Toast Documentation

Release 0.1-dev

Chris Trotman

April 10, 2016

Contents

Toast is an extension for salt. It contains a external_tops extension and an external_pillar extension. The main purpose of toast is to provide a completely customisable way of including states depending on the grains of minions. Toast is more powerful than the normal pillar state files, as it exposes a full python interface to you. This allows you to create a hierachy of minions, simply by implementing classes and subclasses. The sky is the limit, as long as you still return a list of states and a pillar dictionary!

Contents:

CHAPTER 1

Installation

CHAPTER 2

API

Normal usage of toast should not require you to know how toast works internaly. But here is the source documentation anyway.

class toast.toast.**Toaster**(*path*, *grains*)

load()

pillars()

states()

2.1 Salt entry points

This module contains the entry points for the salt adapters.

toast.salt_entry.**pillar**(*__opts__=None*, *minion_id=None*, *pillar=None*, *grains=None*, ***kwargs*)
The main entry point for the ext_pillar extension

toast.salt_entry.**top**(*__opts__=None*, *grains=None*, ***kwargs*)
The main entry point for the external_tops extension

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

t

L

P

S

T