

Visible Flask

AWPUG - June 2017

THE VISIBLE MAN

PATENT PENDING

A Science Assembly Project

THE WONDERS OF
THE HUMAN BODY
REVEALED!

FROM SKIN
TO SKELETON...

assemble, remove,
replace all organs!

Parts undecorated and
paints not included

Includes an
ILLUSTRATED
HANDBOOK
written by Medical
Authorities in
everyday language

THE VISIBLE MAN
by FETTER

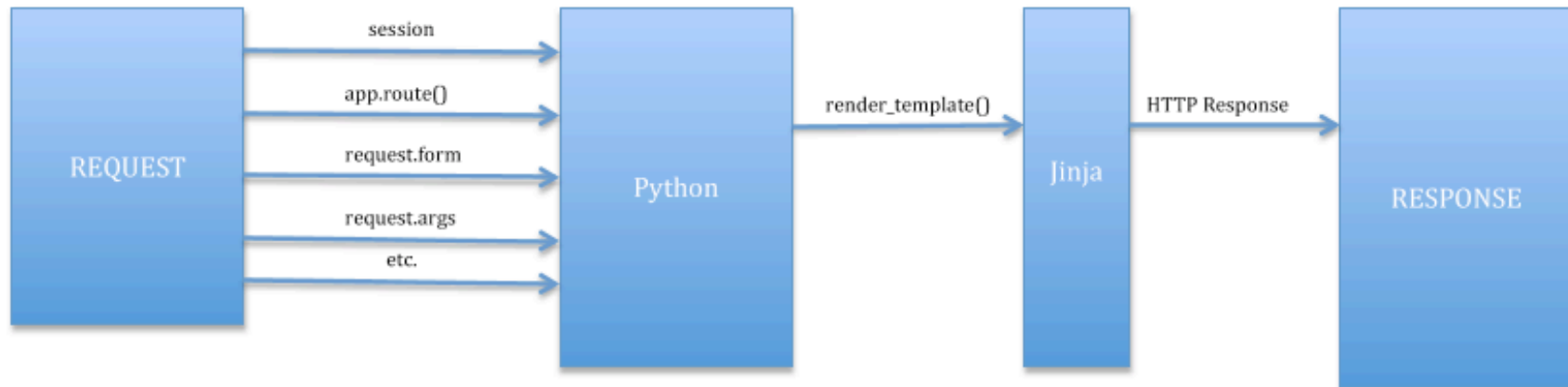
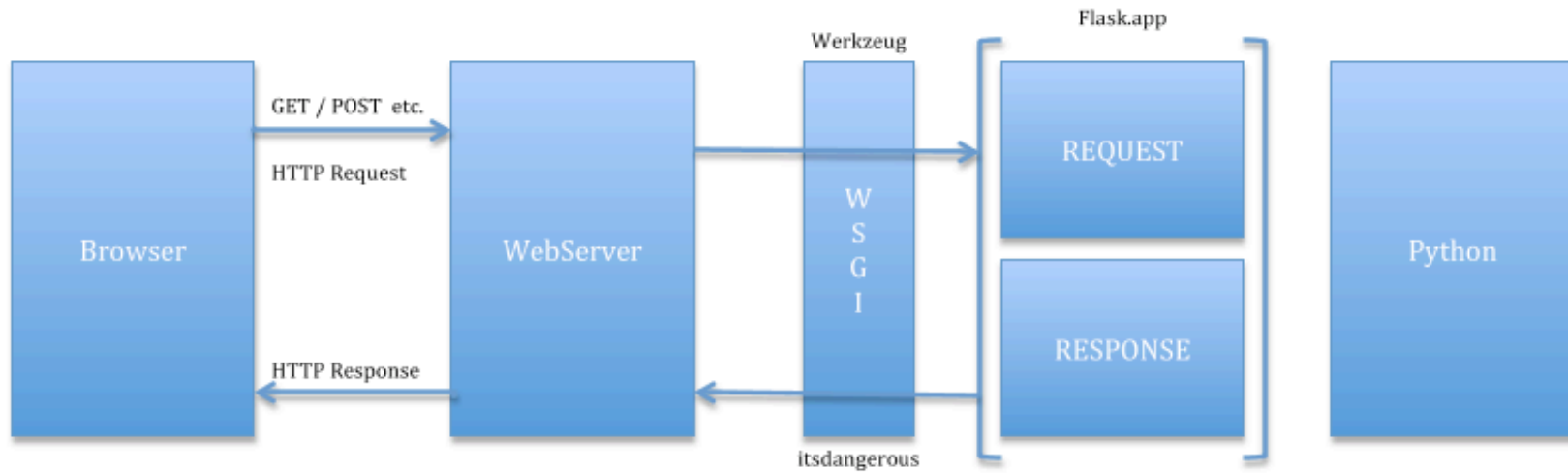
ACTUAL SIZE



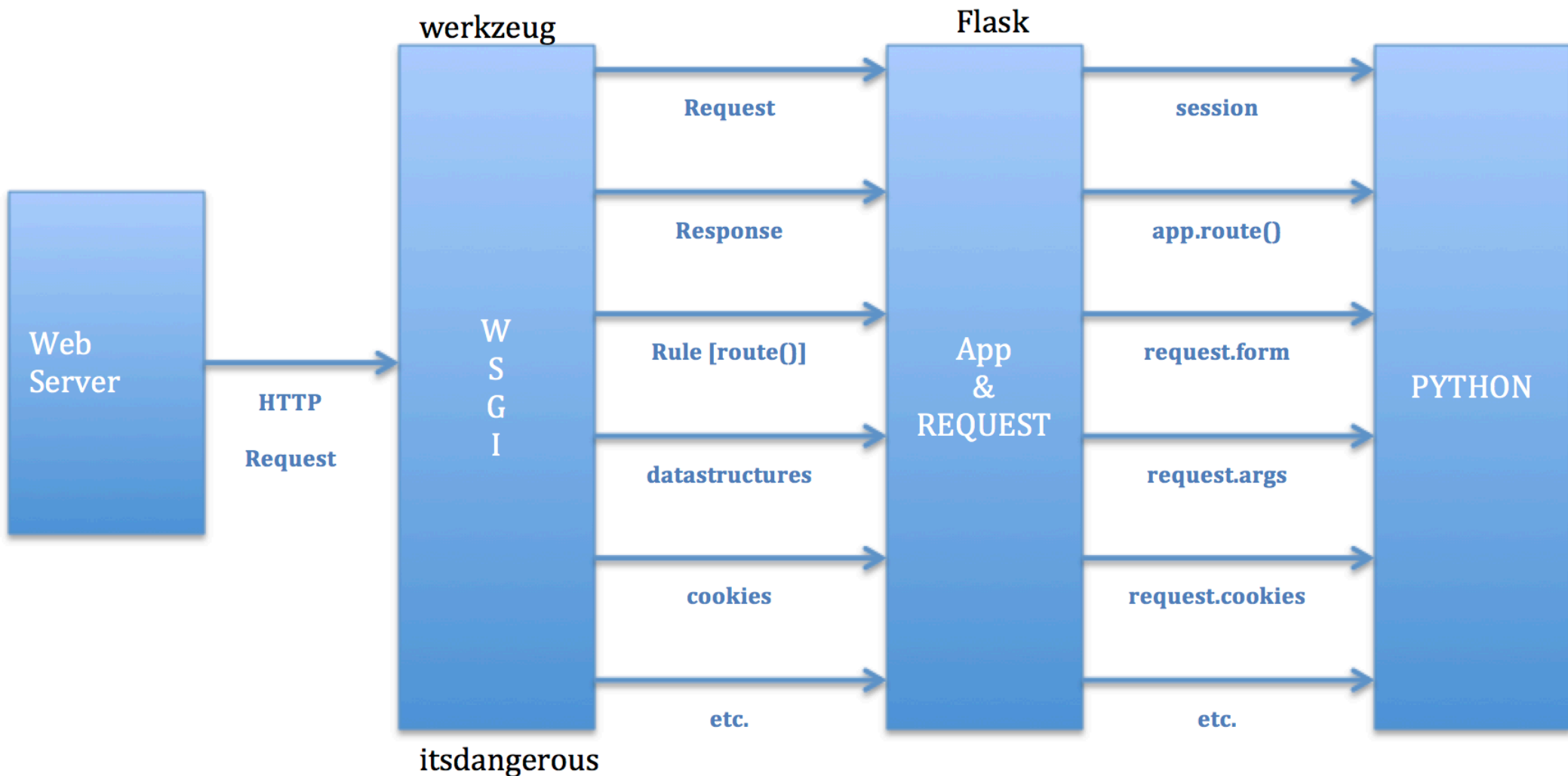
DISCLAIMER:

“Parts undecorated and paints not included.”

FLASK - Overview



App & Request



Moving Data: Browser to Server to Browser

`request.cookies`

`request.args`

`request.form`

`session`

Jinja templates

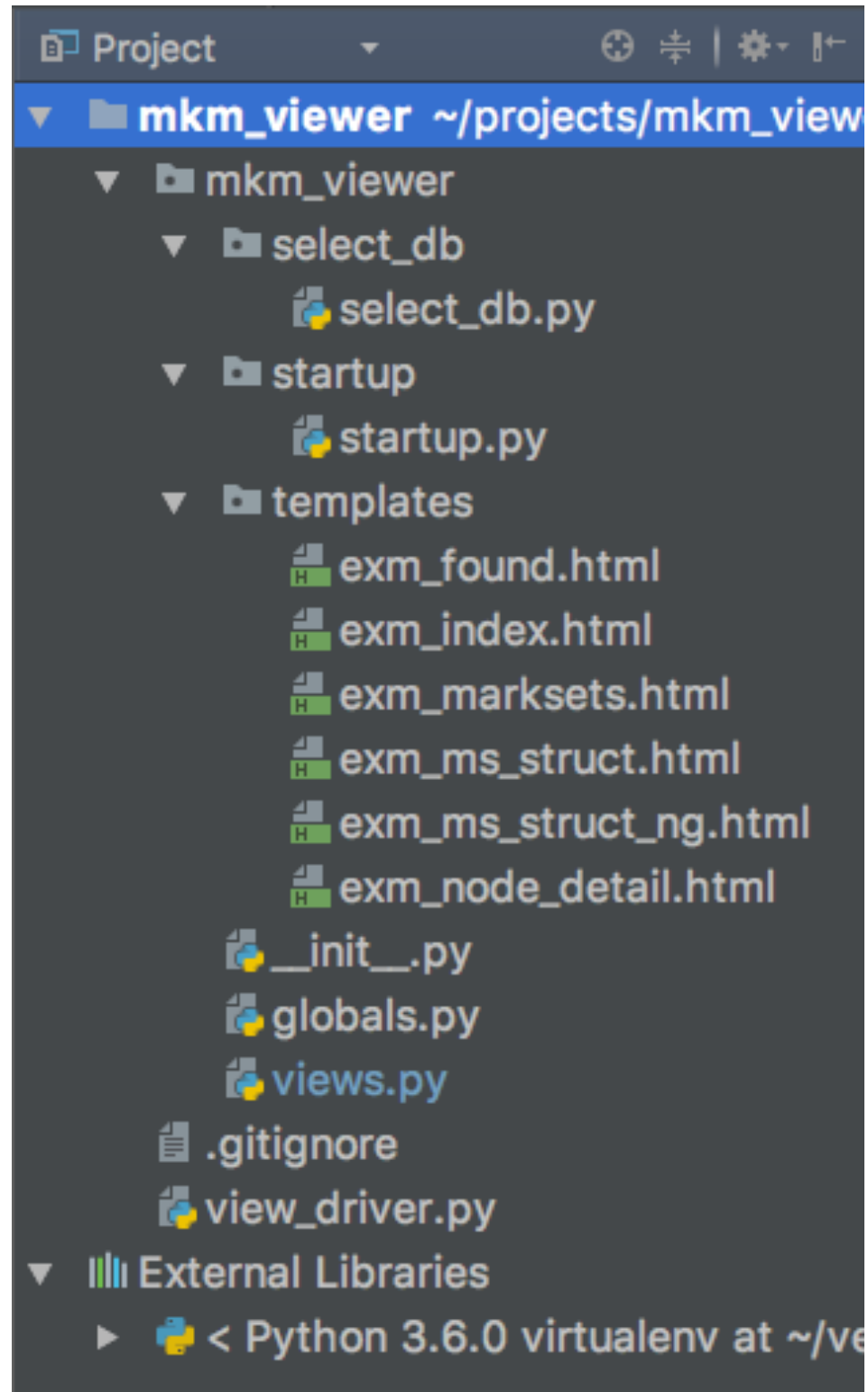
mkm_viewer

`__init__.py`

`view_driver.py`

`views.py`

`exm_node_detail.html`



Run through
example edit.

Initial Run-Through: Fill in form data

guid = 130848da-1178-4a00-8b1c-38017b5d4c41

Name

Desc

created = 2016-05-04 21:51:39.325206

edited = 2016-05-04 21:51:39.325218

mkm_viewer/mkm_viewer/__init__.py

```
from flask import Flask
```

```
from .globals import JTC
```

```
j = JTC()
```

```
from mkm_viewer.startup.startup import startup_bp
```

```
from mkm_viewer.select_db.select_db import select_db_bp
```

```
from . import globals
```

```
exm = Flask(__name__)
```

```
exm.register_blueprint(startup_bp)
```

```
exm.register_blueprint(select_db_bp)
```

mkm_viewer/view_driver.py

```
from mkm_viewer import exm  
  
from mkm_viewer import views  
from mkm_viewer import globals  
exm.secret_key = 'mkm_viewer'  
  
exm.run(port=5050, debug=True)
```

```
from flask import render_template
from flask import request
from flask import session
from . import exm
...
```

```
@exm.route('/update/<node_id>', methods=['POST'])
def update_node(node_id):
    ms = j.MS_DB[session['ms_id']]
    session['to-and-fro'] += 1
    old_node = ms.mg.odg.node[node_id]
    data_movement(old_node)
    return render_template('exm_node_detail.html',
                           ms=ms, db_id=session['db_id'],
                           ms_id=session['ms_id'], node_id=node_id)
```

```
<div style="margin-left: 40px">
  <p>guid = {{ node_id }}</p>
  <form action="/update/{{ node_id }}?a=1&b=2" method="post">
    <label for="name">Name</label>
    <input type="text" name="topic-name" value="{{ node['name'] }}">
    <br>
    <label for="desc">Desc</label>
    <input type="text" name="topic-desc" value="{{ node['desc'] }}">
    <br>
    <input type="submit" value="Submit">
  </form>
  <p>created = {{ node['created'] }}</p>
  <p>edited = {{ node['edited'] }}</p>
</div>
```

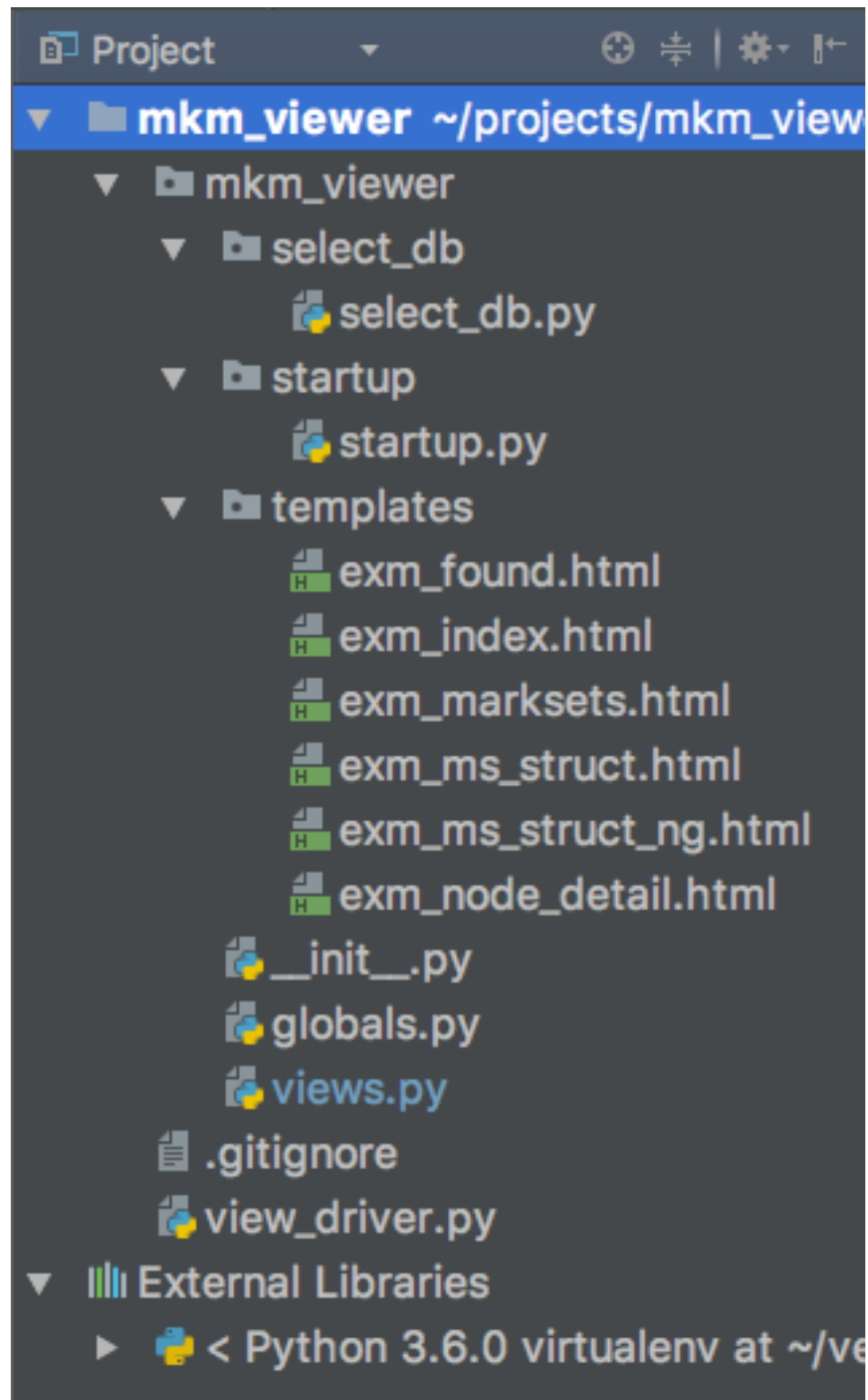
mkm_viewer

`__init__.py`

`view_driver.py`

`views.py`

`exm_node_detail.html`



**Switch PyCharm
to Debug mode**

NODE:

old_node["name"]: 'DnD'

old_node["desc"]: ''

old_node["created"]: '2016-05-04 21:51:39.325206'

old_node["edited"]: '2016-05-04 21:51:39.325218'

Start: Browser to Server

guid = 130848da-1178-4a00-8b1c-38017b5d4c41

Name

Desc

created = 2016-05-04 21:51:39.325206

edited = 2016-05-04 21:51:39.325218

Browser to Server

guid = 130848da-1178-4a00-8b1c-38017b5d4c41

Name

Desc

created = 2016-05-04 21:51:39.325206

edited = 2016-05-04 21:51:39.325218

COOKIES:

```
request.cookies["session"]="eJwlyksKwkAM..."
```

```
request.cookies["panel"]="inbox"
```

```
request.cookies["Pycharm-a4edafa9"]="4ce3e89..."
```

NOTE: `request.cookie["session"]` is not the same as `Flask.session`

`import session` refers to `Flask.session`

request.args

[templates/exm_node_detail.html]

```
<form action="/update/{{ node_id }}?a=1&b=2" method="post">  
    ...  
    <input type="submit" value="Submit">  
</form>
```

request.args

ARG values (request.args):

```
<type(request.args)=<class
```

```
  'werkzeug.datastructures.ImmutableMultiDict'>
```

```
request.args["a"]: '1'
```

```
request.args["b"]: '2'
```

SESSION values:

```
session["db_id"]="fe17c3d5-dccd-466b-b58c-..."  
session["ms_id"]="66ac372c-7687-40a7-82c8-..."  
session["to-and-fro"]=2
```

request.form

```
<form action="/update/{{ node_id }}?a=1&b=2" method="post">
  <label for="name">Name</label>
  <input type="text" name="topic-name" value="{{ node['name'] }}">
  <br>
  <label for="desc">Desc</label>
  <input type="text" name="topic-desc" value="{{ node['desc'] }}">
  <br>
  <input type="submit" value="Submit">
</form>
```

[views.py]

```
@exm.route('/update/<node_id>', methods=['POST'])
def update_node(node_id):
    ...
```

FORM values (request.form):

```
<type(request.form)=<class
```

```
    'werkzeug.datastructures.ImmutableMultiDict'>
```

```
request.form["topic-name"]: 'DnD'
```

```
request.form["topic-desc"]: 'whatshappening'
```

Recap: Browser to Server

- Change data content in `<form>` `<input>` field
- Send request to server Flask application via Submit
- `Flask request.form[<field name>]` updates server side data
- `Flask.session` information used for data context
- `Flask request.args` and `request.cookies` data available

Updated Server Side Data

UPDATED NODE:

updated_node["name"]: 'DnD'

updated_node["desc"]: 'whatshappening'

updated_node["created"]: '2016-05-04 21:51:39.325206'

updated_node["edited"]: '2016-05-04 21:51:39.325218'


```
from flask import render_template
from flask import request
from flask import session
from . import exm
...
```

```
@exm.route('/update/<node_id>', methods=['POST'])
def update_node(node_id):
    ms = j.MS_DB[session['ms_id']]
    session['to-and-fro'] += 1
    old_node = ms.mg.odg.node[node_id]
    data_movement(old_node)
    return render_template('exm_node_detail.html',
                           ms=ms, db_id=session['db_id'],
                           ms_id=session['ms_id'], node_id=node_id)
```

request.form

```
<form action="/update/{{ node_id }}?a=1&b=2" method="post">
  <label for="name">Name</label>
  <input type="text" name="topic-name" value="{{ node['name'] }}">
  <br>
  <label for="desc">Desc</label>
  <input type="text" name="topic-desc" value="{{ node['desc'] }}">
  <br>
  <input type="submit" value="Submit">
</form>
```

Recap: Server to Browser

Data passed to Jinja template in **render_template**

Python data values used in Jinja HTML template

Jinja {% for %} and {% if %} and {{ var }} used to build fully formatted HTML page

HTML page sent to browser

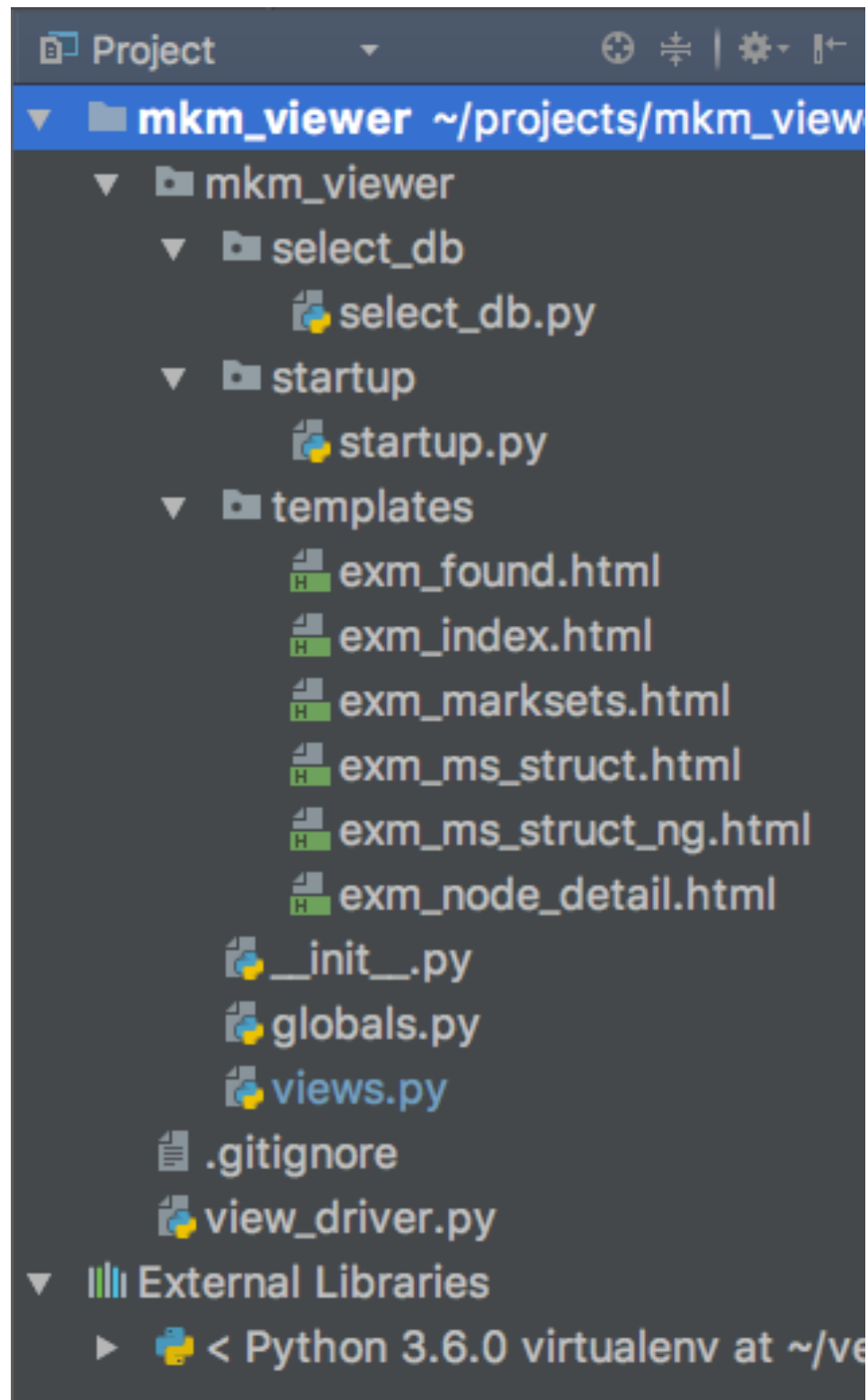
mkm_viewer

`__init__.py`

`view_driver.py`

`views.py`

`exm_node_detail.html`



Questions?

