

Templates

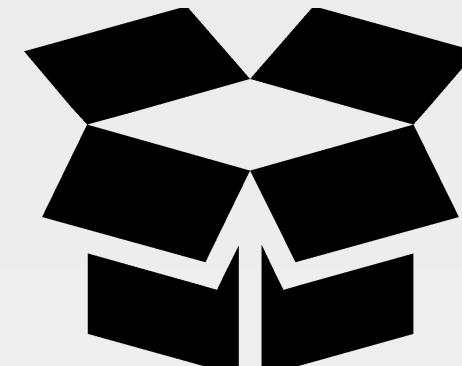
Desired State vs Data

Objectives



After completing this module, you should be able to:

- Explain when to use a template resource
- Create a template file
- Use ERB tags to display node data in a template
- Define a template resource



Cleaner Recipes

Previously, we updated our 'myiis' web server cookbook to display information about our node.

We added the following content to the file resource in the server recipe....

Demo: The 'myiis' Server Recipe

```
~/cookbooks/myiis/recipes/server.rb
```

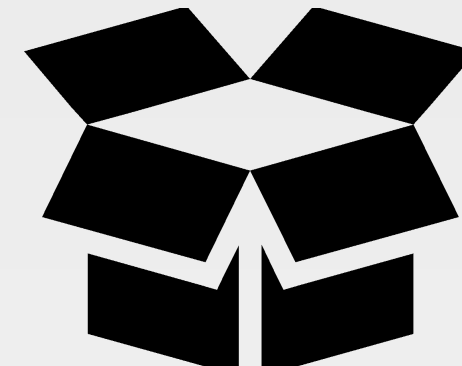
```
powershell_script 'Install IIS' do
  code 'Add-WindowsFeature Web-Server'
end

file 'c:\inetpub\wwwroot\Default.htm' do
  content "<h1>Hello, world!</h1>
<h2>PLATFORM: #{node['platform']}</h2>
<h2>HOSTNAME: #{node['hostname']}</h2>
<h2>MEMORY:    #{node['memory']['total']}</h2>
<h2>CPU Mhz:    #{node['cpu']['0']['mhz']}</h2>"
end

service 'w3svc' do
  action [:enable, :start]
end
```

When you consider the syntax needed to edit this file, like when to use double quotes, or single quotes, etc., manually editing a file like this can become difficult to do.

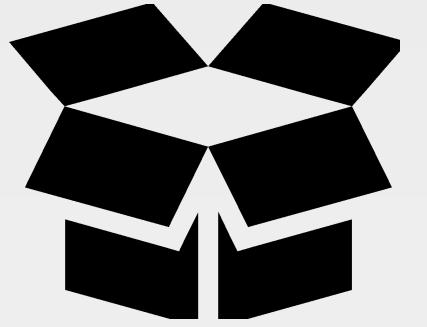
Also, Ruby string interpolation to insert the value of a variable works for simple variables but doesn't allow you to insert any logic into what value should be used, or when.



Manual Editing

This manual process of editing the recipe file is definitely error prone. Especially because a human has to edit the file again before it is deployed.

CONCEPT



What We Need

We need the ability to store the data in another file, which is in the native format of the file we are writing out but that still allows us to insert Ruby code...

...specifically, the node attributes we have defined.



Template

A cookbook template is an Embedded Ruby (ERB) template that is used to generate files. Templates may contain Ruby expressions and statements.

Use the template resource to add cookbook templates to recipes; place the corresponding Embedded Ruby (ERB) template in a cookbook's /templates directory.

<https://docs.chef.io/resources/template/>

EXAMPLE: Template File's Source Matches Up

```
> tree /f cookbooks\myiis\templates\default
Folder PATH listing
Volume serial number is B04A-119C
C:\USERS\ADMINISTRATOR\COOKBOOKS\MYIIS\TEMPLATES
```

Default.htm.erb

No subfolders exist

```
template C:\inetpub\wwwroot\Default.htm' do
```

```
  source 'Default.htm.erb'
```

```
end
```


DOCS

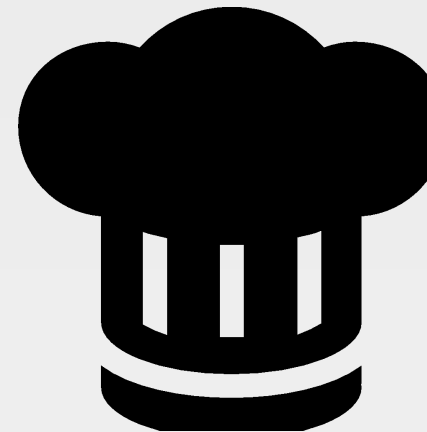
Template



To use a template, two things must happen:

1. A template resource must be added to a recipe
2. An Embedded Ruby (ERB) template must be added to a cookbook

<https://docs.chef.io/resources/template/>



GL: Cleaner Recipes

Adding the node attributes to the index page did make it harder to read the recipe.

Objective:

- ☐ Create a template with chef generate
- ☐ Define the contents of the ERB template
- ☐ Change the file resource to the template resource in the 'myiis' cookbook
- ☐ Use chef-client to apply the "myiis" cookbook's "default" recipe
- ☐ Update the "myiis" cookbook's version for this patch

GL: What Can chef generate Do?



```
> chef generate --help
```

```
Usage: chef generate GENERATOR [options]
```

```
Available generators:
```

app	Generate an application repo
cookbook	Generate a single cookbook
recipe	Generate a new recipe
attribute	Generate an attributes file
template	Generate a file template
file	Generate a cookbook file
lwrp	Generate a lightweight resource/provider
repo	Generate a Chef policy repository
policyfile	Generate a Policyfile for use with the install/push commands

GL: What Can chef generate template Do?



```
> chef generate template --help
```

```
Usage: chef generate template [path/to/cookbook] NAME [options]
```

<code>-C, --copyright COPYRIGHT</code>	Name of the copyright holder - defaults to 'The Authors'
<code>-m, --email EMAIL</code>	Email address of the author - defaults to ...
<code>-a, --generator-arg KEY=VALUE</code>	Use to set arbitrary attribute KEY to VALUE in the
<code>-I, --license LICENSE</code>	all_rights, mit, gplv2, gplv3 - defaults to
<code>-s, --source SOURCE_FILE</code>	Copy content from SOURCE_FILE
<code>-g GENERATOR_COOKBOOK_PATH,</code> <code>code_generator</code>	Use GENERATOR_COOKBOOK_PATH for the
<code>--generator-cookbook</code>	

GL: Use chef to Generate a Template



```
> cd ~  
> chef generate template cookbooks\myiis Default.htm
```

```
Recipe: code_generator::template
```

```
* directory[.\cookbooks/myiis/templates/default] action create
```

```
- create new directory .\cookbooks/myiis/templates/default
```

```
* template[.\cookbooks/myiis/templates/Default.htm.erb] action create
```

```
- create new file .\cookbooks/myiis/templates/Default.htm.erb
```

```
- update content in file .\cookbooks/myiis/templates/Default.htm.erb from  
none to e3b0c4
```

```
(diff output suppressed by config)
```

GL: Lets Look at the Template File



```
> tree /f cookbooks\myiis\templates
```

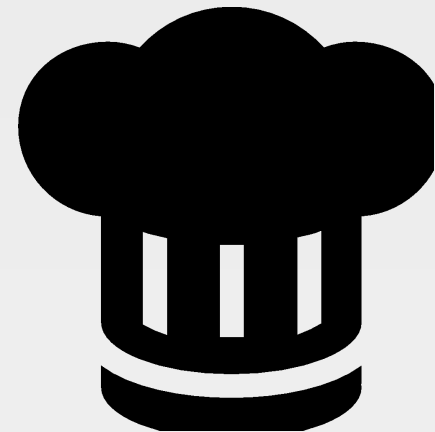
```
Folder PATH listing
```

```
Volume serial number is E04D-394D
```

```
C:\USERS\ADMINISTRATOR\COOKBOOKS\MYIIS\TEMPLATES
```

```
Default.htm.erb
```

```
No subfolders exist
```



GL: Cleaner Recipes

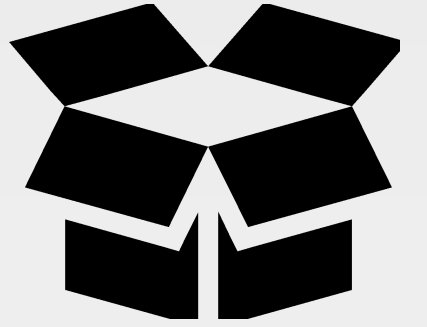
Adding the node attributes to the index page did make it harder to read the recipe.

Objective:

- ✓ Create a template with chef generate
- ☐ Define the contents of the ERB template
- ☐ Change the file resource to the template resource in the 'myiis' cookbook
- ☐ Use chef-client to apply the "myiis" cookbook's "default" recipe
- ☐ Update the "myiis" cookbook's version for this patch

CONCEPT

ERB



An Embedded Ruby (ERB) template allows Ruby code to be embedded inside a text file within specially formatted tags.

Ruby code can be embedded using expressions and statements.

<https://docs.chef.io/templates.html>

Text Within an ERB Template

```
<% if (50 + 50) == 100 -%>
```

```
50 + 50 = <%= 50 + 50 %>
```

```
<% else -%>
```

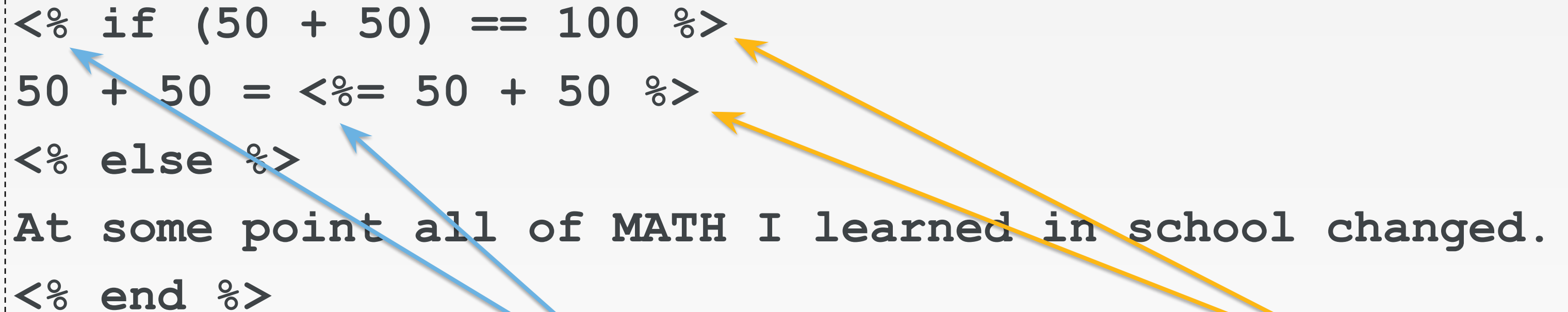
```
At some point all of MATH I learned in school changed.
```

```
<% end -%>
```

Each ERB tag has a beginning tag and a matched ending tag.

Text Within an ERB Template

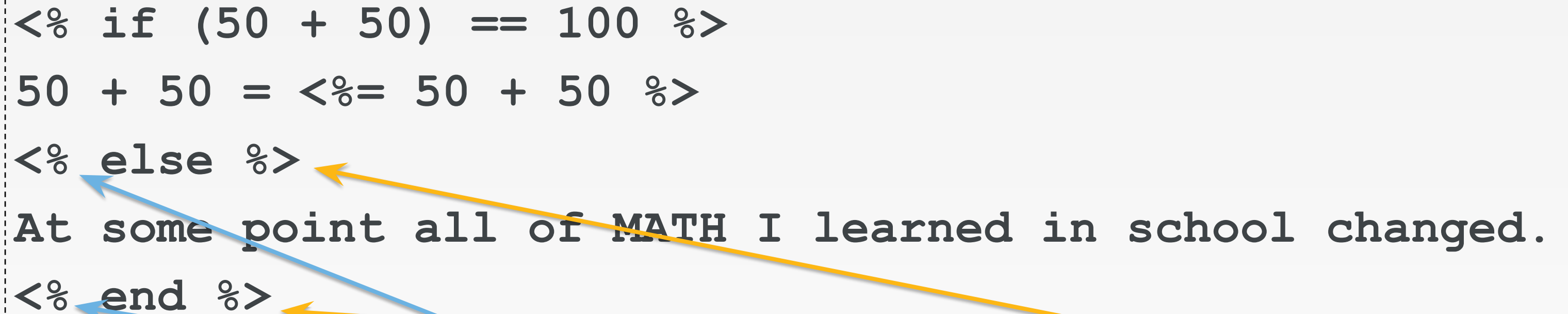
```
<% if (50 + 50) == 100 %>  
50 + 50 = <%= 50 + 50 %>  
<% else %>  
At some point all of MATH I learned in school changed.  
<% end %>
```

A diagram illustrating ERB tag matching. Two blue arrows originate from the opening tags of the 'if' and 'else' blocks and point to the closing 'end' tag. Two yellow arrows originate from the opening 'end' tag and point to the closing tags of the 'if' and 'else' blocks, demonstrating how the parser matches the end of a block to its corresponding opening tag.

Each ERB tag has a beginning tag and a matched ending tag.

Text Within an ERB Template

```
<% if (50 + 50) == 100 %>  
50 + 50 = <%= 50 + 50 %>  
<% else %>  
At some point all of MATH I learned in school changed.  
<% end %>
```

A diagram illustrating ERB tag matching. It shows a code block with an if-else structure. Blue arrows point from the opening tags to the closing tags: from <% if to <% end, and from <% else to <%= 50 + 50 %. Yellow arrows point from the closing tags to the opening tags: from <%= 50 + 50 % to <% else, and from <% end to <% if. This visualizes how the ERB engine matches opening and closing tags to process the template.

Each ERB tag has a beginning tag and a matched ending tag.

Text Within an ERB Template

```
<% if (50 + 50) == 100 %>
```

```
50 + 50 = <%= 50 + 50 %>
```

```
<% else %>
```

```
At some point all of MATH I learned in school changed.
```

```
<% end %>
```

Executes the Ruby code within the brackets and does not display the result.

Text Within an ERB Template

```
<% if (50 + 50) == 100 %>
```

```
50 + 50 = <%= 50 + 50 %>
```

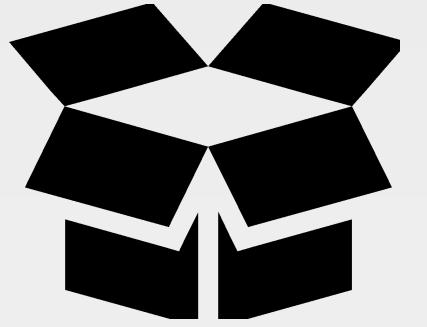
```
<% else %>
```

```
At some point all of MATH I learned in school changed.
```

```
<% end %>
```

Executes the Ruby code within the brackets and displays the results.

CONCEPT



The Angry Squid

< % =

GL: Move Our server.rb Source to the Template

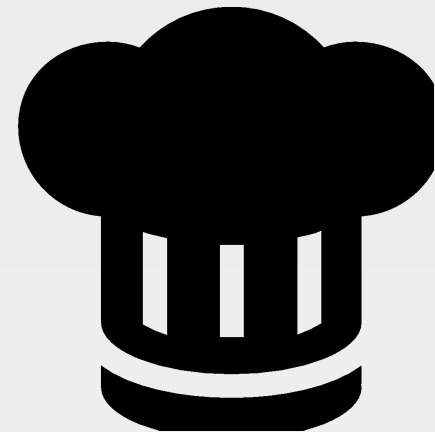
`~\cookbooks\myiis\templates\Default.htm.erb`

```
<html>
  <body>
    <h1>Hello, world!</h1>
    <h2>PLATFORM:  #{node['platform']}</h2>
    <h2>HOSTNAME:   #{node['hostname']}</h2>
    <h2>MEMORY:     #{node['memory']['total']}</h2>
    <h2>CPU Mhz:    #{node['cpu']['0']['mhz']}</h2>"
  </body>
</html>
```

GL: Replace String Interpolation with ERB

~\cookbooks\myiis\templates\Default.htm.erb

```
<html>
  <body>
    <h1>Hello, world!</h1>
    <h2>PLATFORM: <%= node['platform'] %></h2>
    <h2>HOSTNAME: <%= node['hostname'] %></h2>
    <h2>MEMORY: <%= node['memory']['total'] %></h2>
    <h2>CPU Mhz: <%= node['cpu']['0']['mhz'] %></h2>
  </body>
</html>
```

GL: Cleaner Recipes

Adding the node attributes to the index page did make it harder to read the recipe.

Objective:

- ✓ Create a template with chef generate
- ✓ Define the contents of the ERB template
- ☐ Change the file resource to the template resource in the 'myiis' cookbook
- ☐ Use chef-client to apply the "myiis" cookbook's "default" recipe
- ☐ Update the "myiis" cookbook's version for this patch

GL: Remove the Existing Content Attribute from server.rb

```
~\cookbooks\myiis\recipes\server.rb
```

```
# ... POWERSHELL_SCRIPT RESOURCE ...

file 'c:\inetpub\wwwroot\Default.htm' do
  content "<h1>Hello, world!</h1>
<h2>ipaddress: #{node['ipaddress']}</h2>
<h2>hostname: #{node['hostname']}</h2>
<h2>total memory: #{node['memory']['total']}</h2>
<h2>CPU Mhz: #{node['cpu']['0']['mhz']}</h2>"
end

# ... SERVICE RESOURCE ...
```

GL: Change the File Resource to a Template

```
~\cookbooks\myiis\recipes\server.rb
```

```
# ... POWERSHELL_SCRIPT RESOURCE ...  
  
template 'c:\inetpub\wwwroot\Default.htm' do  
  
end  
  
# ... SERVICE RESOURCE ...
```

GL: Change the File Resource to a Template

```
~\cookbooks\myiis\recipes\server.rb
```

```
template 'c:\inetpub\wwwroot\Default.htm' do  
end
```

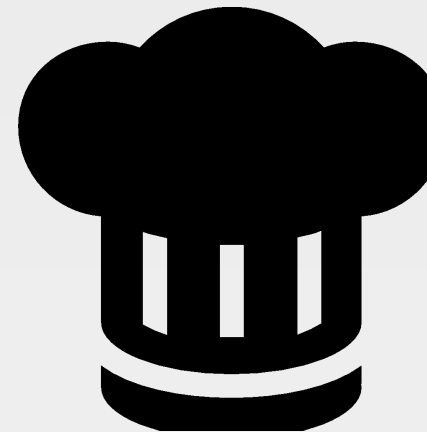
GL: Completed server.rb Code

```
~\cookbooks\myiis\recipes\server.rb
```

```
powershell_script 'Install IIS' do
  code 'Add-WindowsFeature Web-Server'
end

template 'C:\inetpub\wwwroot\Default.htm' do
  source 'Default.htm.erb'
end

service 'w3svc' do
  action [:enable, :start]
end
```



GL: Cleaner Recipes

Adding the node attributes to the index page did make it harder to read the recipe.

Objective:

- ✓ Create a template with chef generate
- ✓ Define the contents of the ERB template
- ✓ Change the file resource to the template resource in the 'myiis' cookbook
- ☐ Use chef-client to apply the "myiis" cookbook's "default" recipe
- ☐ Update the "myiis" cookbook's version for this patch

GL: Change Directories and Apply the Cookbook



```
> cd ~
```

```
> chef-client --local-mode -r "recipe[myiis]"
```

```
Starting Chef Infra Client, version 17.3.48
```

```
resolving cookbooks for run list: ["myiis"]
```

```
Synchronizing Cookbooks:
```

```
- myiis (0.2.0)...
```

```
template[c:\inetpub\wwwroot\Default.htm] action create
```

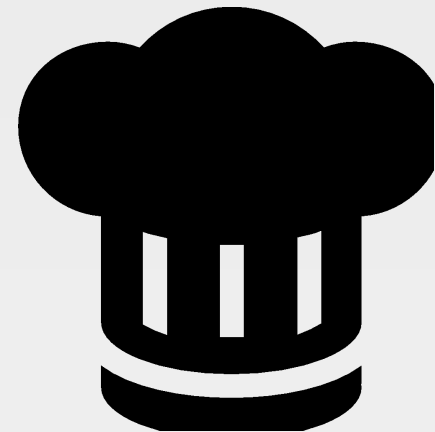
```
- update content in file c:\inetpub\wwwroot\Default.htm from ffa7b2 to d7b236
```

```
...
```

```
Running handlers:
```

```
Running handlers complete
```

```
Chef Client finished, 2/4 resources updated in 06 seconds
```



GL: Cleaner Recipes

Adding the node attributes to the index page did make it harder to read the recipe.

Objective:

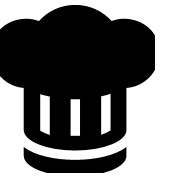
- ✓ Create a template with chef generate
- ✓ Define the contents of the ERB template
- ✓ Change the file resource to the template resource in the 'myiis' cookbook
- ✓ Use chef-client to apply the "myiis" cookbook's "default" recipe
- ☐ Update the "myiis" cookbook's version for this patch

GL: Update the Cookbook's Patch Number

```
~\cookbooks\myiis\metadata.rb
```

```
name                'myiis'  
maintainer           'The Authors'  
maintainer_email     'you@example.com'  
license              'all_rights'  
description          'Installs/Configures myiis'  
long_description     'Installs/Configures myiis'  
version              '0.2.1'
```

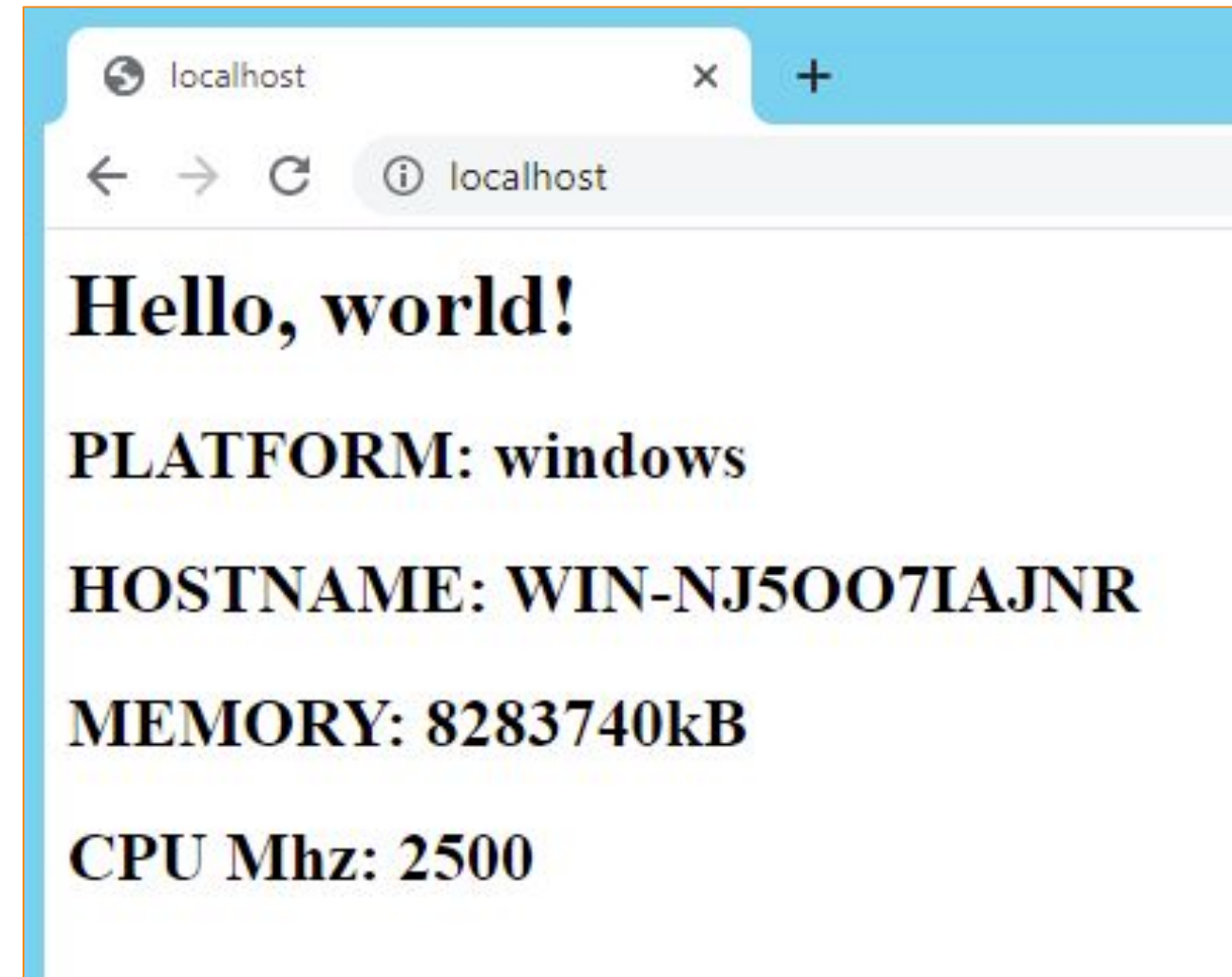
GL: Test the Cookbook's Output

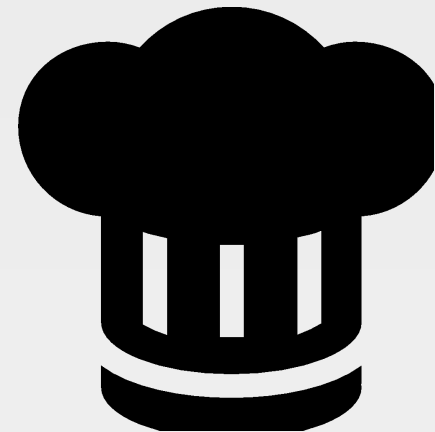


Point a web browser to `localhost` to ensure your cookbook works as expected.

Or you can type this at the command line:

Invoke-WebRequest localhost





GL: Cleaner Recipes

Adding the node attributes to the index page did make it harder to read the recipe.

Objective:

- ✓ Create a template with chef generate
- ✓ Define the contents of the ERB template
- ✓ Change the file resource to the template resource in the 'myiis' cookbook
- ✓ Use chef-client to apply the "myiis" cookbook's "default" recipe
- ✓ Update the "myiis" cookbook's version for this patch



Summary

In previous modules we were using Ruby string interpolation to insert the value of a variable.

This works for simple variables but doesn't allow you to insert any logic into what value should be used, or when.



Summary

In previous modules we were also using a ``file`` resource in our recipe file. But that isn't scalable past anything more than a few words or lines.

Once you have more than that, it can become unmanageable, especially if there are characters that you need to escape, like ``\`, `/`, ```, ```` etc. So it's better to have the logic in one file (the recipe) and the data in another (the template).



Review Questions

1. What is the benefit of using a template over defining the content within a recipe?
2. What are the drawbacks of using a template over defining the content within a recipe?
3. What is an ERB template?



Q&A

What questions can we help you answer?

- Resources
- Templates
- ERB



CHEF™