

## Objectives:

- Write test cases to test recipes with a faster response time
- Understand the importance and limitations of unit testing
- Verify tests using ChefSpec

1. Execute `$ tree spec/` from the root directory of the apache cookbook, to validate spec directory.

Expected output:

```
ubuntu@ip-172-31-62-237:~/chef-repo/cookbooks/apache$ tree spec/  
spec/  
├── spec_helper.rb  
└── unit  
    ├── recipes  
    └── default_spec.rb  
  
2 directories, 2 files
```

NOTE : When generating recipe files using the chef command, the 'spec/unit/recipes' directories are also generated. The ChefSpec defined specifications are all contained within the spec directory.

2. Open `apache/spec/unit/recipe/default_spec.rb`:

```
# Cookbook:: apache
# Spec:: default
#
# Copyright:: 2021, The Authors, All Rights Reserved.

require 'spec_helper'

describe 'apache::default' do
  context 'When all attributes are default, on Ubuntu 20.04' do
    # for a complete list of available platforms and versions see:
    # https://github.com/chefspec/fauxhai/blob/master/PLATFORMS.md
    platform 'ubuntu', '20.04'

    it 'converges successfully' do
      expect { chef_run }.to_not raise_error
    end
  end

  context 'When all attributes are default, on CentOS 8' do
    # for a complete list of available platforms and versions see:
    # https://github.com/chefspec/fauxhai/blob/master/PLATFORMS.md
    platform 'centos', '8'

    it 'converges successfully' do
      expect { chef_run }.to_not raise_error
    end
  end
end
```

Note the two existing tests, showing that it expects the Ubuntu and Centos test environments to **NOT** raise an error

3. The expectations defined in this initially generated specification file should look a little familiar, because ChefSpec is built on Rspec. Similar to how InSpec is built. ChefSpec requires a little more setup as it creates an in-memory execution.
4. Exit out of the file
5. From `~/chef-repo/cookbooks/apache` **execute**:

```
$ chef exec rspec spec/unit/recipes/default_spec.rb
```

6. Expected output:

The two tests (showing that Ubuntu and Centos test environments do not raise an error) are passing

```
..  
  
Finished in 1.62 seconds (files took 1.68 seconds to load)  
2 examples, 0 failures
```

7. There are 4 resources in the recipe (package, file, service enable and service start). We need to write unit tests for each of these 4 resources:

```
package 'httpd'  
  
file '/var/www/html/index.html' do  
  content '<h1>Welcome Home!</h1>'  
end  
  
service 'httpd' do  
  action [:enable, :start]  
end
```

- i. The first test will check if the package is **installed**.
- ii. The second test will confirm whether the index.html file has been created.
- iii. The third test will check if the 'httpd' service is **enabled** and **started**

8. Update the **spec/unit/recipes/default\_spec.rb** file under a CentOS platform as described below:

- a. Note, we are changing the centos version to 7, to match our Inspec test

```

require 'spec_helper'

describe 'apache::default' do
  context 'When all attributes are default, on Ubuntu 20.04' do
    # for a complete list of available platforms and versions see:
    # https://github.com/chefspec/fauxhai/blob/master/PLATFORMS.md
    platform 'ubuntu', '20.04'

    it 'converges successfully' do
      expect { chef_run }.to_not raise_error
    end
  end

  context 'When all attributes are default, on CentOS 8' do
    # for a complete list of available platforms and versions see:
    # https://github.com/chefspec/fauxhai/blob/master/PLATFORMS.md
    platform 'centos', '7'

    it 'converges successfully' do
      expect { chef_run }.to_not raise_error
    end

    it 'installs the httpd package'
  end
end
end

```

9. Save the file and execute the following command:

```
$ chef exec rspec spec/unit/recipes/default_spec.rb
```

Expected output :

```

*
Pending: (Failures listed here are expected and do not affect your suite's status)

  1) apache::default When all attributes are default, on CentOS 8 installs the httpd package
     # Not yet implemented
     # ./spec/unit/recipes/default_spec.rb:29

Finished in 1.57 seconds (files took 1.12 seconds to load)
3 examples, 0 failures, 1 pending

```

*NOTE: When executing 'rspec' again we should see the new pending example that we defined within the specification file.*

RSpec's pending summary is similar to the failure summary. The pending examples are identified and then they are collected together in a list.

Also, the summary is displaying that an additional example has been added and it is displayed as pending.

10. Update the default\_spec.rb file as follows:

```
require 'spec_helper'

describe 'apache::default' do
  context 'When all attributes are default, on Ubuntu 20.04' do
    # for a complete list of available platforms and versions see:
    # https://github.com/chefspec/fauxhai/blob/master/PLATFORMS.md
    platform 'ubuntu', '20.04'

    it 'converges successfully' do
      expect { chef_run }.to_not raise_error
    end
  end

  context 'When all attributes are default, on CentOS 8' do
    # for a complete list of available platforms and versions see:
    # https://github.com/chefspec/fauxhai/blob/master/PLATFORMS.md
    platform 'centos', '8'

    it 'converges successfully' do
      expect { chef_run }.to_not raise_error
    end

    it 'installs the httpd package' do
      expect(chef_run).to install_package('httpd')
    end
  end
end
```

- a. Executing 'rspec' one more time should show the previous pending example now as a passing example (it is no longer pending).

```
$ chef exec rspec spec/unit/recipes/default_spec.rb
```

```
...
```

```
Finished in 1.58 seconds (files took 1.18 seconds to load)  
3 examples, 0 failures
```

11. Cause a failure. Comment out `package 'httpd'` in `recipe/default.rb`

```
# package 'httpd'  
file '/var/www/html/index.html' do  
  content '<h1>Welcome Home!</h1>'  
end  
  
service 'httpd' do  
  action [:enable, :start]  
end
```

12. Execute `rspec`:

```
$ chef exec rspec spec/unit/recipes/default_spec.rb
```

13. Expected output is two dots in the upper left corner followed by an uppercase F and 3 examples with 1 failure:

```
..F
```

```
Failures:
```

```
1) apache::default When all attributes are default, on CentOS 8 installs the httpd package  
Failure/Error: expect(chef_run).to install_package('httpd')  
  
expected "package[httpd]" with action :install to be in Chef run. Other package resources:
```

```
# ./spec/unit/recipes/default_spec.rb:30:in `block (3 levels) in <top (required)>'
```

```
Finished in 1.69 seconds (files took 1.17 seconds to load)
```

```
3 examples, 1 failure
```

```
Failed examples:
```

```
rspec ./spec/unit/recipes/default_spec.rb:29 # apache::default When all attributes are default, on CentOS 8  
installs the httpd package
```

14. Update the specification file to fix the error causing the test to fail.

15. Test again. You should get:

```
...  
  
Finished in 1.58 seconds (files took 1.18 seconds to load)  
3 examples, 0 failures
```

16. Update default\_spec.rb as follows, adding two new pending tests:

```
require 'spec_helper'  
  
describe 'apache::default' do  
  context 'When all attributes are default, on Ubuntu 20.04' do  
    # for a complete list of available platforms and versions see:  
    # https://github.com/chefspec/fauxhai/blob/master/PLATFORMS.md  
    platform 'ubuntu', '20.04'  
  
    it 'converges successfully' do  
      expect { chef_run }.to_not raise_error  
    end  
  end  
  
  context 'When all attributes are default, on CentOS 8' do  
    # for a complete list of available platforms and versions see:  
    # https://github.com/chefspec/fauxhai/blob/master/PLATFORMS.md  
    platform 'centos', '8'  
  
    it 'converges successfully' do  
      expect { chef_run }.to_not raise_error  
    end  
  
    it 'installs the httpd package' do  
      expect(chef_run).to install_package('httpd')  
    end  
  
    it 'creates the index file'  
    it 'starts the httpd service'  
  
  end  
end
```

17. Save this file and execute rspec.

NOTE: Make sure you uncommented the httpd installation step in the default recipe.

```
$ chef exec rspec spec/unit/recipes/default_spec.rb
```

```
...**
Pending: (Failures listed here are expected and do not affect your suite's status)

  1) apache::default When all attributes are default, on CentOS 8 creates the index file
     # Not yet implemented
     # ./spec/unit/recipes/default_spec.rb:33

  2) apache::default When all attributes are default, on CentOS 8 starts the httpd service
     # Not yet implemented
     # ./spec/unit/recipes/default_spec.rb:34

Finished in 1.55 seconds (files took 1.15 seconds to load)
5 examples, 0 failures, 2 pending
```

18. Implement the tests by adding the expectations for each of your new tests:

```
it 'creates the index file' do
  expect(chef_run).to render_file('/var/www/html/index.html').with_content('<h1>Welcome Home!</h1>')
end

it 'starts and enables the httpd service' do
  expect(chef_run).to start_service('httpd')
  expect(chef_run).to enable_service('httpd')
end
```

19. Execute your ChefSpec tests again

Expected output:

```
.....

Finished in 1.63 seconds (files took 1.16 seconds to load)
5 examples, 0 failures
```




20. In order to better troubleshoot issues, purposely introduce an error in recipes/default.rb:

- a. Change the `</h1>` tag to `</h2>`

```
package 'httpd'

file '/var/www/html/index.html' do
  content '<h1>Welcome Home!</h2>'
end

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



21. Execute `rspec` to see the error:

```
$ chef exec rspec spec/unit/recipes/default_spec.rb
```

Expected output:

```

...F.
Failures:

  1) apache::default When all attributes are default, on CentOS 8 create
     Failure/Error: expect(chef_run).to render_file('/var/www/html/index.html')

       expected Chef run to render "/var/www/html/index.html" matching:

       <h1>Welcome Home!</h1>

       but got:

       <h1>Welcome Home!</h2>

       # ./spec/unit/recipes/default_spec.rb:34:in `block (3 levels) in <|
Finished in 1.62 seconds (files took 1.12 seconds to load)
5 examples, 1 failure

```

22. Fix the error created in recipes/default.rb

- a. Execute rspec again to ensure the errors were fixed properly

Expected output with no errors:

```

.....
Finished in 1.62 seconds (files took 1.12 seconds to load)
5 examples, 0 failures

```

Notify your instructor that you are done with the lab

**END OF LAB**