

Google App Engine Adding a Login Screen

ae-08-login

Textbook: Using Google App Engine (Chapter 6)

The logo for open.michigan, featuring the text "open.michigan" in a lowercase, sans-serif font. The "o" in "open" is a stylized circle. Below the text are several thin, light blue wavy lines that create a sense of motion or a digital landscape.

open.michigan

Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License.

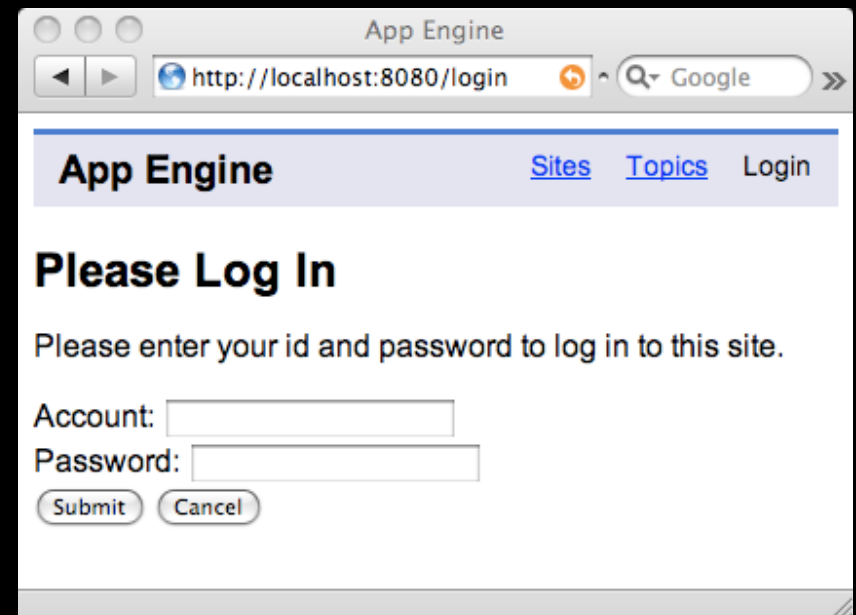
<http://creativecommons.org/licenses/by/3.0/>.

Copyright 2009, Charles Severance



Adding a Login Screen

- One of the benefits of using a “Base” template and extending it is the ability to easily add new screens
- We will quickly add a login screen to our application



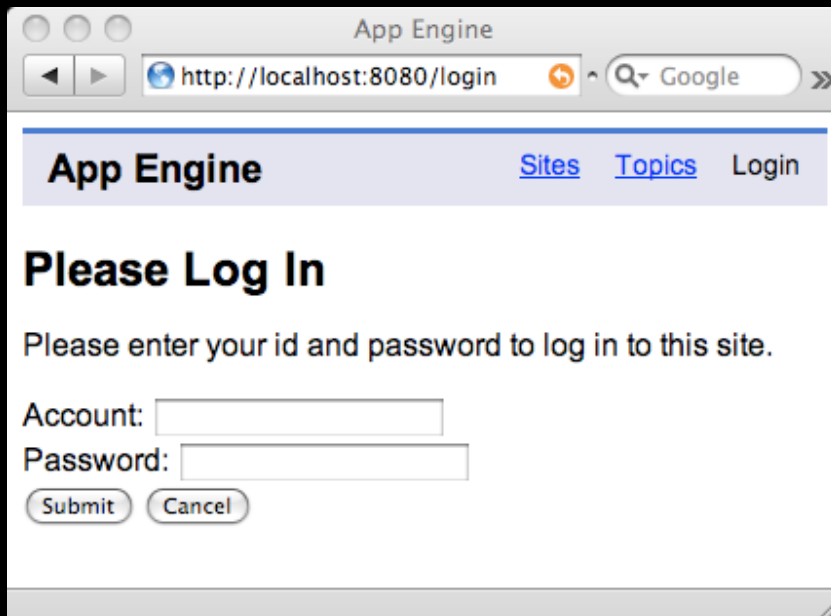
Steps

- Make a bit of utility code to do the render steps
- Determine a URL for the login screen
 - GET will display the screen
 - POST will check the password and optionally log you in
- Add a LoginHandler and add a route for /login to that handler
- Build the template for loginscreen.htm
- Build the Python code for LoginHandler

```
def doRender(handler, tname='index.htm', values={}):
    temp = os.path.join(
        os.path.dirname(__file__),
        'templates/' + tname)
    if not os.path.isfile(temp):
        return False

    # Make a copy and add the path
    newval = dict(values)
    newval['path'] = handler.request.path

    outstr = template.render(temp, newval)
    handler.response.out.write(outstr)
    return True
```



`/login` will be our path for the login screen

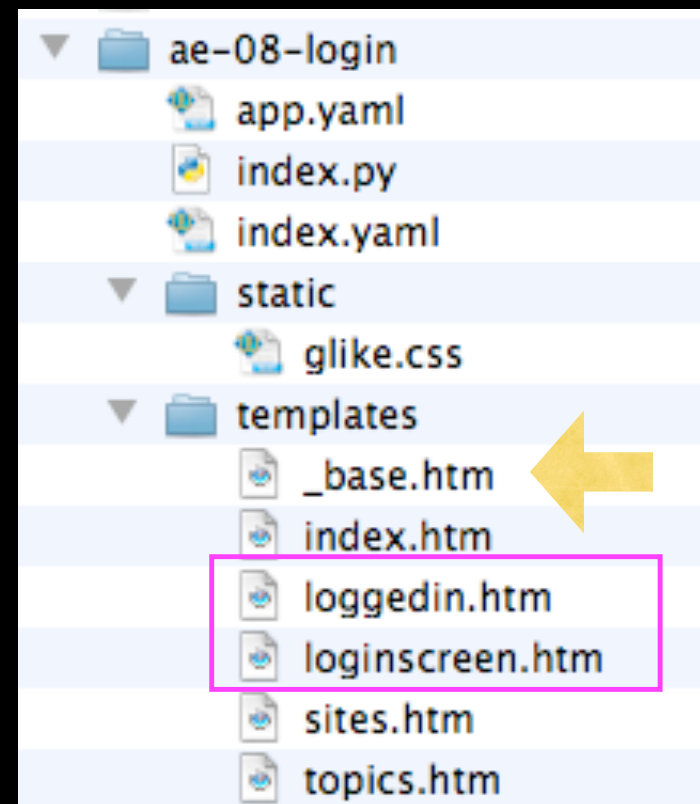
GET = display the screen

POST = check the password

```
class LoginHandler(webapp.RequestHandler):  
  
    def get(self):  
        doRender(self, 'loginscreen.htm')  
  
def main():  
    application = webapp.WSGIApplication([  
        ('/login', LoginHandler),  
        ('/*', MainHandler)],  
        debug=True)  
    wsgiref.handlers.CGIHandler().run(application)  
  
if __name__ == '__main__':  
    main()
```

Modify `_base.htm`

Add two new template files



base.htm

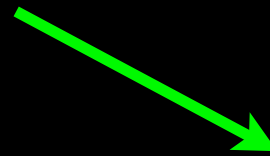
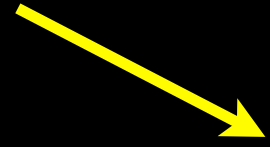
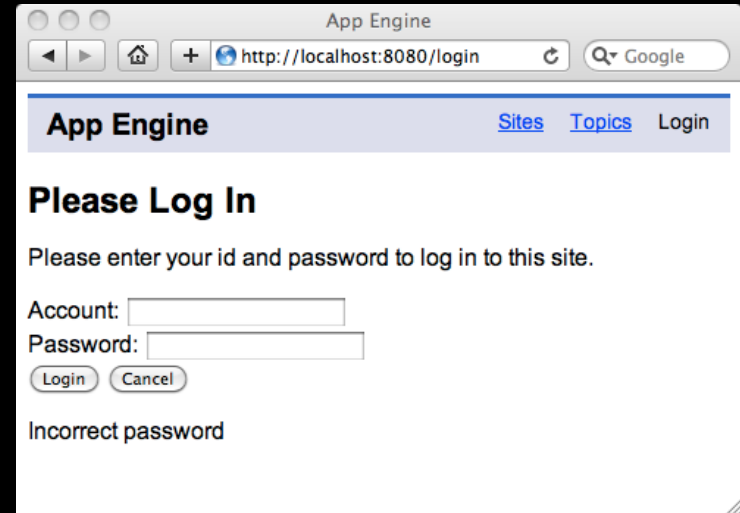
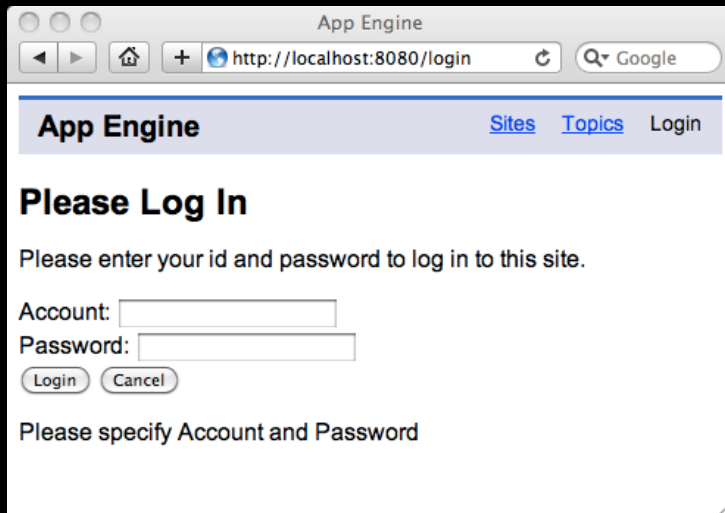
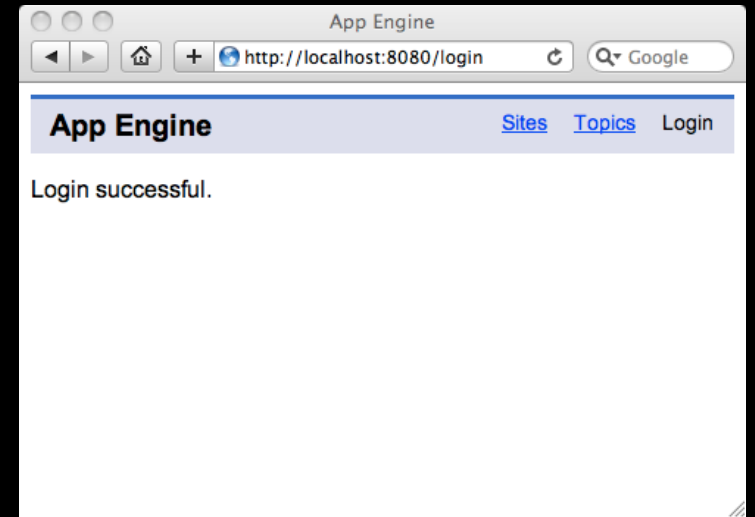
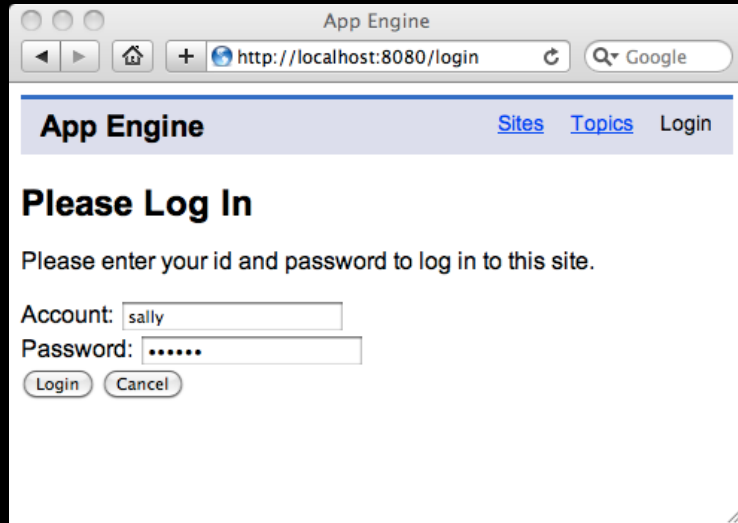
```
<li><a href="sites.htm"
    {% ifequal path '/sites.htm' %}
        class="selected"
    {% endifequal %}
>Sites</a></li>
<li><a href="topics.htm"
    {% ifequal path '/topics.htm' %}
        class="selected"
    {% endifequal %}
>Topics</a></li>
<li><a href="/login"
    {% ifequal path '/login' %}
        class="selected"
    {% endifequal %}
>Login</a></li>
</ul>
```

```
{% extends "_base.htm" %}
{% block bodycontent %}
    <h2>Please Log In</h2>
    <p>
    Please enter your id and password to log in to this site.
    </p>
    <form method="post" action="/login">
    Account: <input type="text" name="account"/> <br>
    Password: <input type="password" name="password"/> <br/>
    <input type="submit" value="Login"/>
    <input type="submit" value="Cancel"
        onclick="window.location='/'; return false;"/>
    </form>
    </p>
{% endblock %}
```

loginscreen.htm

Handling the POST

- The incoming fields come as “account” and “password”
- We check for empty fields, or bad password, and if the users knows the right password, we give them a nice message.



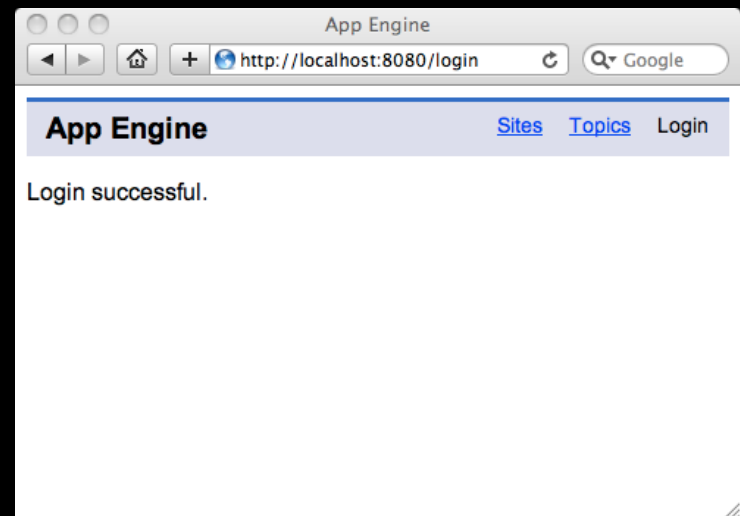
```
def post(self) :
    acct = self.request.get('account')
    pw = self.request.get('password')
    logging.info('Checking account='+acct+' pw='+pw)

    if pw == '' or acct == '':
        doRender(
            self,
            'loginscreen.htm',
            {'error' : 'Specify Acct and PW' } )
    elif pw == 'secret':
        doRender(self, 'loggedin.htm', { } )
    else:
        doRender(
            self,
            'loginscreen.htm',
            {'error' : 'Incorrect password' } )
```

```
{% extends "_base.htm" %}
{% block bodycontent %}
    <h2>Please Log In</h2>
    <p>
Please enter your id and password to log in to this site.
    </p>
    <p>
    <form method="post" action="/login">
Account: <input type="text" name="account"/> <br>
Password: <input type="password" name="password"/> <br/>
    <input type="submit" value="Login"/>
    <input type="submit" value="Cancel"
        onclick="window.location='/' ; return false;"/>
    </form>
    </p>
    {% ifnotequal error None %}
    <p>
    {{ error }}
    </p>
    {% endifnotequal %}
{% endblock %}
```

loginscreen.htm

```
{% extends "_base.htm" %}
{% block bodycontent %}
    <p>
        Login successful.
    </p>
{% endblock %}
```



loggedin.htm