Python

May 22, 2020

Contents

Tuber is an event management system. It helps keep track of your event staff, their jobs and shifts, departments, and attendees.

CHAPTER 1

Tuber Module

1.1 tuber

The base module for Tuber

tuber.**migrate**()

1.2 tuber.static

tuber.static.**default**(*e*)

tuber.static.**files**(*path*)

tuber.static.**home**()

1.3 tuber.permissions

exception tuber.permissions.**PermissionDenied**(*message*, *status_code=None*, *payload=None*)

status_code = 403

to_dict()

tuber.permissions.**check_permission**(*permission=None*, *event=0*, *department=0*)

tuber.permissions.**get_user**()

tuber.permissions.**handle_permission_denied**(*error*)

CHAPTER 2

Tuber API

2.1 Endpoints

2.1.1 Hotels

GET /api/hotels
Retrieve a list of hotels.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/hotels
```

Example Response

```
[{
    "id": 1,
    "name": "The Gaylord",
    "description": "An awesome venue in Maryland!"
}]
```

Query Parameters

- **full** (*string*) – If true returns a list of objects. If false, returns a list of id numbers.

2.1.2 Shifts

GET /api/events/<id>/jobs/available
Retrieve the list of shifts that are available to either the current user or a specific badge.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/events/<id>/jobs/
↪available
```

Example Response

```
[
    {
        "job": {
            "id": 1,
            "name": "Do the thing",
            "description": "",
            "department": null,
            "documentation": "",
            "method": {},
            "signuprules": {},
            "sticky": false,
            "schedules": [],
            "scheduleevents": [],
            "roles": [],
            "shifts": [1]
        },
        "shifts": [
            {
                "blocks": [
                    "Shift is full."
                ],
                "id": 1,
                "job": 1,
                "schedule": null,
                "schedule_event": null,
                "starttime": "2020-05-22T21:15:52.159726",
                "duration": 3600.0,
                "slots": 4,
                "filledslots": 0,
                "weighting": 1.0
            }
        ]
    }
]
```

Query Parameters

- **badge** (*string*) – If provided then the result will be the shifts available to the given badge.

POST /api/events/<id>/shifts/<id>/signup

Sign up for the given shift. If you want to sign up a different badge then the post body should be an object with key badge set the the desired badge id.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" --header 'Content-Type: application/json' -X␣
↪POST --data '{"badge": 4}' https://tuber.magfest.org/api/events/<id>/shifts/<id>
↪/signup
```

Example Response

```
{
    "shift": {
        "id": 1,
        "job": 1,
        "schedule": null,
        "schedule_event": null,
        "starttime": "2020-05-22T21:15:52.159726",
        "duration": 3600.0,
        "slots": 4,
        "filledslots": 0,
        "weighting": 1.0
    },
    "shift_signup": {
        "id": 1,
        "badge": 1,
        "job": 1,
        "shift": 1,
        "schedule": null,
        "scheduleevent": null,
        "starttime": "2020-05-22T21:15:52.159726",
        "duration": 3600.0
    },
    "shift_assignment": {
        "id": 1,
        "badge": 1,
        "shift": 1,
        "signuptime": "2020-05-22T21:15:52.159726"
    }
}
```


POST /api/events/<id>/jobs/<id>/dryrun

This endpoint lets you check what the resulting shifts based on a hypothetical job definition. Calling this endpoint will not commit anything to the database, but will let you see what would have resulted from a PATCH to the corresponding job.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" --header 'Content-Type: application/json' -X␣
↪POST --data '{"method": {"name": "copy"}}' https://tuber.magfest.org/api/events/
↪<id>/jobs/<id>/dryrun
```

Example Response


```
[{
    "id": 1,
    "job": 1,
    "schedule": null,
    "schedule_event": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "slots": 4,
    "filledslots": 0,
    "weighting": 1.0
}]
```

Schedule Events

GET /api/scheduleevents
Retrieve a list of scheduleevents.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/scheduleevents
```

Example Response

```
[{
    "id": 1,
    "name": "Someone's panel",
    "description": "",
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "schedule": 1
}]
```

Query Parameters

- **full** (*string*) – If true returns a list of objects. If false, returns a list of id numbers.

GET /api/scheduleevents/<id>
Retrieve a single scheduleevent.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/scheduleevents/1
```

Example Response

```
{
    "id": 1,
    "name": "Someone's panel",
    "description": "",
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "schedule": 1
}
```

POST /api/scheduleevents
Create a new scheduleevent object.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X POST --header "Content-Type: application/json
↪" --data '{"name": "Someone's panel"}' https://tuber.magfest.org/api/
↪scheduleevents
```

Example Response

```
{
    "id": 1,
    "name": "Someone's panel",
    "description": "",
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "schedule": 1
}
```

PATCH /api/scheduleevents/<id>

Update a scheduleevent.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X PATCH --header "Content-Type: application/
↪json" --data '{"description": "Really Cool"}' https://tuber.magfest.org/api/
↪scheduleevents/<id>
```

Example Response

```
{
    "id": 1,
    "name": "Someone's panel",
    "description": "Really Cool",
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "schedule": 1
}
```

DELETE /api/scheduleevents/<id>

Delete a scheduleevent.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X DELETE https://tuber.magfest.org/api/
↪scheduleevents/1
```

Example Response

```
{
    "id": 1,
    "name": "Someone's panel",
    "description": "",
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "schedule": 1
}
```

Jobs

GET /api/events/1/jobs

Retrieve a list of jobs.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/events/1/jobs
```

Example Response

```
[{
    "id": 1,
    "name": "Do the thing",
    "description": "",
    "department": null,
    "documentation": "",
    "method": {},
    "signuprules": {},
    "sticky": false,
    "schedules": [],
    "scheduleevents": [],
    "roles": [],
    "shifts": []
}]
```

Query Parameters

- **full** (*string*) – If true returns a list of objects. If false, returns a list of id numbers.

GET /api/events/1/jobs/<id>

Retrieve a single job.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/events/1/jobs/1
```

Example Response

```
{
    "id": 1,
    "name": "Do the thing",
    "description": "",
    "department": null,
    "documentation": "",
    "method": {},
    "signuprules": {},
    "sticky": false,
    "schedules": [],
    "scheduleevents": [],
    "roles": [],
    "shifts": []
}
```

POST /api/events/1/jobs

Create a new job object.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X POST --header "Content-Type: application/json
↪" --data '{"name": "Do the thing"}' https://tuber.magfest.org/api/events/1/jobs
```

Example Response

```
{
    "id": 1,
    "name": "Do the thing",
    "description": "",
    "department": null,
    "documentation": "",
    "method": {},
    "signuprules": {},
    "sticky": false,
    "schedules": [],
    "scheduleevents": [],
    "roles": [],
    "shifts": []
}
```

PATCH /api/events/1/jobs/<id>

Update a job.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X PATCH --header "Content-Type: application/
↪json" --data '{"description": "Really Cool"}' https://tuber.magfest.org/api/
↪events/1/jobs/<id>
```

Example Response

```
{
    "id": 1,
    "name": "Do the thing",
    "description": "Really Cool",
    "department": null,
    "documentation": "",
    "method": {},
    "signuprules": {},
    "sticky": false,
    "schedules": [],
    "scheduleevents": [],
    "roles": [],
    "shifts": []
}
```

DELETE /api/events/1/jobs/<id>

Delete a job.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X DELETE https://tuber.magfest.org/api/events/
↪1/jobs/1
```

Example Response

```
{
    "id": 1,
    "name": "Do the thing",
    "description": "",
    "department": null,
    "documentation": "",
    "method": {},
    "signuprules": {},
    "sticky": false,
    "schedules": [],
    "scheduleevents": [],
    "roles": [],
    "shifts": []
}
```

Shifts

GET /api/shifts
Retrieve a list of shifts.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/shifts
```

Example Response

```
[{
    "id": 1,
    "job": 1,
    "schedule": null,
    "schedule_event": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "slots": 4,
    "filledslots": 0,
    "weighting": 1.0
}]
```

Query Parameters

- **full** (*string*) – If true returns a list of objects. If false, returns a list of id numbers.

GET /api/shifts/<id>
Retrieve a single shift.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/shifts/1
```

Example Response

```
{
    "id": 1,
    "job": 1,
    "schedule": null,
    "schedule_event": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "slots": 4,
    "filledslots": 0,
    "weighting": 1.0
}
```


POST /api/shifts

Create a new shift object.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X POST --header "Content-Type: application/json
↪" --data '{"job": 1, "schedule": 2, "slots": 4}' https://tuber.magfest.org/api/
↪shifts
```

Example Response


```
{
    "id": 1,
    "job": 1,
    "schedule": 2,
    "schedule_event": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0,
    "slots": 4,
    "filledslots": 0,
    "weighting": 1.0
}
```


PATCH /api/shifts/<id>

Update a shift.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X PATCH --header "Content-Type: application/
↪json" --data '{"duration": 7200.0}' https://tuber.magfest.org/api/shifts/<id>
```

Example Response


```
{
    "id": 1,
    "job": 1,
    "schedule": 2,
    "schedule_event": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 7200.0,
    "slots": 4,
    "filledslots": 0,
```


(continues on next page)

(continued from previous page)

```
    "weighting": 1.0
}
```

DELETE /api/shifts/<id>

Delete a shift.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X DELETE https://tuber.magfest.org/api/shifts/1
```

Example Response

```
{
    "id": 1,
    "job": 1,
    "schedule": 2,
    "schedule_event": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 7200.0,
    "slots": 4,
    "filledslots": 0,
    "weighting": 1.0
}
```

Shift Assignments

GET /api/shiftassignments

Retrieve a list of shiftassignments.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/shiftassignments
```

Example Response

```
[{
    "id": 1,
    "badge": 1,
    "shift": 1,
    "signuptime": "2020-05-22T21:15:52.159726"
}]
```

Query Parameters

- **full** (*string*) – If true returns a list of objects. If false, returns a list of id numbers.

GET /api/shiftassignments/<id>

Retrieve a single shiftassignment.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/shiftassignments/1
```

Example Response

```
{
    "id": 1,
    "badge": 1,
    "shift": 1,
    "signuptime": "2020-05-22T21:15:52.159726"
}
```

`POST /api/shiftassignments`

Create a new shiftassignment object.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X POST --header "Content-Type: application/json
↪" --data '{"badge": 1, "shift": 1}' https://tuber.magfest.org/api/
↪shiftassignments
```

Example Response

```
{
    "id": 1,
    "badge": 1,
    "shift": 1,
    "signuptime": "2020-05-22T21:15:52.159726"
}
```

`DELETE /api/shiftassignments/<id>`

Delete a shiftassignment.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X DELETE https://tuber.magfest.org/api/
↪shiftassignments/1
```

Example Response

```
{
    "id": 1,
    "badge": 1,
    "shift": 1,
    "signuptime": "2020-05-22T21:15:52.159726"
}
```

Shift Signups

`GET /api/shiftsignups`

Retrieve a list of shiftsignups.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/shiftsignups
```

Example Response

```
[{
    "id": 1,
    "badge": 1,
    "job": 1,
    "shift": 1,
    "schedule": null,
    "scheduleevent": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0
}]
```

Query Parameters

- **full** (*string*) – If true returns a list of objects. If false, returns a list of id numbers.

GET /api/shiftsignups/<id>
Retrieve a single shiftsignup.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" https://tuber.magfest.org/api/shiftsignups/1
```

Example Response

```
{
    "id": 1,
    "badge": 1,
    "job": 1,
    "shift": 1,
    "schedule": null,
    "scheduleevent": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0
}
```

POST /api/shiftsignups
Create a new shiftsignup object.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X POST --header "Content-Type: application/json
↪" --data '{"shift": 2}' https://tuber.magfest.org/api/shiftsignups
```

Example Response

```
{
    "id": 1,
    "badge": 1,
    "job": 1,
    "shift": 2,
```

(continues on next page)

(continued from previous page)

```
    "schedule": null,
    "scheduleevent": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0
}
```

PATCH /api/shiftsignups/<id>

Update a shiftsignup.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X PATCH --header "Content-Type: application/
↪json" --data '{"shift": 1}' https://tuber.magfest.org/api/shiftsignups/<id>
```

Example Response

```
{
    "id": 1,
    "badge": 1,
    "job": 1,
    "shift": 1,
    "schedule": null,
    "scheduleevent": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0
}
```

DELETE /api/shiftsignups/<id>

Delete a shiftsignup.

Example Request

Bash

```
$ curl -H "X-Auth-Token: <Token>" -X DELETE https://tuber.magfest.org/api/
↪shiftsignups/1
```

Example Response

```
{
    "id": 1,
    "badge": 1,
    "job": 1,
    "shift": 1,
    "schedule": null,
    "scheduleevent": null,
    "starttime": "2020-05-22T21:15:52.159726",
    "duration": 3600.0
}
```

2.2 Resources

2.2.1 Hotels

class tuber.api.hotels.**HotelLocationSchema**(**args*, ***kwargs*)

HotelLocation:

Schema for a HotelLocation

```
{
  "address": "",
  "event": 1,
  "id": 1,
  "name": ""
}
```

class tuber.api.hotels.**HotelRoomBlockSchema**(**args*, ***kwargs*)

HotelRoomBlock:

Schema for a HotelRoomBlock

```
{
  "description": "",
  "event": 1,
  "id": 1,
  "name": ""
}
```

class tuber.api.hotels.**HotelRoomNightSchema**(**args*, ***kwargs*)

HotelRoomNight:

Schema for a HotelRoomNight

```
{
  "event": 1,
  "hidden": false,
  "id": 1,
  "name": "",
  "restricted": false,
  "restriction_type": ""
}
```

class tuber.api.hotels.**HotelRoomRequestSchema**(**args*, ***kwargs*)

HotelRoomRequest:

The requested preferences for a staff hotel room.

Parameters

- **badge** (*int.*) – The id of the badge that created this request
- **declined** (*bool.*) – True if the user has declined a hotel room

```
{
  "badge": 1,
  "declined": false,
  "id": 1,
  "noise_level": "",
  "notes": "",
```

(continues on next page)

(continued from previous page)


```
  "prefer_department": false,
  "prefer_single_gender": false,
  "room_night_justification": "",
  "sleep_time": "",
  "smoke_sensitive": false
}
```


class tuber.api.hotels.**HotelRoomSchema**(**args*, ***kwargs*)

HotelRoom:

Schema for a HotelRoom


```
{
  "completed": false,
  "hotel_block": 1,
  "hotel_location": 1,
  "id": 1,
  "messages": "",
  "name": "",
  "notes": ""
}
```


2.2.2 Emails

class tuber.api.emails.**EmailReceiptSchemaRead**(**args*, ***kwargs*)

EmailReceipt:

Schema for a EmailReceipt


```
{
  "body": "",
  "email": 1,
  "from_address": "",
  "id": 1,
  "source": 1,
  "subject": "",
  "timestamp": "2020-05-22T23:20:07.574831",
  "to_address": ""
}
```


class tuber.api.emails.**EmailSchemaRead**(**args*, ***kwargs*)

Email:

Schema for a Email


```
{
  "active": false,
  "body": "",
  "code": "",
  "description": "",
  "event": 1,
  "id": 1,
  "name": "",
  "send_once": false,
  "source": 1,
```


(continues on next page)

(continued from previous page)

```
  "subject": ""
}
```

class tuber.api.emails.**EmailSchemaWrite**(**args*, ***kwargs*)

Email:

Schema for a Email

```
{
  "active": false,
  "body": "",
  "code": "",
  "description": "",
  "event": 1,
  "id": 1,
  "name": "",
  "send_once": false,
  "source": 1,
  "subject": ""
}
```

class tuber.api.emails.**EmailSourceSchemaRead**(**args*, ***kwargs*)

EmailSource:

Schema for a EmailSource

```
{
  "active": false,
  "address": "",
  "description": "",
  "event": 1,
  "id": 1,
  "name": "",
  "region": "",
  "ses_access_key": "",
  "ses_secret_key": ""
}
```

class tuber.api.emails.**EmailSourceSchemaWrite**(**args*, ***kwargs*)

EmailSource:

Schema for a EmailSource

```
{
  "active": false,
  "address": "",
  "description": "",
  "event": 1,
  "id": 1,
  "name": "",
  "region": "",
  "ses_access_key": "",
  "ses_secret_key": ""
}
```

2.2.3 Badges

class tuber.api.badges.**BadgeSchema**(**args*, ***kwargs*)

Badge:

Schema for a Badge

```
{
  "badge_type": 1,
  "email": "",
  "event": 1,
  "first_name": "",
  "id": 1,
  "last_name": "",
  "legal_name": "",
  "legal_name_matches": false,
  "printed_name": "",
  "printed_number": "",
  "search_name": "",
  "uber_id": "",
  "user": 1
}
```

class tuber.api.badges.**BadgeTypeSchema**(**args*, ***kwargs*)

BadgeType:

Schema for a BadgeType

```
{
  "description": "",
  "id": 1,
  "name": ""
}
```

class tuber.api.badges.**DepartmentSchema**(**args*, ***kwargs*)

Department:

Schema for a Department

```
{
  "description": "",
  "id": 1,
  "name": "",
  "uber_id": ""
}
```

class tuber.api.badges.**RibbonTypeSchema**(**args*, ***kwargs*)

RibbonType:

Schema for a RibbonType

```
{
  "description": "",
  "id": 1,
  "name": ""
}
```

2.2.4 Events

class tuber.api.events.**EventSchema**(**args*, ***kwargs*)

Event:

Schema for a Event

```
{
  "description": "",
  "id": 1,
  "name": ""
}
```

2.2.5 Users

class tuber.api.users.**GrantSchema**(**args*, ***kwargs*)

Grant:

Schema for a Grant

```
{
  "department": 1,
  "id": 1,
  "role": 1,
  "user": 1
}
```

class tuber.api.users.**PermissionSchema**(**args*, ***kwargs*)

Permission:

Schema for a Permission

```
{
  "id": 1,
  "operation": "",
  "role": 1
}
```

class tuber.api.users.**RoleSchema**(**args*, ***kwargs*)

Role:

Schema for a Role

```
{
  "description": "",
  "event": 1,
  "id": 1,
  "name": ""
}
```

class tuber.api.users.**UserSchema**(**args*, ***kwargs*)

User:

Schema for a User

```
{
  "active": false,
```

(continues on next page)

(continued from previous page)

```
  "email": "",
  "id": 1,
  "username": ""
}
```

class tuber.api.users.**UserWriteSchema**(**args*, ***kwargs*)

User:

Schema for a User

```
{
  "active": false,
  "email": "",
  "id": 1,
  "username": ""
}
```

2.2.6 shifts

class tuber.api.shifts.**JobSchema**(**args*, ***kwargs*)

Job:

A Job describes something we might ask a volunteer to do. It holds the actual job description for the human as well as scheduling rules for the system to create Shifts. All Shifts are linked to a Job.

```
{
  "department": 1,
  "description": "",
  "documentation": "",
  "id": 1,
  "method": null,
  "name": "",
  "signuprules": null,
  "sticky": false
}
```

class tuber.api.shifts.**ScheduleEventSchema**(**args*, ***kwargs*)

ScheduleEvent:

A ScheduleEvent is used to store when something will happen during an Event.

```
{
  "description": "",
  "duration": null,
  "id": 1,
  "name": "",
  "schedule": 1,
  "starttime": "2020-05-22T23:20:07.576663"
}
```

class tuber.api.shifts.**ScheduleSchema**(**args*, ***kwargs*)

Schedule:

A Schedule is used to store ScheduleEvents that are used for shift creation and the public event schedule.

```
{
  "description": "",
  "id": 1,
  "name": "",
  "tags": null
}
```


class tuber.api.shifts.**ShiftAssignmentSchema**(**args*, ***kwargs*)

ShiftAssignment:

A ShiftAssignment connect badges to shifts that they will work. They store the current state, and may be blown away when jobs or schedules are changed without direct user intervention.


```
{
  "badge": 1,
  "id": 1,
  "shift": 1
}
```


class tuber.api.shifts.**ShiftSchema**(**args*, ***kwargs*)

Shift:

A Shift is a block of time that a staffer can sign up to work. All Shifts are linked to a Job.


```
{
  "duration": null,
  "filledslots": 1,
  "id": 1,
  "job": 1,
  "schedule": 1,
  "schedule_event": 1,
  "slots": 1,
  "starttime": "2020-05-22T23:20:07.577305",
  "weighting": null
}
```


class tuber.api.shifts.**ShiftSignupSchema**(**args*, ***kwargs*)

ShiftSignup:

A ShiftSignup tracks the intent of a user to signup for a shift. This is different than a ShiftAssignment. A ShiftSignup persists even if the underlying shift is removed, so that the user's desires are still known if a new, similar shift is created. Thus it must hold a copy of the associated shift so that it can be compared to potential replacement shifts.


```
{
  "badge": 1,
  "duration": null,
  "id": 1,
  "job": 1,
  "schedule": 1,
  "shift": 1,
  "starttime": "2020-05-22T23:20:07.577910"
}
```


tuber.api.shifts.**reschedule_job**(*job*, *schedule_event=None*)

Regenerates the shifts associated with this job. If a schedule_event is passed then it will only regenerate overlapping shifts.

Python Module Index

t

/api

GET /api/events/1/jobs, **??**
GET /api/events/1/jobs/<id>, **??**
GET /api/events/<id>/jobs/available, **??**
GET /api/events/<id>/schedules, **??**
GET /api/events/<id>/schedules/<id>, **??**
GET /api/hotels, **??**
GET /api/scheduleevents, **??**
GET /api/scheduleevents/<id>, **??**
GET /api/shiftassignments, **??**
GET /api/shiftassignments/<id>, **??**
GET /api/shifts, **??**
GET /api/shifts/<id>, **??**
GET /api/shiftsignups, **??**
GET /api/shiftsignups/<id>, **??**
POST /api/events/1/jobs, **??**
POST /api/events/<id>/jobs/<id>/dryrun, **??**
POST /api/events/<id>/schedules, **??**
POST /api/events/<id>/shifts/<id>/signup, **??**
POST /api/scheduleevents, **??**
POST /api/shiftassignments, **??**
POST /api/shifts, **??**
POST /api/shiftsignups, **??**
DELETE /api/events/1/jobs/<id>, **??**
DELETE /api/events/<id>/schedules/<id>, **??**
DELETE /api/scheduleevents/<id>, **??**
DELETE /api/shiftassignments/<id>, **??**
DELETE /api/shifts/<id>, **??**
DELETE /api/shiftsignups/<id>, **??**
PATCH /api/events/1/jobs/<id>, **??**
PATCH /api/events/<id>/schedules/<id>, **??**
PATCH /api/scheduleevents/<id>, **??**
PATCH /api/shifts/<id>, **??**
PATCH /api/shiftsignups/<id>, **??**

Index

U