msaSDK Module

`.utils.scheduler`

Classes

MSAScheduler

Bases: `Rocketry`

Scheduling application

> **Note**
>
> If enabled, it creates a memory log as this info is neede for some conditions. So ensure the memory doesnt run full Define a task that clears the log
>
> ```
> repo: MemoryRepo = self.scheduler.session.get_repo()
> repo.collection.clear()
> ```

Execution conditions

Execution on fixed time interval Syntax

```
[hourly | daily | weekly | monthly]
[hourly | daily | weekly | monthly] between <start> and <end>
[hourly | daily | weekly | monthly] [before | after | starting] <time>
```

Fixed time interval Syntax

```
time of [hour | day | week | month] between <start> and <end>
time of [hour | day | week | month] [after | before] <time>
```

Timedelta Syntax

```
every <timedelta>
```

Task Status Syntax

```
has [succeeded | failed | finished | started | terminated] [this hour | today | this
week | this month] between <start> and <end>
has [succeeded | failed | finished | started | terminated] [this hour | today | this
week | this month] [before | after] <time>
has [succeeded | failed | finished | started | terminated] past <timedelta>
```

Task Dependence Syntax

```
after task '<task>'
after task '<task>' [succeeded | failed | finished | terminated]
after tasks '<task 1>', '<task 2>' ...
after tasks '<task 1>', '<task 2>' ... [succeeded | failed | finished]
after any tasks '<task 1>', '<task 2>' ... [succeeded | failed | finished]
```

Examples:

```python
# Execution on fixed time interval
app.scheduler.task("hourly")
app.scheduler.task("daily between 22:00 and 23:00")
app.scheduler.task("weekly before Friday")
app.scheduler.task("monthly starting 3rd")

# Fixed time interval
app.scheduler.task("time of hour before 45:00")
app.scheduler.task("time of day between 10:00 and 16:00")
app.scheduler.task("time of week after Monday")
app.scheduler.task("time of month after 5th")

# Time delta
app.scheduler.task("every 1 hour")
app.scheduler.task("every 30 sec")
app.scheduler.task("every 2 hours, 30 minutes")
app.scheduler.task("every 1 day, 12 hour, 30 min, 20 sec")

# Task Status
app.scheduler.task("has succeeded this hour")
app.scheduler.task("has failed today between 08:00 and 16:00")
app.scheduler.task("has started this week before Friday")
app.scheduler.task("has terminated this month after 6th")
app.scheduler.task("has succeeded past 2 hours, 30 minutes")

# Task Dependence
app.scheduler.task("after task 'a_task'")
app.scheduler.task("after task 'a_task' succeeded")
app.scheduler.task("after task 'a_task' failed")
app.scheduler.task("after task 'a_task' finished")
app.scheduler.task("after tasks 'a_task', 'another_task' finished")
```

Functions

__init__

```
__init__(
    session: Session = None,
    logger_repo: Optional[BaseRepo] = None,
    execution=None,
    **kwargs
)
```

Functions

get_time

```
get_time(local_time_zone = 'UTC')
```

get_time_stamp

```
get_time_stamp(
    local_time_zone="UTC", time_format="HMS"
)
```

Last update: September 16, 2022

Created: September 16, 2022