

# BaseWorkflow

The ``BaseWorkflow`` class serves as a foundational structure for defining and managing workflows. It allows users to add, remove, update, and manage tasks and agents within a workflow, offering flexibility and extensibility for various applications.

### Key Concepts

- **Agents**: Entities participating in the workflow.
- **Tasks**: Units of work to be executed within the workflow.
- **Models**: Computational models used within the workflow.
- **Workflow State**: The state of the workflow, which can be saved and restored.

## Attributes

### Arguments

Argument	Type	Default	Description
----- ----- ----- -----			
<code>`agents`</code>	<code>`List[Agent]`</code>	<code>`None`</code>	A list of agents participating in the workflow.
<code>`task_pool`</code>	<code>`List[Task]`</code>	<code>`None`</code>	A list of tasks in the workflow.
<code>`models`</code>	<code>`List[Any]`</code>	<code>`None`</code>	A list of models used in the workflow.
<code>`*args`</code>			Variable length argument list.
<code>`**kwargs`</code>			Arbitrary keyword arguments.

### Attributes

Attribute	Type	Description
----- ----- -----		
`agents`	`List[Agent]`	A list of agents participating in the workflow.
`task_pool`	`List[Task]`	A list of tasks in the workflow.
`models`	`List[Any]`	A list of models used in the workflow.

## Methods

### add\_task

Adds a task or a list of tasks to the task pool.

**\*\*Arguments:\*\***

Parameter	Type	Default	Description
----- ----- ----- -----			
`task`	`Task`	`None`	A single task to add.
`tasks`	`List[Task]`	`None`	A list of tasks to add.

**\*\*Raises:\*\***

- `ValueError`: If neither task nor tasks are provided.

**\*\*Examples:\*\***

```
```python
```

```
workflow = BaseWorkflow()
```

```
task1 = Task(description="Task 1")
```

```
task2 = Task(description="Task 2")
```

```
# Adding a single task
```

```
workflow.add_task(task=task1)
```

```
# Adding multiple tasks
```

```
workflow.add_task(tasks=[task1, task2])
```

```
```
```

```
### add_agent
```

Adds an agent to the workflow.

**\*\*Arguments:\*\***

| Parameter | Type | Description |
|-----------|------|-------------|
|-----------|------|-------------|

|                   |
|-------------------|
| ----- ----- ----- |
|-------------------|

|         |         |                                   |
|---------|---------|-----------------------------------|
| `agent` | `Agent` | The agent to add to the workflow. |
|---------|---------|-----------------------------------|

**\*\*Examples:\*\***

```
```python
```

```
workflow = BaseWorkflow()
```

agent = Agent(name="Agent 1")

# Adding an agent to the workflow

workflow.add\_agent(agent=agent)

...

### run

Abstract method to run the workflow.

### \_\_sequential\_loop

Abstract method for the sequential loop.

### \_\_log

Logs a message if verbose mode is enabled.

**\*\*Arguments:\*\***

Parameter	Type	Description
message	str	The message to log.

### \_\_str\_\_

Returns a string representation of the workflow.

```
### __repr__
```

Returns a string representation of the workflow for debugging.

```
### reset
```

Resets the workflow by clearing the results of each task.

```
**Examples:**
```

```
```python
workflow = BaseWorkflow()

workflow.reset()
```
```

```
### get_task_results
```

Returns the results of each task in the workflow.

```
**Returns:**
```

| Return Type   Description |                                           |
|---------------------------|-------------------------------------------|
| ----- -----               |                                           |
| `Dict[str, Any]`          | The results of each task in the workflow. |

**\*\*Examples:\*\***

```
```python
workflow = BaseWorkflow()

results = workflow.get_task_results()

...

```

**### remove\_task**

Removes a task from the workflow.

**\*\*Arguments:\*\***

Parameter	Type	Description
----- ----- -----		
`task`	`str`	The description of the task to remove.

**\*\*Examples:\*\***

```
```python
workflow = BaseWorkflow()

workflow.remove_task(task="Task 1")

...

```

**### update\_task**

Updates the arguments of a task in the workflow.

**\*\*Arguments:\*\***

Parameter	Type	Description
`task`	`str`	The description of the task to update.
`**updates`		The updates to apply to the task.

**\*\*Raises:\*\***

- `ValueError`: If the task is not found in the workflow.

**\*\*Examples:\*\***

```
```python
workflow = BaseWorkflow()

task = Task(description="Task 1", kwargs={"param": 1})

# Adding a task to the workflow
workflow.add_task(task=task)

# Updating the task
workflow.update_task("Task 1", param=2)
```
```

### delete\_task

Deletes a task from the workflow.

**\*\*Arguments:\*\***

| Parameter | Type | Description |
|-----------|------|-------------|
|-----------|------|-------------|

|       |       |       |
|-------|-------|-------|
| ----- | ----- | ----- |
|-------|-------|-------|

|        |       |                                        |
|--------|-------|----------------------------------------|
| `task` | `str` | The description of the task to delete. |
|--------|-------|----------------------------------------|

**\*\*Raises:\*\***

- `ValueError`: If the task is not found in the workflow.

**\*\*Examples:\*\***

```
```python
```

```
workflow = BaseWorkflow()
```

```
task = Task(description="Task 1")
```

```
# Adding a task to the workflow
```

```
workflow.add_task(task=task)
```

```
# Deleting the task
```

```
workflow.delete_task("Task 1")
```



...

### save\_workflow\_state

Saves the workflow state to a json file.

**Arguments:**

Parameter	Type	Default	Description
filepath	Optional[str]	"sequential_workflow_state.json"	The path to save the workflow state to.

**Examples:**

```
python
workflow = BaseWorkflow()
workflow.save_workflow_state(filepath="workflow_state.json")
...
```

### add\_objective\_to\_workflow

Adds an objective to the workflow.

**Arguments:**

Parameter	Type	Description
task	str	The description of the task.
kwargs		Additional keyword arguments for the task.

**Examples:**

```
python
workflow = BaseWorkflow()
workflow.add_objective_to_workflow(task="New Objective", agent=agent, args=[], kwargs={})

```

**load\_workflow\_state**

Loads the workflow state from a json file and restores the workflow state.

**Arguments:**

Parameter	Type	Default	Description
filepath	str	None	The path to load the workflow state from.

**Examples:**

```
python
workflow = BaseWorkflow()

```

```
workflow.load_workflow_state(filepath="workflow_state.json")
...

```

### workflow\_dashboard

Displays a dashboard for the workflow.

**Arguments:**

Parameter	Type	Description
kwargs	dict	Additional keyword arguments to pass to the dashboard.

**Examples:**

```
python
workflow = BaseWorkflow()
workflow.workflow_dashboard()
...

```

### workflow\_bootup

Initializes the workflow.

**Examples:**

```
```python
```

```
workflow = BaseWorkflow()
```

```
workflow.workflow_bootup()
```

```
```
```