

---

# **Yolapy Documentation**

*Release 0.5.0*

**Yola**

**Jul 03, 2017**



---

## Contents

---

<b>1 Service Class</b>	<b>1</b>
<b>2 Indices and tables</b>	<b>7</b>
<b>Python Module Index</b>	<b>9</b>



The `Yola` class is a thin wrapper over the Yola API.

```
class yolapy.services.Yola (**kwargs)
    Client for Yola's API.
```

If using `yolapy.configuration`:

```
configure(
    url='https://wl.yola.net/',
    auth=('username', 'password'))
yola = Yola()
yola.get_user('user_id')
```

Or configured manually:

```
yola = Yola(
    url='https://wl.yola.net/',
    auth=('username', 'password'))
yola.get_user('user_id')
```

When appropriate, successful responses will return parsed json objects.

Failures will raise instances of `demands.HTTPServiceError`.

```
__init__ (**kwargs)
```

Initialize with optional headers.

Auth and url defaults are pulled from `yolapy.configuration`.

Passed arguments will override configuration:

```
Yola(headers={'Header-Name': 'value'})
```

## Partners

The following methods are mixed in to the `Yola` class to provide access to the Partner resource:

**class** `yolapy.resources.partner.PartnerResourceMixin`  
Methods for managing Partner resources.

**create\_partner** (*\*\*attributes*)  
Create a partner.

```
>>> partner = yola.create_partner(
    id='COMPANY_ID',
    name='Company Name',
    parent_partner_id='PARENT_ID',
    properties={'name': 'value'})
>>> partner['name']
'Company Name'
```

See <https://wl.qa.yola.net/partners/> for available properties.

**delete\_partner** (*partner\_id*)  
Delete a partner.

```
>>> yola.delete_partner('PARTNER_ID')
```

**get\_partner** (*partner\_id*)  
Return details for a particular partner.

```
>>> partner = yola.get_partner('PARTNER_ID')
>>> partner['name']
'Company Name'
```

**list\_partners** (*\*\*options*)  
Return paginated list of partners.

```
>>> yola.list_partners()
{
  'count': 999,
  'previous': None,
  'next': 'https://wl.qa.yola.net/pr/partners/?page=2',
  'results': [{...}, {...}, ...]
}
```

You can pass `page_size` and `page` as keyword arguments:

```
>>> yola.list_partners(page_size=50, page=2)
```

**update\_partner** (*partner\_id*, *\*\*attributes*)  
Update a partner.

```
>>> yola.update_partner('PARTNER_ID', name='New name')
```

## Users

The following methods are mixed in to the `Yola` class to provide access to the User resource:

**class** `yolapy.resources.user.UserResourceMixin`

Methods for managing User resources.

**create\_user** (\*\**attributes*)

Create a user.

```
>>> user = yola.create_user(
    name='John',
    surname='Smith',
    email='johnsmith@example.com',
    partner_id='WL_PARTNER_ID',
    preferences={'preference_name': 'preference_value'})
>>> user['name']
'John'
```

**delete\_user** (*user\_id*)

Delete a user.

```
>>> yola.delete_user('user_id')
```

**get\_sso\_create\_site\_url** (*user\_id*, *domain*, *locale=None*)

Get SSO create site url for a particular user and domain.

```
>>> yola.get_sso_create_site_url('user_id', 'example.com')
```

**get\_sso\_open\_site\_url** (*user\_id*, *site\_id=None*, *locale=None*)

Get SSO open site url for a particular user.

```
>>> yola.get_sso_open_site_url('user_id')
```

**get\_user** (*user\_id*)

Get details for a particular user.

```
>>> user = yola.get_user('user_id')
>>> user['name']
'John'
```

**list\_users** (\*\**filters*)

Return paginated list of users.

```
>>> yola.list_users()
{
  'count': 999,
  'previous': None,
  'next': 'https://wl.qa.yola.net/pr/users/?page=2',
  'results': [
    {'name': 'John', 'surname': 'Smith', ...}
  ]
}
```

If there are no users, `results` will be an empty list. No exception will be raised.

You may pass pagination options and attribute filters as keyword arguments. See <https://wl.qa.yola.net/users/> for available parameters.

For example:

```
>>> yola.list_users(page=2, page_size=50, partner_id='WL_YOLA')
```

**update\_user** (*user\_id*, *\*\*attributes*)

Update a user.

```
>>> yola.update_user('user_id', name='New name')
```

## Sites

The following methods are mixed in to the `Yola` class to provide access to the Site resource:

**class** `yolapy.resources.site.SiteResourceMixin`

Methods for managing Site resources.

**change\_site\_domain** (*site\_id*, *new\_domain*)

Change site's domain.

```
>>> yola.change_site_domain('site_id', 'newdomain.com')
```

**change\_site\_owner** (*site\_id*, *new\_user\_id*)

Change site owner.

```
>>> yola.change_site_owner('site_id', 'new_user_id')
```

**delete\_site** (*site\_id*)

Delete a site.

```
>>> yola.get_site('site_id')['deleted_at']
None
>>> yola.delete_site('site_id')
>>> yola.get_site('site_id')['deleted_at']
'2015-08-27T10:00:07'
```

**disable\_site** (*site\_id*)

Disable a site.

A disabled site can be edited but can't be published.

```
>>> yola.disable_site('site_id')
```

**enable\_site** (*site\_id*)

Enable a site.

```
>>> yola.enable_site('site_id')
```

**get\_site** (*site\_id*)

Return details for a particular site.

```
>>> site = yola.get_site('site_id')
>>> site['name']
'My Site'
```

**list\_sites** (*\*\*options*)

Return paginated list of sites.

```
>>> yola.list_sites()
{
  'count': 999,
```

```
'previous': None,
'next': 'https://wl.qa.yola.net/pr/sites/?page=2',
'results': [{...}, {...}, ...]
}
```

You can pass `page_size` and `page` as keyword arguments:

```
>>> yola.list_sites(page_size=50, page=2)
```

You can also pass filters and ordering options as keyword arguments. See <https://wl.qa.yola.net/sites/> for available options.

**undelele\_site** (*site\_id*)

Un-delete a site.

```
>>> yola.get_site('site_id')['deleted_at']
'2015-08-27T10:00:07'
>>> yola.undelele_site('site_id')
>>> yola.get_site('site_id')['deleted_at']
None
```

## Subscriptions

The following methods are mixed in to the `Yola` class to provide access to the Subscription resource:

**class** `yolapy.resources.subscription.SubscriptionResourceMixin`

Methods for managing Subscription resources.

**activate\_trial\_subscription** (*subscription\_id*)

Convert trial subscription to active.

```
>>> yola.activate_trial_subscription('subscription_id')
```

**cancel\_subscription** (*subscription\_id, reason*)

Cancel active subscription.

```
>>> yola.cancel_subscription('subscription_id', 'some reason')
```

**change\_subscription\_type** (*subscription\_id, new\_type*)

Change subscription type.

```
>>> yola.change_subscription_type('subscription_id', 'new_type')
```

See <https://wl.qa.yola.net/subscriptions/> for available types.

**create\_subscription** (*subscription\_type, user\_id, properties*)

Create a new subscription.

```
>>> subscription_type = 'wl_basic'
>>> user_id = 'abcdef0123456789abcdef0123456789'
>>> properties = {...}
>>> yola.create_subscription(subscription_type, user_id, properties)
```

See <https://wl.qa.yola.net/subscriptions/> for available types and properties.

**get\_subscription** (*subscription\_id*)  
Return details for a particular subscription.

```
>>> subscription = yola.get_subscription('subscription_id')
>>> subscription['name']
'My Subscription'
```

**list\_subscriptions** (\*\*options)  
Return paginated list of subscriptions.

```
>>> yola.list_subscriptions()
{
  'count': 999,
  'previous': None,
  'next': 'https://wl.qa.yola.net/pr/subscriptions/?page=2',
  'results': [{...}, {...}, ...]
}
```

You can pass `page_size` and `page` as keyword arguments:

```
>>> yola.list_subscriptions(page_size=50, page=2)
```

You can also pass filters and ordering options as keyword arguments. See <https://wl.qa.yola.net/subscriptions/> for available options.

**reactivate\_subscription** (*subscription\_id*, *reason*)  
Re-activate a cancelled subscription.

```
>>> yola.reactivate_subscription('subscription_id', 'some reason')
```

## Campaigns

The following methods are mixed in to the `Yola` class to provide access to the Campaign resource:

**class** `yolapy.resources.campaign.CampaignResourceMixin`  
Methods for managing Campaign resources.

**delete\_campaign** (*site\_id*, *campaign\_id*)  
Delete a campaign.

**get\_campaign** (*site\_id*, *campaign\_id*)  
Return details for a particular campaign.

```
>>> campaign = yola.get_campaign('site_id', 'campaign_id')
>>> campaign['name']
'My Campaign'
```

**list\_campaigns** (*site\_id*, \*\*options)  
Return list of campaigns for a site.

## CHAPTER 2

---

### Indices and tables

---

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

`yolapy` provides a `Yola` service class for interacting directly with the Yola API. Future versions will provide higher-level abstractions.



**y**

yolapy.resources.campaign, 6  
yolapy.resources.partner, 2  
yolapy.resources.site, 4  
yolapy.resources.subscription, 5  
yolapy.resources.user, 2



## Symbols

`__init__()` (yolapy.services.Yola method), 1

## A

`activate_trial_subscription()`  
(yolapy.resources.subscription.SubscriptionResourceMixin  
method), 5

## C

`CampaignResourceMixin` (class in  
yolapy.resources.campaign), 6

`cancel_subscription()` (yolapy.resources.subscription.SubscriptionResourceMixin  
method), 5

`change_site_domain()` (yolapy.resources.site.SiteResourceMixin  
method), 4

`change_site_owner()` (yolapy.resources.site.SiteResourceMixin  
method), 4

`change_subscription_type()`  
(yolapy.resources.subscription.SubscriptionResourceMixin  
method), 5

`create_partner()` (yolapy.resources.partner.PartnerResourceMixin  
method), 2

`create_subscription()` (yolapy.resources.subscription.SubscriptionResourceMixin  
method), 5

`create_user()` (yolapy.resources.user.UserResourceMixin  
method), 3

## D

`delete_campaign()` (yolapy.resources.campaign.CampaignResourceMixin  
method), 6

`delete_partner()` (yolapy.resources.partner.PartnerResourceMixin  
method), 2

`delete_site()` (yolapy.resources.site.SiteResourceMixin  
method), 4

`delete_user()` (yolapy.resources.user.UserResourceMixin  
method), 3

`disable_site()` (yolapy.resources.site.SiteResourceMixin  
method), 4

## E

`enable_site()` (yolapy.resources.site.SiteResourceMixin  
method), 4

## G

`get_campaign()` (yolapy.resources.campaign.CampaignResourceMixin  
method), 6

`get_partner()` (yolapy.resources.partner.PartnerResourceMixin  
method), 2

`get_site()` (yolapy.resources.site.SiteResourceMixin  
method), 4

`get_sso_create_site_url()`  
(yolapy.resources.user.UserResourceMixin  
method), 3

`get_sso_open_site_url()` (yolapy.resources.user.UserResourceMixin  
method), 3

`get_subscription()` (yolapy.resources.subscription.SubscriptionResourceMixin  
method), 5

`get_user()` (yolapy.resources.user.UserResourceMixin  
method), 3

`list_campaigns()` (yolapy.resources.campaign.CampaignResourceMixin  
method), 6

`list_partners()` (yolapy.resources.partner.PartnerResourceMixin  
method), 2

`list_sites()` (yolapy.resources.site.SiteResourceMixin  
method), 4

`list_subscriptions()` (yolapy.resources.subscription.SubscriptionResourceMixin  
method), 6

`list_users()` (yolapy.resources.user.UserResourceMixin  
method), 3

## P

`PartnerResourceMixin` (class in  
yolapy.resources.partner), 2

## R

`reactivate_subscription()` (yolapy.resources.subscription.SubscriptionResourceMixin  
method), 6

## S

SiteResourceMixin (class in yolapy.resources.site), 4

SubscriptionResourceMixin (class in yolapy.resources.subscription), 5

## U

undelele\_site() (yolapy.resources.site.SiteResourceMixin method), 5

update\_partner() (yolapy.resources.partner.PartnerResourceMixin method), 2

update\_user() (yolapy.resources.user.UserResourceMixin method), 3

UserResourceMixin (class in yolapy.resources.user), 2

## Y

Yola (class in yolapy.services), 1

yolapy.resources.campaign (module), 6

yolapy.resources.partner (module), 2

yolapy.resources.site (module), 4

yolapy.resources.subscription (module), 5

yolapy.resources.user (module), 2