SBT Made Simple

@囚千任 - http://afoo.me

Immutability

Setting

- A Build File is just a sequence of Settings
- A Setting is just a transformation to key-value pairs
- A Transformation to some key-value pair is immutable

State Transition

- Project State is volatile
- state transition is immutable

Composition

Task

- dependency model to composition model
- task composition to form bigger granularity unit

InputTask

- Task with input
- input validation and completion support with parser
- parser combinator is a form of composition

Command

- Special InputTask
  - accept Project state and settings as input
  - project state is driven by a composition of command execution

Modularity


THE ONLY CRIMINAL
HE CAN'T CATCH IS HIMSELF.
NICOLAS CAGE
EVA MENDES
BAD LIEUTENANT
PORT OF CALL NEW ORLEANS
A FILM BY WERNER HERZOG


THE FROZEN GROUND
BASED ON A TRUE STORY
The Hunt for Alaska's
Most Prolific Serial Killer


NICOLAS CAGE
IN THEATERS JAN 7
SEASON OF THE WITCH

Configuration

- Ivy concept similar to Maven's Scope
- Predefined or custom Configurations setup Modularity boundary

Project

- project is another level of modularity
- multiple project build definitions have their specific configuration

Plugins

- SBT plugin mechanism is another modularity strategy
- enhance reusability too.

Consistency

One Rule To Rule Them All

Same Effect?

```
name := {
    “hello” + “sbt”
}

sbt := {
    “hello” + “sbt”
}
```

Not Really!

```
name := {
    “hello” + “sbt”          =>      Setting[String]
}

sbt := {
    “hello” + “sbt”          => Setting[Task[String]]
}
```

Key is the key

```
val name = settingKey[String](“desc.”)
name := {
    “hello” + “sbt”    => Setting[String]
}

val sbt = taskKey[String](“task desc”)
sbt := {
    “hello” + “sbt”    => Setting[Task[String]]
}
```

Key is the key

- SettingKey[T]
- TaskKey[T]
- InputKey[T]

Demo Time~

thank you!