StateMachine

Release 0.0.1

Rafael S. Müller

Aug 31, 2021

CONTENTS:

1 Getting Started **1**
- 1.1 Context . . . 1
- 1.2 State . . . 1

2 study_state_machine **3**
- 2.1 study_state_machine package . . . 3

3 Indices and tables **5**

Python Module Index **7**

Index **9**

CHAPTER

ONE

GETTING STARTED

The State Machine is used to implement a workflow. Through a transition you change from one state to another. The current state determines the behavior.

1.1 Context

The `Context` serves as an interface between the state machine and the client . It delegates a request from the client to the current state. Depending on the current state, the request results in a different behavior.

The following example shows how to create a context and load a state by its name:

```
from state_machine.context import Context

context = Context()
context.load_state("RegisteredState")
context.change_state()
```

1.2 State

Each state is modeled as a JSON document with two set of strategies (create_study and change_state) that dictates a set of behaviors.

The existing states need to be passed to the context. Here is an example:

```
{
    "name" : "RegisteredState",
    "strategies_create_study" : [
        {
            "name" : "expression",
            "value" : "len(kwargs.get('datasets', [])) > 0",
            "state_if_true" : "DatasetState"
        }
    ],
    "strategies_change_state" : [
        {
            "name" : "expression",
            "value" : "len(kwargs.get('datasets', [])) > 0",
            "state_if_true" : "DatasetState"
        }
    ]
}
```

CHAPTER

TWO

STUDY_STATE_MACHINE

2.1 study_state_machine package

2.1.1 Subpackages

2.1.2 Submodules

2.1.3 study_state_machine.context module

Implementation of a finite state machine for studies

class study_state_machine.context.**Context**(*available_states*, *initial_state=None*)

Bases: object

Only if an initial state is passed to the constructor, the context of the current state is set. Otherwise, call *transition_to()* or *load_state()* with a name of a State, respectively

add_sample(**args*, ***kwargs*)

property available_states

change_state(**args*, ***kwargs*)

create_study(**args*, ***kwargs*)

property current_state

get_state_dict(*state_name*)

load_state(*state_name*)

Load state with given name

Parameters **state_name** – Name of the state

Raises ***StateNotFoundException*** – If the state is not found

parse_strategies(*strategies*, **args*, ***kwargs*)

Parse strategies list and return a state to be transitioned to

transition_to(*state_name*)

Set state as the new current state and set its context

Parameters **state_name** – new current state

2.1.4 study_state_machine.errors module

Collection of state machine related exceptions.

All exceptions inherit from *StateMachineException*

exception study_state_machine.errors.**BehaviorNotAllowedException**
Bases: *study_state_machine.errors.StateMachineException*

Exception if a state does not support a behavior

exception study_state_machine.errors.**StateMachineException**
Bases: Exception

Generic state machine error

exception study_state_machine.errors.**StateNotFoundException**
Bases: *study_state_machine.errors.StateMachineException*

Exception if state is not found by its name

2.1.5 study_state_machine.interfaces module

class study_state_machine.interfaces.**IState**(*context=None*)
Bases: abc.ABC

Base class for all states.

The state has a reference to the context in order to change into the next state.

property context

class study_state_machine.interfaces.**IStudyState**(*context=None*)
Bases: *study_state_machine.interfaces.IState*

Base class for all study states

add_sample(**args*, ***kwargs*)

change_state(**args*, ***kwargs*)

create_study(**args*, ***kwargs*)

2.1.6 Module contents

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

study_state_machine, 4
study_state_machine.context, 3
study_state_machine.errors, 4
study_state_machine.interfaces, 4

INDEX

A

add_sample() (*study_state_machine.context.Context method*), 3
add_sample() (*study_state_machine.interfaces.IStudyState method*), 4
available_states() (*study_state_machine.context.Context property*), 3

B

BehaviorNotAllowedException, 4

C

change_state() (*study_state_machine.context.Context method*), 3
change_state() (*study_state_machine.interfaces.IStudyState method*), 4
Context (*class in study_state_machine.context*), 3
context() (*study_state_machine.interfaces.IState property*), 4
create_study() (*study_state_machine.context.Context method*), 3
create_study() (*study_state_machine.interfaces.IStudyState method*), 4
current_state() (*study_state_machine.context.Context property*), 3

G

get_state_dict() (*study_state_machine.context.Context method*), 3

I

IState (*class in study_state_machine.interfaces*), 4
IStudyState (*class in study_state_machine.interfaces*), 4

L

load_state() (*study_state_machine.context.Context method*), 3

P

parse_strategies() (*study_state_machine.context.Context method*), 3

S

StateMachineException, 4
StateNotFoundException, 4
study_state_machine (*module*), 4
study_state_machine.context (*module*), 3
study_state_machine.errors (*module*), 4
study_state_machine.interfaces (*module*), 4

T

transition_to() (*study_state_machine.context.Context method*), 3