EECS118

Fall 2019

Mini Project 2 - Web Application

Assigned on: 10/10/2019

In this mini-project, you will learn how to set up a web server and create a simple web application that asks the user to upload an image and input the information (title, artist, etc.) of the image. The web application should then display the above on a new page. The project is due at 11:59 pm, Thursday, 10/24/2019.

Deliverables

A .py file that is the webpage you write.

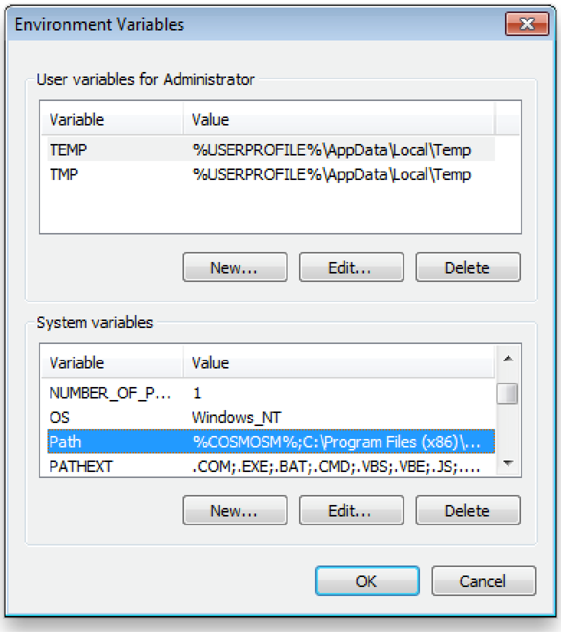
Web Programming Using CGI

A CGI (Common Gateway Interface) script is invoked by an HTTP server, usually to process user input submitted through an HTML element. Most often, CGI scripts live in the server's special cgi-bin directory. The HTTP server places all sorts of information about the request in the script's shell environment, executes the script, and sends the script's output back to the client.

STEP 1 - Install JDK & Python

You can type "java –version" in the command line to see if your system already has Java or not. If you do not have JDK or JRE on your computer, go to the JAVA download page: http://www.oracle.com/technetwork/java/javase/downloads/index.html) to download the JDK. Select the correct package for your system and download it. Execute the package to install it.

For Python, go to https://www.python.org/downloads/windows/ to download Python 3.6.8 64-bit version, not the latest one. For later mini projects, 3.6.8 is a version that has compatible libraries. Finish the installation process. Go to System Properties->Advanced system settings -> Environment Variables. add two paths, installed python file and python scripts file, under the "path" variable:
(e.g. C:\Users\USER\AppData\Local\Programs\Python\Python36\Scripts,
C:\Users\USER\AppData\Local\Programs\Python\Python36\Scripts)



Environment Variables

User variables for Administrator

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

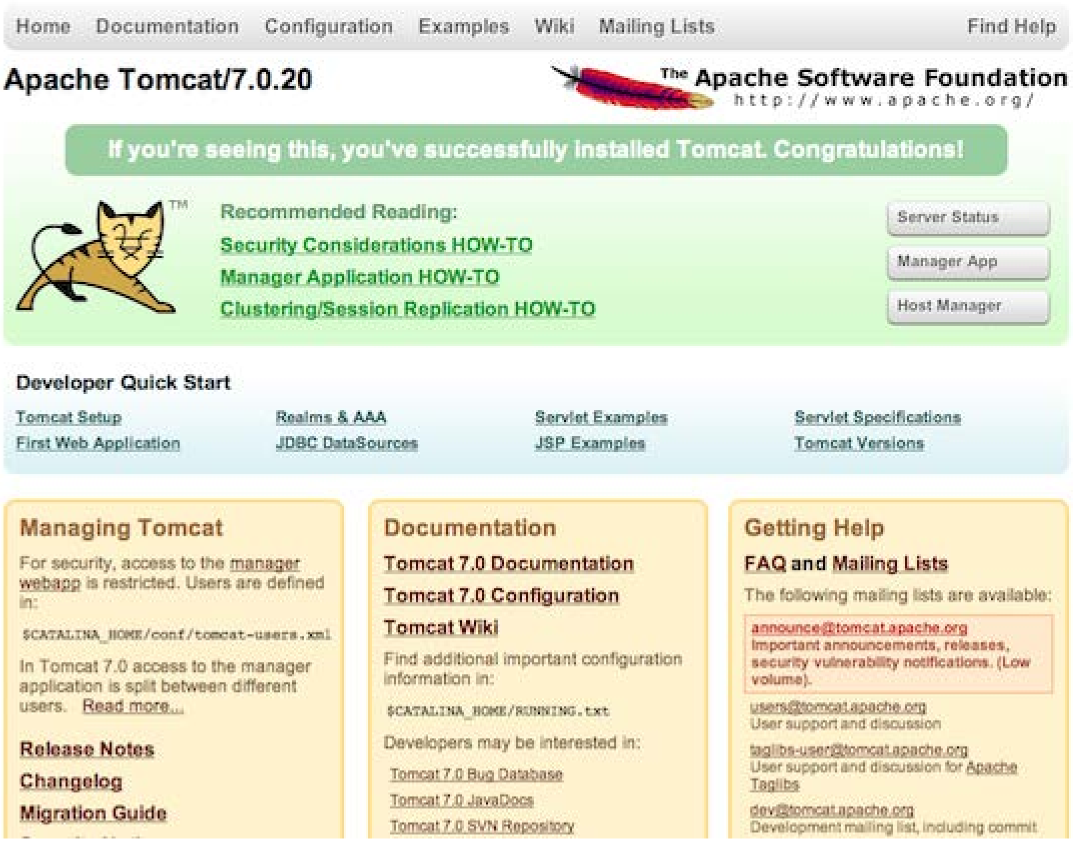
System variables

Variable	Value
NUMBER_OF_P...	1
OS	Windows_NT
Path	%COSMOSM%;C:\Program Files (x86)\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

STEP 2 – Install Apache Tomcat
Go to the Tomcat download page (http://tomcat.apache.org/) to download Tomcat 7.0. Download “tar.gz” under the “core” section. If you are using Windows, you may also choose 32-bit/64-bit Windows Service Installer which will install Tomcat as a service. Instructions for MAC can be found at: https://wolfpaulus.com/mac/tomcat/

The following steps are based on “tar.gz” on Windows 7. Extract the file to wherever you like (don’t put it in protected files, such as program file). Go to System Properties->Advanced system settings -> Environment Variables. If you are using JRE, add JRE_HOME to the path you installed the JRE (e.g., JRE_HOME=C:\Program Files (x86)\Java\jre7). If you are using JDK, add JAVA_HOME to the path you installed the JDK (e.g., JAVA_HOME = C:\Program Files\Java\jdk1.7.0_02). If you cannot find the place to set the environment variables, you may also edit setclasspath.bat in the /bin folder under your Tomcat; and add “set JRE_HOME=C:\Program Files (x86)\Java\jre7” at the beginning of the file.

Double click startup.bat in the /bin folder to start the Tomcat. If everything goes well, your will see “Server startup in xxxx ms”. Open your browser and type in “http://localhost:8080/”. You will see the following page if your Tomcat was installed successfully:



Apache Tomcat/7.0.20

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Servlet Examples
JSP Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

$CATALINA_HOME/conf/tomcat-users.xml

In Tomcat 7.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide

Documentation

Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

$CATALINA_HOME/RUNNING.txt

Developers may be interested in:

Tomcat 7.0 Bug Database
Tomcat 7.0 JavaDocs
Tomcat 7.0 SVN Repository

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

announce@tomcat.apache.org
Important announcements, releases, security vulnerability notifications. (Low volume).

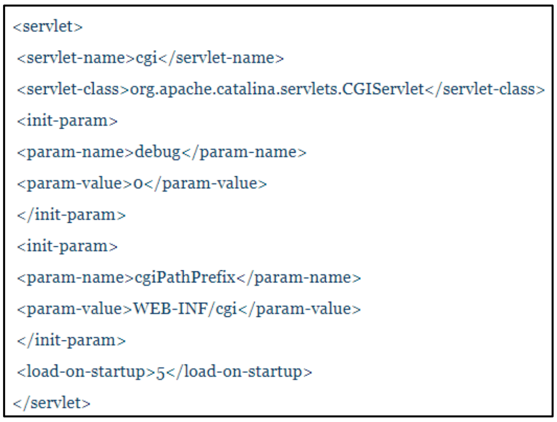
users@tomcat.apache.org
User support and discussion

taglibs-user@tomcat.apache.org
User support and discussion for Apache Taglibs

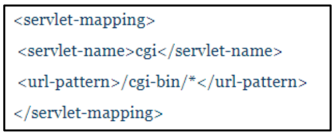
dev@tomcat.apache.org
Development mailing list, including commit

To enable CGI support:

1. modify the web.xml file. Go to the folder (should be something like this): “C:\YOUR TOMCAT PATH\apache-tomcat-7.0.96\conf”. Open the web.xml file. Find “servlet” and “servlet mapping”, then uncomment all the codes like below:

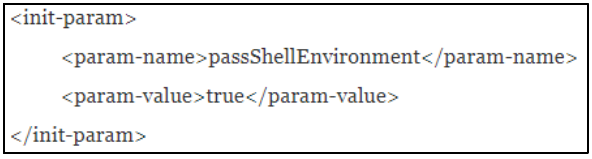


```
<servlet>
 <servlet-name>cgi</servlet-name>
 <servlet-class>org.apache.catalina.servlets.CGIServlet</servlet-class>
 <init-param>
 <param-name>debug</param-name>
 <param-value>0</param-value>
 </init-param>
 <init-param>
 <param-name>cgiPathPrefix</param-name>
 <param-value>WEB-INF/cgi</param-value>
 </init-param>
 <load-on-startup>5</load-on-startup>
</servlet>
```



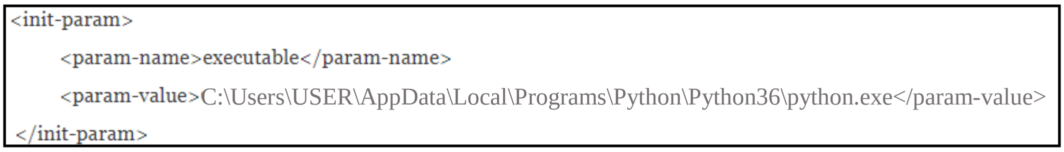
```
<servlet-mapping>
 <servlet-name>cgi</servlet-name>
 <url-pattern>/cgi-bin/*</url-pattern>
</servlet-mapping>
```

2. Add a servlet parameter "passShellEnvironment" and set it to “true” ("force" the environment variables to be passed)



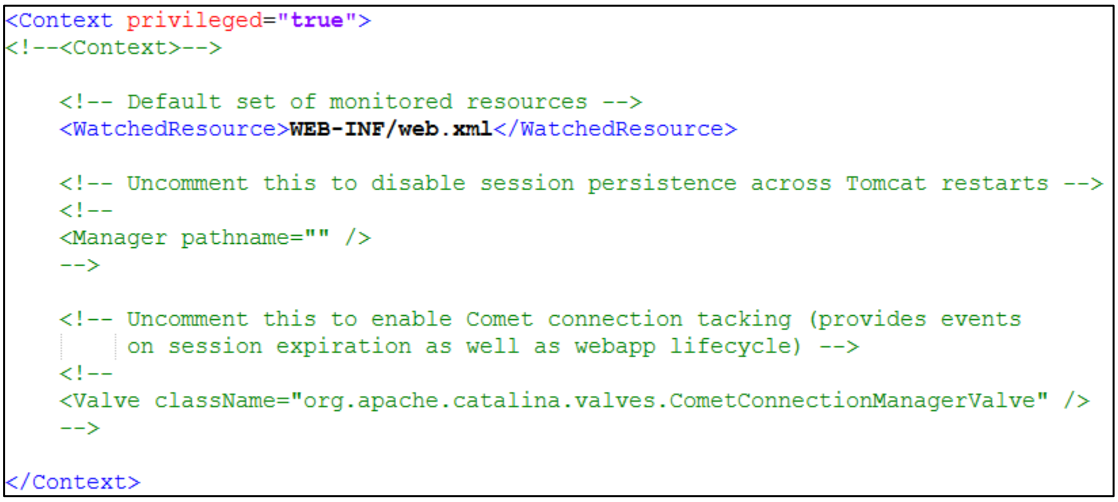
```
<init-param>
       <param-name>passShellEnvironment</param-name>
       <param-value>true</param-value>
</init-param>
```

3. Add a servlet parameter "executable"



```
<init-param>
       <param-name>executable</param-name>
       <param-value>C:\Users\USER\AppData\Local\Programs\Python\Python36\python.exe</param-value>
 </init-param>
```

4. Modify the context.xml file at “C:\YOUR TOMCAT PATH\apache-tomcat-7.0.96\conf”, add a property on <Context>:



```
<Context privileged="true">
<!--<Context>-->

    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->

    <!-- Uncomment this to enable Comet connection tacking (provides events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
    -->

</Context>
```

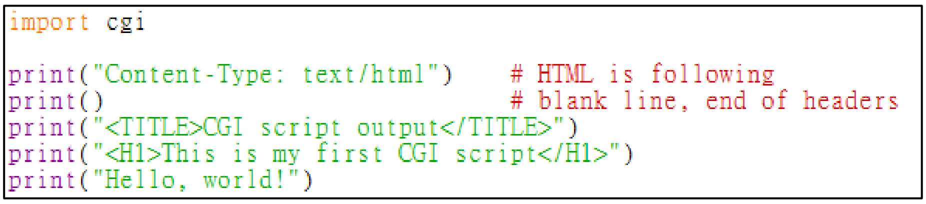
Set up Tomcat Server with CGI:

1. Go to the apache-tomcat-7.0.96\webapps folder (should be something like this): C:\YOUR TOMCAT PATH\apache-tomcat-7.0.96\webapps
2. Create a folder "test" in this folder
3. Create a “WEB-INF” folder inside the test folder
4. Create a “cgi” folder inside WEB-INF folder, your Python cgi script should be put here inside the cgi folder like this:
5. C:\YOUR TOMCAT PATH\apache-tomcat-7.0.96\webapps\test\WEB-INF\cgi

STEP 3 – Design and Implement the Web Application

Create Your First Python CGI Script (hello.py):

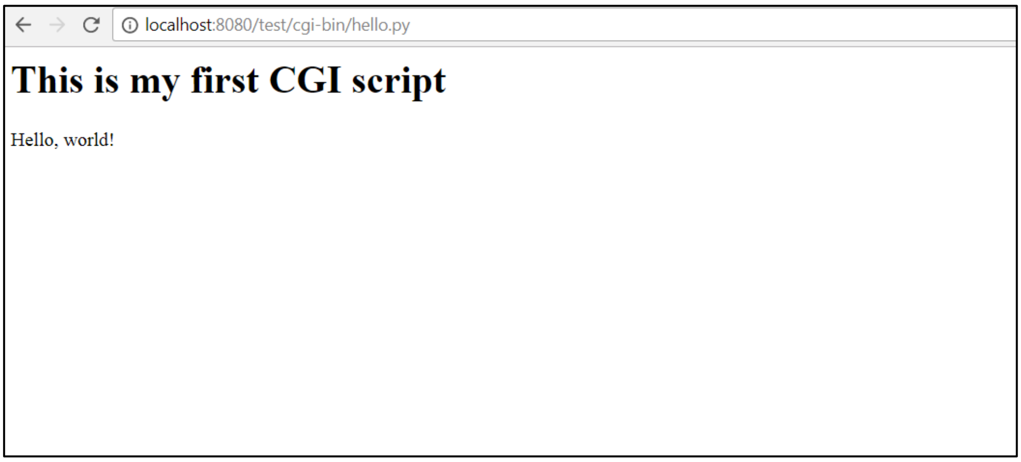
1. C:\YOUR TOMCAT PATH\apache-tomcat-7.0.96\webapps\test\WEB-INF\cgi
   Source code:



```python
import cgi

print("Content-Type: text/html")    # HTML is following
print()                             # blank line, end of headers
print("<TITLE>CGI script output</TITLE>")
print("<H1>This is my first CGI script</H1>")
print("Hello, world!")
```

2. Start Tomcat Server:
   C:\YOUR TOMCAT PATH\apache-tomcat-7.0.96\bin
   Execute startup.bat, browse the url in your web browser
   “http://localhost:8080/test/cgi-bin/hello.py”
   Results on the web browser:



This is my first CGI script

Hello, world!

3. CGI function:
   FieldStorage() class get submitted form data
   FORM = cgi.FieldStorage()
   This method always returns a list of values associated with form field name, it returns a list consisting of one item if only one such value exists.
   numbers = FORM.getlist("number")

4. Frequently Used HTML Syntax:

   <html> tag tells the browser that this is an HTML document
   <head> element is a container for all the head elements
   <body> tag defines the document's body
   <h1> defines the most important heading (Largest font)
   <h6> defines the least important heading (Smallest font)
   <b> tag specifies bold text

<p> tag defines a paragraph
<form> tag is used to create an HTML form for user input, specifies the HTTP method to use when sending form-data, is able to contain one or more elements like <input>
<input> tag specifies an input field where the user can enter data
type: specifies the value, could be number or text ...etc
name: give this input a variable name

5. Some Basic JavaScript Syntax:
   Use <script> and </script> to enclose JavaScript code block in HTML.
   To use JavaScript in Python CGI, you will need to create a JavaScript function and call it later in HTML.
   To create a function “ChangeText” to change the text in HTML:

```
<script>
function ChangeText()
{
document.getElementById("para").innerHTML = "Here is something new."";
}
</script>
```

   The following HTML has the id “para” referred above, so this text will be changed.

```
<p id="para">This will be replaced.</p>
```

   This HTML snippet creates a button which will invoke the JavaScript function ChangeText() upon clicking.

```
<button type="button" onclick='ChangeText()'>Change Text</button>
```

STEP 4 – Requirements

In this project, you are required to write the following functions in your web application:

1. Ask the user to upload two images. To simplify, you can just ask the user to input the links of the pictures instead of actually uploading them.
2. Collect the details of the images from the user, including but not limited to image tile, year, artist, and a brief description of the images.
3. Display the first image (not the link) and all the information above on a new webpage (a new Python file).
4. On the same webpage, create two buttons: “First Image” and “Second Image”. If the “Second Image” button is clicked, you should replace the first image with the second image and change the description. If the “First Image” button is clicked, the image should be switched back to the first one and change the description as well. You should use JavaScript to replace the image and text in place instead of switching between two webpages.

Please turn in your Python file in the EEE dropbox under mp2. The name of the file should be “mp2_xxxxxxxx.py”, xxxxxxxx being your student id. If you have more than one file to turn in, you should archive the files into”mp2_xxxxxxxx.zip”, xxxxxxxx being your student id.

The procedure of our testing

We will put your web application on our own server. Make sure to use a relative URL for any link of local resources. We will try to upload an image and enter image tile, year, artist, and a brief description of the image. If your application does not support uploading images, we will use some existing external URLs of images.

We will only input normal data and will not try to input a value that may exceed the column size.

If you have any special requirements for your application to run correctly, please contact the TAs.