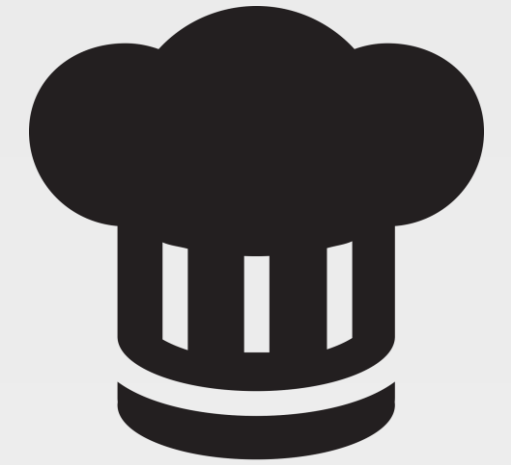# Creating Ohai Plugins

Objectives

After completing this module, you should be able to:

➢ Create a tested Ohai plugin

EXERCISE

Creating an Ohai Plugin

Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.

Objective:

❑ Define expectations for the Ohai plugin

❑ Create the Ohai plugin

❑ Define expectations for the recipe to deliver the Ohai plugin

❑ Create the recipe that delivers the Ohai plugin

❑ Define an integration test

Installing the chefspec-ohai gem

```
> chef gem install chefspec-ohai
```

```
Successfully installed chefspec-ohai-0.1.1
1 gem installed
```

Adding the Gem to the Spec Helper

~/apache/spec/spec_helper.rb

```
require 'chefspec'
require 'chefspec/berkshelf'
require 'chefspec/ohai'
```

CHEF

Creating a Directory for Plugins Tests

```
> mkdir ~/apache/spec/unit/plugins
```

Defining the First Expectation for Plugin

~/apache/spec/unit/plugins/apache_modules_spec.rb

```
require 'spec_helper'

describe_ohai_plugin :Apache do
  let(:plugin_file) { 'files/default/apache_modules.rb' }

  it 'provides apache/modules' do
    expect(plugin).to provides_attribute('apache/modules')
  end
end
```

Executing the Tests to See Failure

```
> chef exec rspec
```

```
F..........

Failures:


 1) Apache provides 'apache/modules'
     Failure/Error: let(:plugin_source) { File.read(plugin_file) }


     Errno::ENOENT:
        No such file or directory @ rb_sysopen -
files/default/apache_modules.rb
```
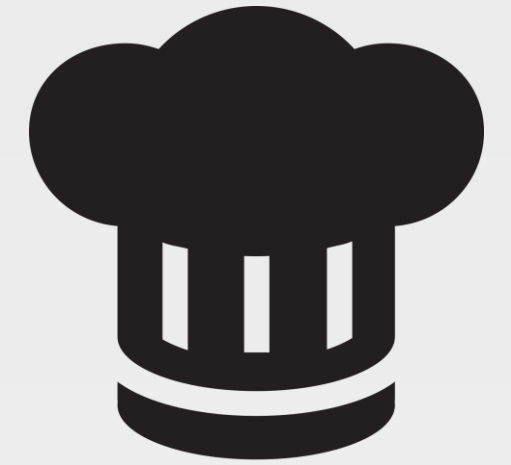
Creating an Ohai Plugin

Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.

Objective:

- ✓ Define expectations for the Ohai plugin
- ❑ Create the Ohai plugin
- ❑ Define expectations for the recipe to deliver the Ohai plugin
- ❑ Create the recipe that delivers the Ohai plugin
- ❑ Define an integration test

CHEF

Creating the Plugin as a Cookbook File

```
> chef generate file apache_modules.rb
```

```
Recipe: code_generator::cookbook_file
  * directory[/home/chef/apache/files/default] action create
    - create new directory /home/chef/apache/files/default
    - restore selinux security context
  * template[/home/chef/apache/files/default/apache_modules] action create
    - create new file /home/chef/apache/files/default/apache_modules
    - update content in file /home/chef/apache/files/default/apache_modules
from none to e3b0c4
```

Defining the Plugin that Provides Values

`~/apache/files/default/apache_modules.rb`

```
Ohai.plugin(:Apache) do
  provides 'apache/modules'
end
```

Executing the Tests to See Success

```
> chef exec rspec
```

```
...........

Finished in 2.15 seconds (files took 1.88 seconds to load)
11 examples, 0 failures
```

Defining an Expectation for Attribute Value

~/apache/spec/unit/plugins/apache_modules_spec.rb

```ruby
require 'spec_helper'

describe_ohai_plugin :Apache do
  let(:plugin_file) { 'files/default/apache_modules.rb' }

  it 'provides apache/modules' do
    expect(plugin).to provides_attribute('apache/modules')
  end


  it 'correctly captures output' do
    allow(plugin).to receive(:shell_out).with('apachectl -t -D
DUMP_MODULES').and_return(double(stdout: 'OUTPUT'))
    expect(plugin_attribute('apache/modules')).to eq('OUTPUT')
  end
end
```

CHEF

Executing the Tests to See Failure

```
> chef exec rspec
```

```
.F.........

Failures:


  1) Apache correctly captures output
     Failure/Error: expect(plugin_attribute('apache/modules')).to
eq("OUTPUT")

     NoMethodError:
        undefined method `[]' for nil:NilClass
```

Capturing the Apache Modules Content

~/apache/files/default/apache_modules.rb

```
Ohai.plugin :Apache do
  provides 'apache/modules'

  collect_data :default do
    apache(Mash.new)
    modules_cmd = shell_out('apachectl -t -D DUMP_MODULES')
    apache[:modules] = modules_cmd.stdout
  end
end
```

CHEF

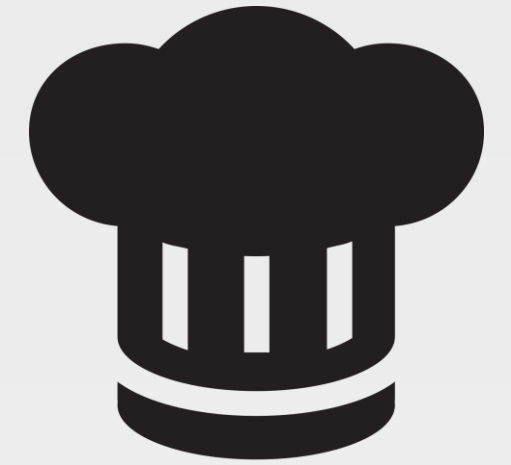# Executing the Tests to See Success

```
> chef exec rspec
```

```
............


Finished in 2.08 seconds (files took 1.88 seconds to load)
12 examples, 0 failures
```

EXERCISE

Creating an Ohai Plugin

Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.

Objective:

- ✓ Define expectations for the Ohai plugin
- ✓ Create the Ohai plugin
- ❑ Define expectations for the recipe to deliver the Ohai plugin
- ❑ Create the recipe that delivers the Ohai plugin
- ❑ Define an integration test

Adding a Dependency on the Ohai Cookbook

~/apache/metadata.rb

```
name 'apache'
maintainer 'The Authors'
maintainer_email 'you@example.com'
license 'All Rights Reserved'
description 'Installs/Configures apache'
long_description 'Installs/Configures apache'
version '0.1.0'
chef_version '>= 12.1' if respond_to?(:chef_version)

depends 'ohai'
# The `issues_url` points to the location where issues for this cookbook are
# tracked.  A `View Issues` link will be displayed on this cookbook's page when
# uploaded to a Supermarket.
#
# issues_url 'https://github.com/<insert_org_here>/apache/issues'
```

CHEF

Creating the Recipe to Add the Plugin

```
> chef generate recipe ohai_apache_modules
```

```
Recipe: code_generator::recipe
  * directory[/home/chef/apache/spec/unit/recipes] action create
(up to date)
  * cookbook_file[/home/chef/apache/spec/spec_helper.rb] action
create_if_missing (up to date)
  *
template[/home/chef/apache/spec/unit/recipes/ohai_apache_modules_s
pec.rb] action create_if_missing
...
```

Delete the Context for Ubuntu

`~/apache/spec/unit/recipes/ohai_apache_modules_spec.rb`

```ruby
require 'spec_helper'

describe 'apache::ohai_apache_modules' do
  context 'When all attributes are default, on Ubuntu 16.04' do
    let(:chef_run) do
      # for a complete list of available platforms and versions see:
      # https://github.com/customink/fauxhai/blob/master/PLATFORMS.md
      runner = ChefSpec::ServerRunner.new(platform: 'ubuntu', version: '16.04')
      runner.converge(described_recipe)
    end

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

Defining the Expectation Plugin Deployed

~/apache/spec/unit/recipes/ohai_apache_modules_spec.rb

```ruby
describe 'apache::ohai_apache_modules' do
  context 'When all attributes are default, on CentOS 6.9' do
    let(:chef_run) do
      # for a complete list of available platforms and versions see:
      # https://github.com/customink/fauxhai/blob/master/PLATFORMS.md
      runner = ChefSpec::ServerRunner.new(platform: 'centos', version: '6.9')
      runner.converge(described_recipe)
    end

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

    it 'installs the modules plugin' do
      expect(chef_run).to create_ohai_plugin('apache_modules')
    end
  end
end
```

CHEF

Execute the Tests to See Failure

```
> chef exec rspec
```

```
...........F

Failures:

  1) apache::ohai_apache_modules When all attributes are default,
on an unspecified platform installs the modules plugin
     Failure/Error: expect(chef_run).to
create_ohai_plugin('apache_modules')


     expected "ohai_plugin[apache_modules]" with action :create
to be in Chef run. Other ohai_plugin resources:
```
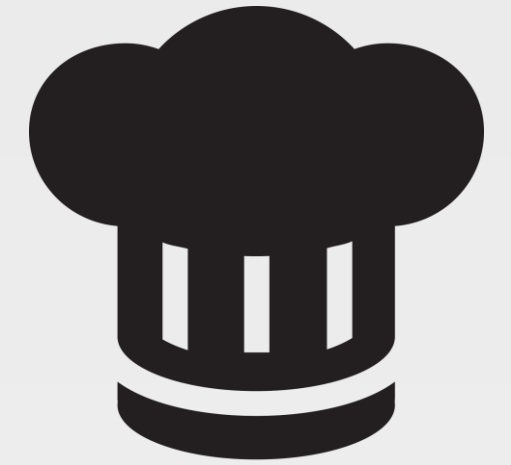
CHEF

EXERCISE

Creating an Ohai Plugin

Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.

Objective:

✓ Define expectations for the Ohai plugin

✓ Create the Ohai plugin

✓ Define expectations for the recipe to deliver the Ohai plugin

❑ Create the recipe that delivers the Ohai plugin

❑ Define an integration test

Defining the Ohai Plugin in the Recipe

~/apache/recipes/ohai_apache_modules.rb

```
#
# Cookbook Name:: apache
# Recipe:: ohai_apache_modules
#
# Copyright (c) 2017 The Authors, All Rights Reserved.
ohai_plugin 'apache_modules'
```
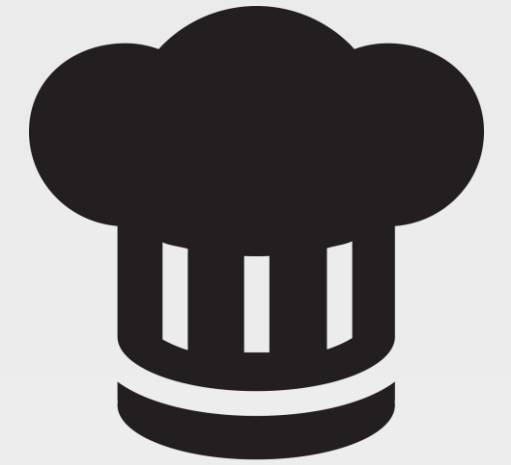
Executing the Tests to See Success

```
> chef exec rspec
```

```
..............

Finished in 3.33 seconds (files took 1.87 seconds to load)
14 examples, 0 failures
```

EXERCISE

Creating an Ohai Plugin

Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.

Objective:

✓ Define expectations for the Ohai plugin

✓ Create the Ohai plugin

✓ Define expectations for the recipe to deliver the Ohai plugin

✓ Create the recipe that delivers the Ohai plugin

❑ Define an integration test

Appending the New Recipe to the Suite

```
# ... TOP OF KITCHEN CONFIGURATION ...
suites:
  - name: default
    run_list:
        - recipe[apache::default]
        - recipe[apache::ohai_apache_modules]
    verifier:
      inspec_tests:
        - test/smoke/default
    attributes:
```

Converging the Updated Default Suite

```
> kitchen converge
```

```
-----> Starting Kitchen (v1.19.2)
-----> Converging <default-centos-69>...
       Preparing files for transfer
       Preparing dna.json
       Resolving cookbook dependencies with Berkshelf 6.3.1...
       Removing non-cookbook files before transfer
       Preparing validation.pem
       Preparing client.rb
-----> Chef Omnibus installation detected (install only if missing)
       Transferring files to <default-centos-69>
       Starting Chef Client, version 13.8.5
       resolving cookbooks for run list: ["apache::default", "apache::ohai_apache_modu
les"]
       Synchronizing Cookbooks:
```

Defining an Expectation for the Plugin

~/apache/test/smoke/default/ohai_apache_modules.rb

```
plugin_directory = '/tmp/kitchen/ohai/plugins'

describe command("ohai -d #{plugin_directory} apache") do
  its(:stdout) { should match(/core_module/) }
end
```

Verifying the New Expectation

```
> kitchen verify
```

```
Target:  ssh://kitchen@localhost:32769


  Command curl
     ✔   http://localhost stdout should match /Welcome users/
  Command curl
     ✔   http://localhost:8080 stdout should match /Welcome admins/
  Command ohai
     ✔   -d /tmp/kitchen/ohai/plugins apache stdout should match /core_module/


Test Summary: 3 successful, 0 failures, 0 skipped
```
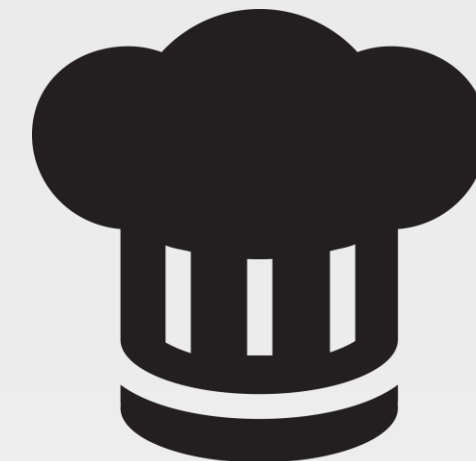
CHEF

EXERCISE

Creating an Ohai Plugin

Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.

Objective:

✓ Define expectations for the Ohai plugin

✓ Create the Ohai plugin

✓ Define expectations for the recipe to deliver the Ohai plugin

✓ Create the recipe that delivers the Ohai plugin

✓ Define an integration test

DISCUSSION

Q&A

What questions can we answer for you?

CHEF