

system.user.getSchedule

This function is used in **Python Scripting**.

Description

Returns a specific schedule.

Client Permission Restrictions

This scripting function has no [Client Permission](#) restrictions.

Syntax

system.user.getSchedule(scheduleNames)

- Parameters

String scheduleName - The name of the schedule to return. Case-sensitive

- Returns

AbstractScheduleModel - The schedule, which can be a BasicSchedule, CompositeSchedule, or another type registered by a module, or None if not found.

Once the schedule is returned, functions of the resulting object are used to get various information from that schedule. Which functions are available depend on the type of schedule returned. See examples in the SDK documentation for an [abstract schedule](#), [basic schedule](#), or [composite schedule](#).

- Scope

All

Code Examples

Code Snippet

```
# This example will get a schedule and print info about it:

# This function handles recursive printing of the different schedule types. Modules can
register more types than listed here.
def printScheduleInfo(aSchedule):
    if aSchedule.getType() == "basic schedule":
        print "Basic schedule type: ",aSchedule.getName(), aSchedule.getDescription(),
aSchedule.isAllDays(), aSchedule.isObserveHolidays()
    elif aSchedule.getType() == "composite schedule":
        compositePieces = aSchedule.getModels()
        print "Composite schedule type:",aSchedule.getName(), aSchedule.
getDescription(), " which is made up of..."
        for piece in compositePieces:
            printScheduleInfo(piece)
    else:
        print "Other schedule type: ", aSchedule.getName(), aSchedule.
getDescription(), aSchedule.getType(), aSchedule.isObserveHolidays()

# The main function
scheduleName = "MySchedule"
schedule = system.user.getSchedule(scheduleName)
if schedule == None:
    print "Schedule", scheduleName, "was not found"
else:
    printScheduleInfo(schedule)
```

Output

```
Basic schedule type: MySchedule A description False True
```