

Basic Function Testing with Pester



Robert C. Cain, MVP

OWNER, ARCANE TRAINING AND CONSULTING

@arcanecode www.arcanecode.com





Introduction to Pester

Implementing Your First Pester Tests



Requirements – Get-PodcastData



Parameter with URL of RSS feed

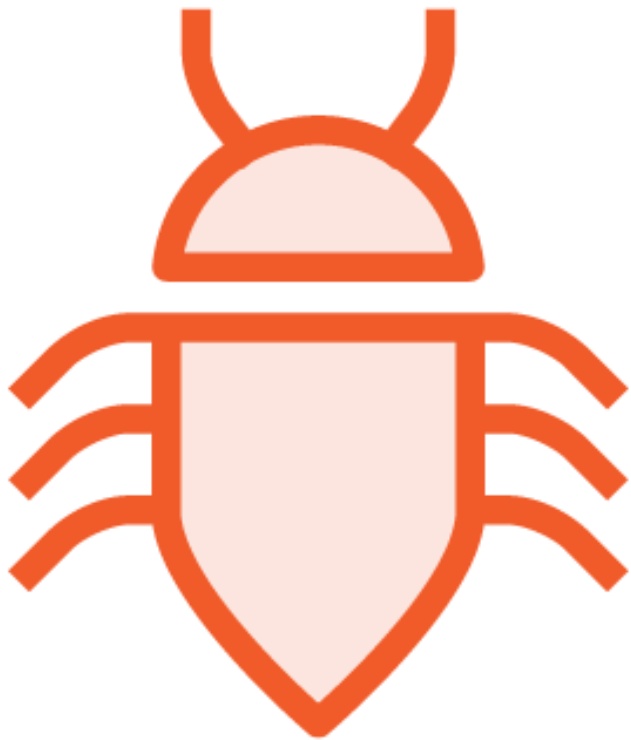
Return a list of objects with properties of:

- Title, ShowURL, EmbeddedHTML,
- Hosts, PublicationDate, ImageURL,
- AudioURL, AudioLength

Same number of rows as in podcast feed

Validate values in the data

Tests – Get-PodcastData



Validate the set of custom objects has the correct properties

Validate there are the correct number of rows as in the RSS feed (15)

Validate the data returned is accurate

Requirements – Get-PodcastImage



Accept data from Get-PodcastData as parameter named RSSData

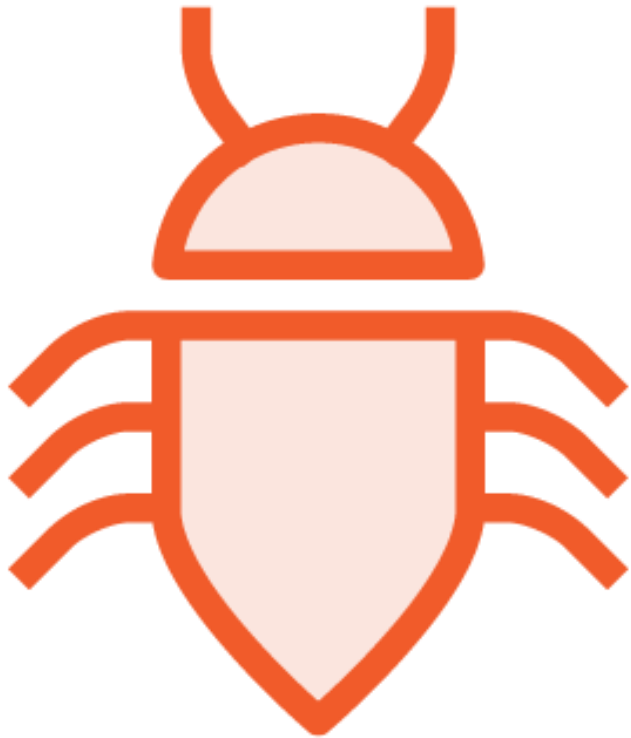
Use the ImageURL with Invoke-WebRequest to download the image

Store image in folder specified in parameter named OutputPathFolder, which will have a default value

Use the filename from the ImageURL as the file name in OutputPathFolder

Returns a list of the file names downloaded

Unit Tests – Get-PodcastImage

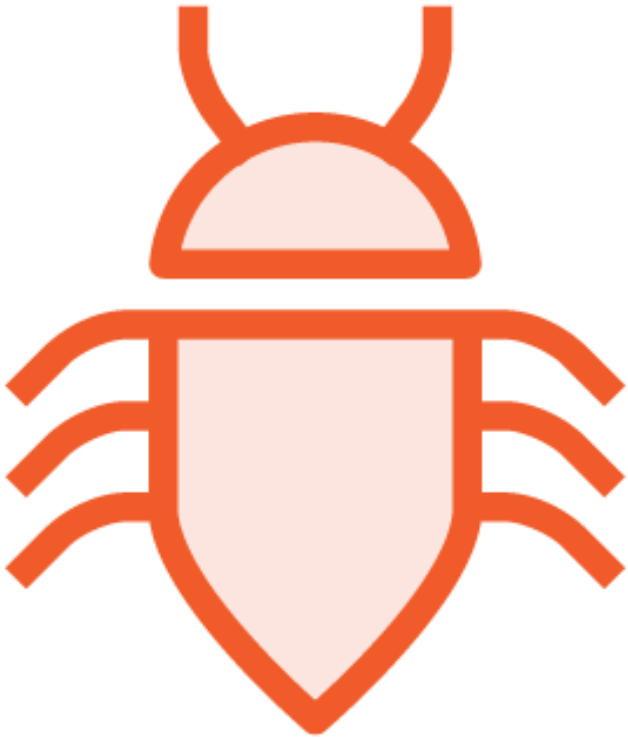


Use mocked, known data.

Validate each file from the RSS feed exists in the unit testing location indicated in the OutputPathFolder parameter

Validate each file name exists in the array of file names returned from the function

Acceptance Tests – Get-PodcastImage



Use live data from Get-PodcastData.

Validate each file from the RSS feed exists in both the default value and a passed in value for the OutputPathFolder parameter

Validate each file name exists in the array of file names returned from the function

Summary



Context

Dynamic Tests

Mock Data

InModuleScope

