

---

# **yawf Documentation**

*Release*

**Author**

August 09, 2016



<b>1 yawf Package</b>	<b>3</b>
1.1 yawf Package	3
1.2 actions Module	3
1.3 allowed Module	3
1.4 annotation Module	3
1.5 base Module	3
1.6 base_model Module	3
1.7 config Module	4
1.8 creation Module	4
1.9 dispatch Module	4
1.10 effects Module	4
1.11 exceptions Module	5
1.12 forms Module	6
1.13 graph Module	6
1.14 graph_views Module	6
1.15 handlers Module	6
1.16 library Module	8
1.17 models Module	8
1.18 permissions Module	8
1.19 serialize_utils Module	8
1.20 signals Module	8
1.21 state_transition Module	8
1.22 utils Module	8
1.23 version Module	9
1.24 views Module	9
1.25 Subpackages	9
1.25.1 management Package	9
Subpackages	9
1.25.2 message_log Package	9
admin Module	9
models Module	9
1.25.3 messages Package	9
messages Package	9
allowed Module	9
common Module	10
spec Module	10
submessage Module	11
transformation Module	11

1.25.4	resources Package	11
	resources Package	11
	allowed Module	11
	base Module	11
1.25.5	revision Package	12
	revision Package	12
	models Module	12
	utils Module	12
	Subpackages	12
1.25.6	tests Package	12
	tests Package	12
	message_specs Module	12
	permissions Module	12
<b>2</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>

Contents:



---

## yawf Package

---

### 1.1 yawf Package

```
class yawf.__init__.WorkflowChoices
    Bases: object

yawf.__init__.autodiscover()
yawf.__init__.get_registered_workflows()
yawf.__init__.get_workflow(workflow_id)
yawf.__init__.get_workflow_by_instance(obj)
yawf.__init__.get_workflow_display_name(id)
yawf.__init__.get_workflow_name_map()
yawf.__init__.register_workflow(workflow)
```

### 1.2 actions Module

### 1.3 allowed Module

```
yawf.allowed.get_allowed(sender, obj)
```

### 1.4 annotation Module

### 1.5 base Module

### 1.6 base\_model Module

```
class yawf.base_model.WorkflowAwareModelBase
    Bases: object

    get_clarified_instance()
    state_display()
```

```
workflow
workflow_type_display()
```

## 1.7 config Module

## 1.8 creation Module

## 1.9 dispatch Module

## 1.10 effects Module

```
class yawf.effects.SideEffect (message_id=None, states_from=None, states_to=None, mes-
                               sage_group=None)
```

Bases: object

Basic class to express side-effect actions.

SideEffect in yawf is a side-effect action that can be performed after certain changes in workflow.

User can treat effects as event handlers and put that handlers on any transition. To specify a transition, one can specify a list of states before transition, a list of states after transition and a list of message ids. Everything above is optional, so if you register a default side-effect without specify any of this variables, such side-effect will be called after *each* transition.

User can put any number of side-effects on each and every transition. In case of multiple effects matching specific transition each side-effect will be executed one by one.

Yawf executes side-effect *after* state transition, when yawf obtains new state (and possibly modified object).

### Yawf passes keyword arguments to side-effect:

- *old\_obj*: the object *before* the state transition (before locking);
- *obj*: the object *after* the state transition;
- *sender*: message sender;
- *params*: message arguments;
- ***message\_spec*: message spec class (because we may want to distinguish between different messages in effects);**
- ***extra\_context*: extra context that was passed to `yawf.dispatch.dispatch()`;**
- *handler\_result*: result of the state transition routine.

```
is_transactional = False
```

```
message_group = None
```

```
message_id = None
```

```
name
```

```
perform (**kwargs)
```

```
set_performer (performer)
```

```
states_from = None
```



---

`states_to = None`

## 1.11 exceptions Module

**exception** `yawf.exceptions.ConcurrentRevisionUpdate`  
Bases: `yawf.exceptions.YawfException`  
`code = 'concurrent_revision_update'`

**exception** `yawf.exceptions.CreateValidationError` (*validator*)  
Bases: `yawf.exceptions.MessageValidationError`  
`code = 'create_validation_error'`

**exception** `yawf.exceptions.GroupPathEmptyError`  
Bases: `yawf.exceptions.YawfException`  
`code = 'group_path_empty_error'`

**exception** `yawf.exceptions.IllegalStateError`  
Bases: `yawf.exceptions.YawfException`  
`code = 'illegal_state_error'`

**exception** `yawf.exceptions.MessageIgnored`  
Bases: `yawf.exceptions.YawfException`  
`code = 'message_ignored'`  
`context`

**exception** `yawf.exceptions.MessageSpecNotRegisteredError`  
Bases: `yawf.exceptions.YawfException`  
`code = 'message_spec_not_registered_error'`

**exception** `yawf.exceptions.MessageValidationError` (*validator*)  
Bases: `yawf.exceptions.YawfException`  
`code = 'yawf_validation_error'`  
`context`

**exception** `yawf.exceptions.NoAvailableMessagesError`  
Bases: `yawf.exceptions.YawfException`  
`code = 'no_available_messages_error'`

**exception** `yawf.exceptions.OldStateInconsistenceError`  
Bases: `yawf.exceptions.YawfException`  
`code = 'old_state_inconsistence_error'`

**exception** `yawf.exceptions.PermissionDeniedError`  
Bases: `yawf.exceptions.YawfException`  
`code = 'permission_denied_error'`

**exception** `yawf.exceptions.ResourceNotFoundError`  
Bases: `yawf.exceptions.YawfException`  
`code = 'resource_not_found_error'`

**exception** `yawf.exceptions.ResourcePermissionDeniedError`

Bases: `yawf.exceptions.PermissionDeniedError`

`code = 'resource_permission_denied_error'`

**exception** `yawf.exceptions.UnhandledMessageError`

Bases: `yawf.exceptions.YawfException`

`code = 'unhandled_message_error'`

`context`

**exception** `yawf.exceptions.WorkflowAlreadyRegisteredError`

Bases: `yawf.exceptions.YawfException`

`code = 'workflow_already_registered_error'`

**exception** `yawf.exceptions.WorkflowNotLoadedError`

Bases: `yawf.exceptions.YawfException`

`code = 'workflow_not_loaded_error'`

**exception** `yawf.exceptions.WrongHandlerResultError`

Bases: `yawf.exceptions.YawfException`

`code = 'wrong_handler_result_error'`

**exception** `yawf.exceptions.YawfException`

Bases: `coded_exceptions.CodedException`

`code = 'yawf_exception'`

## 1.12 forms Module

`yawf.forms.get_action_form_html(obj, sender)`

`yawf.forms.get_create_form_html(workflow_type, sender=None)`

`yawf.forms.get_object_as_html(obj, sender)`

## 1.13 graph Module

## 1.14 graph\_views Module

## 1.15 handlers Module

**class** `yawf.handlers.Handler` (*message\_id=None, states\_from=None, message\_group=None, permission\_checker=None*)

Bases: `object`

Basic class to express state transitions, both simple and extended.

Handler object is a callable. It gets a workflow object, message sender and arbitrary keyword arguments (passed with message) and must return new state id based on this information (it can return None and message will be ignored).

If handler returns a callable, then this callable applies as transaction on object (to change its extended state). The result of the latter callable represents a transition result. If callable returns generator, transition routine will iterate over it, collecting the yielded values. For more information about state transition, see `yawf.state_transition` module.

Transitioning callable can actually return anything besides generator. That is totally an application designer business.

**copy\_before\_call** = False

**defer** = True

**message\_group** = None

**message\_id** = None

**perform** (*obj*, *sender*, **\*\*kwargs**)

**permission\_checker** = None

**replace\_if\_exists** = False

**set\_performer** (*handle\_func*)

**states\_from** = None

**class** `yawf.handlers.SimpleStateTransition` (*\*args*, **\*\*kwargs**)

Bases: `yawf.handlers.Handler`

**perform** (*obj*, *sender*, **\*\*kwargs**)

**state\_to** = None

**class** `yawf.handlers.ComplexStateTransition` (*message\_id=None*, *states\_from=None*, *message\_group=None*, *permission\_checker=None*)

Bases: `yawf.handlers.Handler`

**perform** (*obj*, *sender*, **\*\*kwargs**)

**transition** (*obj*, *sender*, **\*\*kwargs**)

**class** `yawf.handlers.EditHandler` (*message\_id=None*, *states\_from=None*, *message\_group=None*, *permission\_checker=None*)

Bases: `yawf.handlers.ComplexStateTransition`

**field\_names** = None

**post\_hook** (*obj*)

**transition** (*obj*, *sender*, **\*\*kwargs**)

**class** `yawf.handlers.SerializableHandlerResult` (**\*\*kwargs**)

Bases: `object`

**get\_serializable\_value** ()

**type** = None

## 1.16 library Module

## 1.17 models Module

## 1.18 permissions Module

```
class yawf.permissions.AndChecker (*checkers)
    Bases: yawf.permissions.BasePermissionChecker

class yawf.permissions.BasePermissionChecker (*checkers)
    Bases: object

    add_checker (checker)
    fill_cache (obj, sender)
    get_atomical_checkers ()
    perform_child_checker (checker, obj, sender, cache)

yawf.permissions.C
    alias of AndChecker

class yawf.permissions.NotChecker (checker)
    Bases: yawf.permissions.BasePermissionChecker

class yawf.permissions.OrChecker (*checkers)
    Bases: yawf.permissions.BasePermissionChecker
```

## 1.19 serialize\_utils Module

## 1.20 signals Module

## 1.21 state\_transition Module

## 1.22 utils Module

```
yawf.utils.chained_apply (callables_iterable)
yawf.utils.common_cancel (obj, cancel_state='canceled', soft_delete_attr=None)
yawf.utils.common_start (obj, state, soft_delete_attr=None)
yawf.utils.filter_handler_result (result, cls_to_filter)
yawf.utils.filter_side_effect_results (results, cls_to_filter)
yawf.utils.make_common_cancel (cancel_state='canceled', soft_delete_attr=None)
yawf.utils.make_common_start (state, soft_delete_attr=None)
yawf.utils.make_common_updater (kwargs, field_names=None, post_hook=None)
yawf.utils.maybe_list (a)
yawf.utils.memoizable_property (getter)
```

`yawf.utils.metadefaultdict` (*fabric*)  
`yawf.utils.model_diff` (*instance1, instance2, full=False*)  
`yawf.utils.model_diff_fields` (*instance1, instance2*)  
`yawf.utils.optionally_edit` (*handler*)  
`yawf.utils.select_for_update` (*queryset*)

## 1.23 version Module

## 1.24 views Module

## 1.25 Subpackages

### 1.25.1 management Package

#### Subpackages

commands Package

`graph_yawf_effects` Module

`graph_yawf_handlers` Module

### 1.25.2 message\_log Package

admin Module

models Module

### 1.25.3 messages Package

messages Package

allowed Module

`class yawf.messages.allowed.AllowedWrapper` (*sender, obj, specs=None*)

Bases: object

`yawf.messages.allowed.annotate_with_allowed_messages` (*sender, objects*)

`yawf.messages.allowed.get_allowed_messages` (*sender, obj, cache=None*)

`yawf.messages.allowed.get_allowed_messages_for_many` (*sender, objects*)

`yawf.messages.allowed.get_message_specs` (*sender, obj*)

`yawf.messages.allowed.get_message_specs_for_many` (*sender, objects*)

`yawf.messages.allowed.is_valid_message` (*obj, message\_id*)

### common Module

```
class yawf.messages.common.BasicCancelMessage (**attrs)
    Bases: yawf.messages.spec.MessageSpec
```

```
    id = 'cancel'
```

```
    rank = 1000
```

```
    verb = <django.utils.functional.__proxy__ object>
```

```
class yawf.messages.common.BasicDeleteMessage (**attrs)
    Bases: yawf.messages.spec.MessageSpec
```

```
    id = 'delete'
```

```
    rank = 1001
```

```
    verb = <django.utils.functional.__proxy__ object>
```

```
class yawf.messages.common.BasicEditMessage (**attrs)
    Bases: yawf.messages.spec.MessageSpec
```

```
    id = 'edit'
```

```
    rank = 900
```

```
    verb = <django.utils.functional.__proxy__ object>
```

```
class yawf.messages.common.BasicStartMessage (**attrs)
    Bases: yawf.messages.spec.MessageSpec
```

```
    id = 'start_workflow'
```

```
    verb = <django.utils.functional.__proxy__ object>
```

```
yawf.messages.common.message_spec_fabric(id, verb=None, rank=0, base_spec=<class
    'yawf.messages.spec.MessageSpec'>, **attrs)
```

### spec Module

```
class yawf.messages.spec.EmptyValidator (*args, **kwargs)
    Bases: object
```

Class with django-form behaviour that filters out all incoming data, always becomes valid and returns empty dict as cleaned data.

```
    cleaned_data
```

```
    is_valid()
```

```
class yawf.messages.spec.MessageSpec (**attrs)
    Bases: object
```

Specifies a message that can be passed to workflow.

Basic class for one of main entities. Message spec knows:

- it's unique “id”;
- how to represent spec for humans (using “verb” in `__unicode__`);
- how to validate parameters for this message (using “validator\_cls”);
- priority (“rank” attribute).

**dehydrate\_params** (*obj, message*)

Method, that returns dehydrated message params for serialization.

**id** = None

**id\_grouper** = '\_'

**is\_grouped** = False

**params\_wrapper** (*params*)

Wrapper used to wrap cleaned\_data returned by validator\_cls before passing it to handler.

With wrapper one has additional layer to transform structure of cleaned\_data dict to make it more convenient to use.

For example, see yawf.messages.common.BasicEditMessage class.

**rank** = 0

**validator\_cls**

alias of *EmptyValidator*

**verb** = None

### submessage Module

**class** yawf.messages.submessage.**RecursiveSubmessage** (*message\_id, sender, params=None, raw\_params=None*)

Bases: *yawf.messages.submessage.Submessage*

**dispatch** (*parent\_obj, parent\_message*)

**class** yawf.messages.submessage.**Submessage** (*obj, message\_id, sender, params=None, need\_lock\_object=True, raw\_params=None*)

Bases: object

**as\_message** (*parent*)

**dispatch** (*parent\_obj, parent\_message*)

**need\_lock\_object** = True

### transformation Module

#### 1.25.4 resources Package

##### resources Package

##### allowed Module

yawf.resources.allowed.**get\_allowed\_resources** (*sender, obj*)

yawf.resources.allowed.**get\_resource** (*obj, resource\_id*)

##### base Module

**class** yawf.resources.base.**WorkflowResource** (*handler, resource\_id, permission\_checker, description=None, slug=None*)

Bases: object

## 1.25.5 revision Package

revision Package

models Module

utils Module

Subpackages

migrations Package

0001\_initial Module

0002\_add\_is\_created\_field Module

## 1.25.6 tests Package

tests Package

message\_specs Module

```
class yawf.tests.message_specs.MessageSpecsTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
        test_fabric()
        test_grouping()
        test_repr()
        test_validator()
        test_without_id()
```

permissions Module

```
class yawf.tests.permissions.PermissionsTestCase (methodName='runTest')
    Bases: django.test.testcases.TestCase
        static call_count (func)
        setUp()
        test_atomics()
        test_cache()
            Call complex checker three times and check times that checker functions actually been called
        test_checker_expressions()
        test_custom_sender()
        test_cyclic()
        test_fill_cache()
        test_inversion()
```



`test_inversion_invariance()`



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



## y

- yawf.\_\_init\_\_, 3
- yawf.allowed, 3
- yawf.base\_model, 3
- yawf.config, 4
- yawf.effects, 4
- yawf.exceptions, 5
- yawf.forms, 6
- yawf.handlers, 6
- yawf.messages, 9
  - yawf.messages.allowed, 9
  - yawf.messages.common, 10
  - yawf.messages.spec, 10
  - yawf.messages.submessage, 11
- yawf.permissions, 8
- yawf.resources, 11
  - yawf.resources.allowed, 11
  - yawf.resources.base, 11
- yawf.signals, 8
- yawf.tests, 12
  - yawf.tests.message\_specs, 12
  - yawf.tests.permissions, 12
- yawf.utils, 8
- yawf.version, 9



**A**

add\_checker() (yawf.permissions.BasePermissionChecker method), 8  
 AllowedWrapper (class in yawf.messages.allowed), 9  
 AndChecker (class in yawf.permissions), 8  
 annotate\_with\_allowed\_messages() (in module yawf.messages.allowed), 9  
 as\_message() (yawf.messages.submessage.Submessage method), 11  
 autodiscover() (in module yawf.\_\_init\_\_), 3

**B**

BasePermissionChecker (class in yawf.permissions), 8  
 BasicCancelMessage (class in yawf.messages.common), 10  
 BasicDeleteMessage (class in yawf.messages.common), 10  
 BasicEditMessage (class in yawf.messages.common), 10  
 BasicStartMessage (class in yawf.messages.common), 10

**C**

C (in module yawf.permissions), 8  
 call\_count() (yawf.tests.permissions.PermissionsTestCase static method), 12  
 chained\_apply() (in module yawf.utils), 8  
 cleaned\_data (yawf.messages.spec.EmptyValidator attribute), 10  
 code (yawf.exceptions.ConcurrentRevisionUpdate attribute), 5  
 code (yawf.exceptions.CreateValidationError attribute), 5  
 code (yawf.exceptions.GroupPathEmptyError attribute), 5  
 code (yawf.exceptions.IllegalStateError attribute), 5  
 code (yawf.exceptions.MessageIgnored attribute), 5  
 code (yawf.exceptions.MessageSpecNotRegisteredError attribute), 5  
 code (yawf.exceptions.MessageValidationError attribute), 5  
 code (yawf.exceptions.NoAvailableMessagesError attribute), 5

code (yawf.exceptions.OldStateInconsistenceError attribute), 5  
 code (yawf.exceptions.PermissionDeniedError attribute), 5  
 code (yawf.exceptions.ResourceNotFoundError attribute), 5  
 code (yawf.exceptions.ResourcePermissionDeniedError attribute), 6  
 code (yawf.exceptions.UnhandledMessageError attribute), 6  
 code (yawf.exceptions.WorkflowAlreadyRegisteredError attribute), 6  
 code (yawf.exceptions.WorkflowNotLoadedError attribute), 6  
 code (yawf.exceptions.WrongHandlerResultError attribute), 6  
 code (yawf.exceptions.YawfException attribute), 6  
 common\_cancel() (in module yawf.utils), 8  
 common\_start() (in module yawf.utils), 8  
 ComplexStateTransition (class in yawf.handlers), 7  
 ConcurrentRevisionUpdate, 5  
 context (yawf.exceptions.MessageIgnored attribute), 5  
 context (yawf.exceptions.MessageValidationError attribute), 5  
 context (yawf.exceptions.UnhandledMessageError attribute), 6  
 copy\_before\_call (yawf.handlers.Handler attribute), 7  
 CreateValidationError, 5

**D**

defer (yawf.handlers.Handler attribute), 7  
 dehydrate\_params() (yawf.messages.spec.MessageSpec method), 10  
 dispatch() (yawf.messages.submessage.RecursiveSubmessage method), 11  
 dispatch() (yawf.messages.submessage.Submessage method), 11

**E**

EditHandler (class in yawf.handlers), 7  
 EmptyValidator (class in yawf.messages.spec), 10

## F

field\_names (yawf.handlers.EditHandler attribute), 7  
 fill\_cache() (yawf.permissions.BasePermissionChecker method), 8  
 filter\_handler\_result() (in module yawf.utils), 8  
 filter\_side\_effect\_results() (in module yawf.utils), 8

## G

get\_action\_form\_html() (in module yawf.forms), 6  
 get\_allowed() (in module yawf.allowed), 3  
 get\_allowed\_messages() (in module yawf.messages.allowed), 9  
 get\_allowed\_messages\_for\_many() (in module yawf.messages.allowed), 9  
 get\_allowed\_resources() (in module yawf.resources.allowed), 11  
 get\_atomical\_checkers() (yawf.permissions.BasePermissionChecker method), 8  
 get\_clarified\_instance() (yawf.base\_model.WorkflowAwareModelBase method), 3  
 get\_create\_form\_html() (in module yawf.forms), 6  
 get\_message\_specs() (in module yawf.messages.allowed), 9  
 get\_message\_specs\_for\_many() (in module yawf.messages.allowed), 9  
 get\_object\_as\_html() (in module yawf.forms), 6  
 get\_registered\_workflows() (in module yawf.\_\_init\_\_), 3  
 get\_resource() (in module yawf.resources.allowed), 11  
 get\_serializable\_value() (yawf.handlers.SerializableHandlerResult method), 7  
 get\_workflow() (in module yawf.\_\_init\_\_), 3  
 get\_workflow\_by\_instance() (in module yawf.\_\_init\_\_), 3  
 get\_workflow\_display\_name() (in module yawf.\_\_init\_\_), 3  
 get\_workflow\_name\_map() (in module yawf.\_\_init\_\_), 3  
 GroupPathEmptyError, 5

## H

Handler (class in yawf.handlers), 6

## I

id (yawf.messages.common.BasicCancelMessage attribute), 10  
 id (yawf.messages.common.BasicDeleteMessage attribute), 10  
 id (yawf.messages.common.BasicEditMessage attribute), 10  
 id (yawf.messages.common.BasicStartMessage attribute), 10  
 id (yawf.messages.spec.MessageSpec attribute), 11  
 id\_grouper (yawf.messages.spec.MessageSpec attribute), 11

IllegalStateError, 5  
 is\_grouped (yawf.messages.spec.MessageSpec attribute), 11  
 is\_transactional (yawf.effects.SideEffect attribute), 4  
 is\_valid() (yawf.messages.spec.EmptyValidator method), 10  
 is\_valid\_message() (in module yawf.messages.allowed), 9

## M

make\_common\_cancel() (in module yawf.utils), 8  
 make\_common\_start() (in module yawf.utils), 8  
 make\_common\_updater() (in module yawf.utils), 8  
 maybe\_list() (in module yawf.utils), 8  
 memoizable\_property() (in module yawf.utils), 8  
 message\_group (yawf.effects.SideEffect attribute), 4  
 message\_group (yawf.handlers.Handler attribute), 7  
 message\_id (yawf.effects.SideEffect attribute), 4  
 message\_id (yawf.handlers.Handler attribute), 7  
 message\_spec\_fabric() (in module yawf.messages.common), 10  
 MessageIgnored, 5  
 MessageSpec (class in yawf.messages.spec), 10  
 MessageSpecNotRegisteredError, 5  
 MessageSpecsTestCase (class in yawf.tests.message\_specs), 12  
 MessageValidationError, 5  
 metadefaultdict() (in module yawf.utils), 8  
 model\_diff() (in module yawf.utils), 9  
 model\_diff\_fields() (in module yawf.utils), 9

## N

name (yawf.effects.SideEffect attribute), 4  
 need\_lock\_object (yawf.messages.submessage.Submessage attribute), 11  
 NoAvailableMessagesError, 5  
 NotChecker (class in yawf.permissions), 8

## O

OldStateInconsistenceError, 5  
 optionally\_edit() (in module yawf.utils), 9  
 OrChecker (class in yawf.permissions), 8

## P

params\_wrapper() (yawf.messages.spec.MessageSpec method), 11  
 perform() (yawf.effects.SideEffect method), 4  
 perform() (yawf.handlers.ComplexStateTransition method), 7  
 perform() (yawf.handlers.Handler method), 7  
 perform() (yawf.handlers.SimpleStateTransition method), 7



- perform\_child\_checker()  
(yawf.permissions.BasePermissionChecker method), 8
- permission\_checker (yawf.handlers.Handler attribute), 7
- PermissionDeniedError, 5
- PermissionsTestCase (class in yawf.tests.permissions), 12
- post\_hook() (yawf.handlers.EditHandler method), 7
- ## R
- rank (yawf.messages.common.BasicCancelMessage attribute), 10
- rank (yawf.messages.common.BasicDeleteMessage attribute), 10
- rank (yawf.messages.common.BasicEditMessage attribute), 10
- rank (yawf.messages.spec.MessageSpec attribute), 11
- RecursiveSubmessage (class in yawf.messages.submessage), 11
- register\_workflow() (in module yawf.\_\_init\_\_), 3
- replace\_if\_exists (yawf.handlers.Handler attribute), 7
- ResourceNotFoundError, 5
- ResourcePermissionDeniedError, 5
- ## S
- select\_for\_update() (in module yawf.utils), 9
- SerializableHandlerResult (class in yawf.handlers), 7
- set\_performer() (yawf.effects.SideEffect method), 4
- set\_performer() (yawf.handlers.Handler method), 7
- setUp() (yawf.tests.permissions.PermissionsTestCase method), 12
- SideEffect (class in yawf.effects), 4
- SimpleStateTransition (class in yawf.handlers), 7
- state\_display() (yawf.base\_model.WorkflowAwareModelBase method), 3
- state\_to (yawf.handlers.SimpleStateTransition attribute), 7
- states\_from (yawf.effects.SideEffect attribute), 4
- states\_from (yawf.handlers.Handler attribute), 7
- states\_to (yawf.effects.SideEffect attribute), 4
- Submessage (class in yawf.messages.submessage), 11
- ## T
- test\_atomics() (yawf.tests.permissions.PermissionsTestCase method), 12
- test\_cache() (yawf.tests.permissions.PermissionsTestCase method), 12
- test\_checker\_expressions()  
(yawf.tests.permissions.PermissionsTestCase method), 12
- test\_custom\_sender() (yawf.tests.permissions.PermissionsTestCase method), 12
- test\_cyclic() (yawf.tests.permissions.PermissionsTestCase method), 12
- test\_fabric() (yawf.tests.message\_specs.MessageSpecsTestCase method), 12
- test\_fill\_cache() (yawf.tests.permissions.PermissionsTestCase method), 12
- test\_grouping() (yawf.tests.message\_specs.MessageSpecsTestCase method), 12
- test\_inversion() (yawf.tests.permissions.PermissionsTestCase method), 12
- test\_inversion\_invariance()  
(yawf.tests.permissions.PermissionsTestCase method), 12
- test\_repr() (yawf.tests.message\_specs.MessageSpecsTestCase method), 12
- test\_validator() (yawf.tests.message\_specs.MessageSpecsTestCase method), 12
- test\_without\_id() (yawf.tests.message\_specs.MessageSpecsTestCase method), 12
- transition() (yawf.handlers.ComplexStateTransition method), 7
- transition() (yawf.handlers.EditHandler method), 7
- type (yawf.handlers.SerializableHandlerResult attribute), 7
- ## U
- UnhandledMessageError, 6
- ## V
- validator\_cls (yawf.messages.spec.MessageSpec attribute), 11
- verb (yawf.messages.common.BasicCancelMessage attribute), 10
- verb (yawf.messages.common.BasicDeleteMessage attribute), 10
- verb (yawf.messages.common.BasicEditMessage attribute), 10
- verb (yawf.messages.common.BasicStartMessage attribute), 10
- verb (yawf.messages.spec.MessageSpec attribute), 11
- ## W
- workflow (yawf.base\_model.WorkflowAwareModelBase attribute), 3
- workflow\_type\_display()  
(yawf.base\_model.WorkflowAwareModelBase method), 4
- WorkflowAlreadyRegisteredError, 6
- WorkflowAwareModelBase (class in yawf.base\_model), 3
- WorkflowChoices (class in yawf.\_\_init\_\_), 3
- WorkflowNotLoadedError, 6
- WorkflowResource (class in yawf.resources.base), 11
- WrongHandlerResultError, 6

## Y

- yawf.\_\_init\_\_ (module), 3
- yawf.allowed (module), 3
- yawf.base\_model (module), 3
- yawf.config (module), 4
- yawf.effects (module), 4
- yawf.exceptions (module), 5
- yawf.forms (module), 6
- yawf.handlers (module), 6
- yawf.messages (module), 9
  - yawf.messages.allowed (module), 9
  - yawf.messages.common (module), 10
  - yawf.messages.spec (module), 10
  - yawf.messages.submessage (module), 11
- yawf.permissions (module), 8
- yawf.resources (module), 11
  - yawf.resources.allowed (module), 11
  - yawf.resources.base (module), 11
- yawf.signals (module), 8
- yawf.tests (module), 12
  - yawf.tests.message\_specs (module), 12
  - yawf.tests.permissions (module), 12
- yawf.utils (module), 8
- yawf.version (module), 9
- YawfException, 6