avocado Documentation

Release 0.0.1

Lucas Meneghel Rodrigues

Jul 14, 2020

Contents

Contents:

CHAPTER 1

About Avocado

Avocado is a set of tools and libraries (what people call these days a framework) to perform automated testing.

Avocado is composed by:

- Programs that let you run tests. Those tests can be either written on your language of choice, or use the python API available. In both cases, you get facilities such as automated log and system information collection.
- APIs that help you write tests in a concise, yet expressive way.

CHAPTER 2

Getting started guide - users

If you want to simply use avocado as a test runner/test API, you can install a distro package. For Ubuntu, you can look at lmr's autotest PPA, while for Fedora, you can look at lmr's autotest COPR.

CHAPTER 3

Installing avocado - Ubuntu

You can install the debian package by performing the following commands:

```
sudo add-apt-repository ppa:lmr/autotest
sudo apt-get update
sudo apt-get install avocado
```

CHAPTER 4

Installing avocado - Fedora

You can install the rpm package by performing the following commands:

```
sudo curl http://copr.fedoraproject.org/coprs/lmr/Autotest/repo/fedora-20-i386/ > /
↪etc/yum.repos.d/autotest.repo
sudo yum update
sudo yum install avocado
```

CHAPTER 5

Writing Avocado Tests

Avocado tests closely resemble autotest tests: All you need to do is to create a test module, which is a python file with a class that inherits from *avocado.test.Test*. This class only really needs to implement a method called *action*, which represents the actual test payload. Let's re-create an old time functional test for autotest, a simple *time.sleep([number-seconds])*:

```
#!/usr/bin/python

import time

from avocado import job
from avocado import test


class sleeptest(test.Test):

    """
    Example test for avocado.
    """

    def action(self, length=1):
        """
        Sleep for length seconds.
        """
        self.log.debug("Sleeping for %d seconds", length)
        time.sleep(length)


if __name__ == "__main__":
    job.main()
```

This is about the simplest test you can write for avocado (at least, one using the avocado APIs). Note that the test object provides you with a number of convenience attributes, such as *self.log*, that lets you log debug, info, error and warning messages.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search