

Hooks and Tags

Advanced Cucumber

Organization

- Hooks
- Tags
- Formatting Output
- Continuous Integration

Hooks and the Spec Life Cycle

- **Before and After**
 - Run before and after each scenario
- **Around**
 - Allows you to wrap a scenario in a block of code
- **AfterStep**
 - Runs after each step
- **AfterConfiguration**
 - Runs after Cucumber is configured before any scenarios
- **at_exit**
 - From Kernel
 - Called when Cucumber exits

Hooks and the Spec Life Cycle

```
AfterConfiguration do |config|  
  puts "After Configuration"  
end
```

```
at_exit do  
  puts "Exiting"  
end
```

```
Before do |scenario|  
  puts "Before"  
end
```

```
After do |scenario|  
  puts "After"  
end
```

```
AfterStep do |step|  
  puts "After Step"  
end
```

```
Around do |scenario, block|  
  puts "Around Starting"  
  block.call  
  puts "Around Ending"  
end
```

After Configuration
Around Starting
Before
Step Executing
After Step
Step Executing
After Step
After
Around Ending
Exiting

Hooks and the Spec Life Cycle

- Can have multiple Before and After hooks
- Exceptions in hooks are not visible, can cause unexpected results when running tests

Tags

- Allow categorization of a scenario or feature
- Scenario or feature can have many tags
- Scenarios inherit tags from the Feature

Tags

@authorization

Feature: Accessing Account Information

Scenario: First Scenario

@smoke_test @wip

Scenario: Second Scenario

@wip

Scenario: Third Scenario

Tags

- Can use cucumber executable to include or exclude tags

```
cucumber features/x.feature --tag @smoke_test --tag ~@wip
```


Tags

Tags can be used to add scope to a hook

Feature: New User Setup

@new-user

Scenario: First time user gets orientation

Given I access the system as a new user

Then I should be presented with the orientation

```
Before ( '@new-user' ) do |scenario|  
  #code to create a new user  
end
```

Formatting Output

- Can specify different output formats
 - Simple: HTML, Pretty, Progress
 - Advanced: JUnit, debug
- Can use the rerun formatter to support rerunning failed tests
- Can specify multiple formatters at a time

Continuous Integration

- Important to have test running continuously
- Use different suites to run at different intervals or perform parallel runs
- Cucumber executable has a 0 exit status if all tests pass
- Some CI systems integrate with Rake
- Different formatters: HTML, JUnit, Progress