even when using additional paths per blueprint.

edited Jun 15 at 15:08

answered Apr 27 '14 at 18:36



EXPLAIN TEMPLATE LOADING is very useful to debug path issues around templates. Also, if you use Blueprints, make sure you set the template folder path per-blueprint. – Justin Krause Jun 14 at 19:48

@JustinKrause: thanks for those, EXPLAIN_TEMPLATE_LOADING was added after this answer was initially written. – Martiin Pieters ♦ Jun 15 at 14:59



I think Flask uses the directory templates by default. So your code should be

from flask import Flask, render_template

```
@app.route("/")
def home
    retu
    Thanks for the feedback! Votes cast by those with less than 15
    reputation are recorded, but do not change the publicly displayed post
    score.
def about():
    return render_template('about.html')

if __name__=="__main__":
    app.run(debug=True)
```

app=Flask(__name___,template_folder='template')

answered Mar 15 at 6:35





I don't know why, but I had to use the following folder structure instead. I put "templates" one level up.

3

```
project/
    app/
    hello.py
    static/
        main.css
    templates/
    home.html
    venv/
```

This probably indicates a misconfiguration elsewhere, but I couldn't figure out what that was and this worked.



answered Jan 24 '17 at 16:23



François Breton **921** 12 23



2

After following this thread and others for a solution to the same issue without success I found a working solution for my current project. (Please note that the above accepted Answer provided for file/project structure is going to work for most cases and is absolutely correct, I'm just showing what specifically worked for me.)



```
app = Flask(__name__, template_folder='../templates')
```

the same worked for /static/style.css after discovering .css files weren't linking properly to .html files...

```
app = Flask(__name__, template_folder='../templates',
static_folder='../static')
```

i hope this helps

edited Aug 13 at 4:23

answered Jun 1 at 2:11

Thanks for the feedback! Votes cast by those with less than 15 reputation are recorded, but do not change the publicly displayed post score.



6



Check that:

2

- 1. the template file has the right name
- 2. the template file is in a subdirectory called templates
- 3. the name you pass to render_template is relative to the template directory (index.html would be directly in the templates directory, auth/login.html would be under the auth directory in the templates directory.)
- 4. you either do not have a subdirectory with the same name as your app, or the templates directory is inside that subdir.

If that doesn't work, turn on debugging (app.debug = True) which might help figure out what's wrong.



answered Apr 27 '14 at 18:41





You need to put all you .html files in the **template** folder next to your python module. And if there are any images that you are using in your html files then you need put all your files in the folder named **static**



In the following Structure

```
project/
    hello.py
    static/
        image.jpg
        style.css
    templates/
        homepage.html
    virtual/
        filename.json
```

answered Jun 25 '17 at 14:14

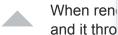




I had the same error turns out the only thing i did wrong was to name my 'templates' folder, 'template' without 's'. After changing that it worked fine, dont know why its a thing but it is.







Thanks for the feedback! Votes cast by those with less than 15 reputation are recorded, but do not change the publicly displayed post score.

×

called templates

0

- 1. the html file do not exist or
- 2. when templates folder do not exist

To solve the problem:

create a folder with name templates in the same directory where the python file is located and place the html file created in the templates folder.

answered May 23 at 20:06



Atal Kumar



Another alternative is to set the <code>root_path</code> which fixes the problem both for templates and static folders.

0



```
root_path = Path(sys.executable).parent if getattr(sys, 'frozen', False) else
Path(__file__).parent
app = Flask(__name__.split('.')[0], root_path=root_path)
```

If you render templates directly via Jinja2, then you write:

```
ENV = jinja2.Environment(loader=jinja2.FileSystemLoader(str(root_path /
'templates')))
template = ENV.get_template(your_template_name)
```

answered Jun 14 at 0:20



601

1 7 15

protected by davidism Jul 15 at 17:38

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?