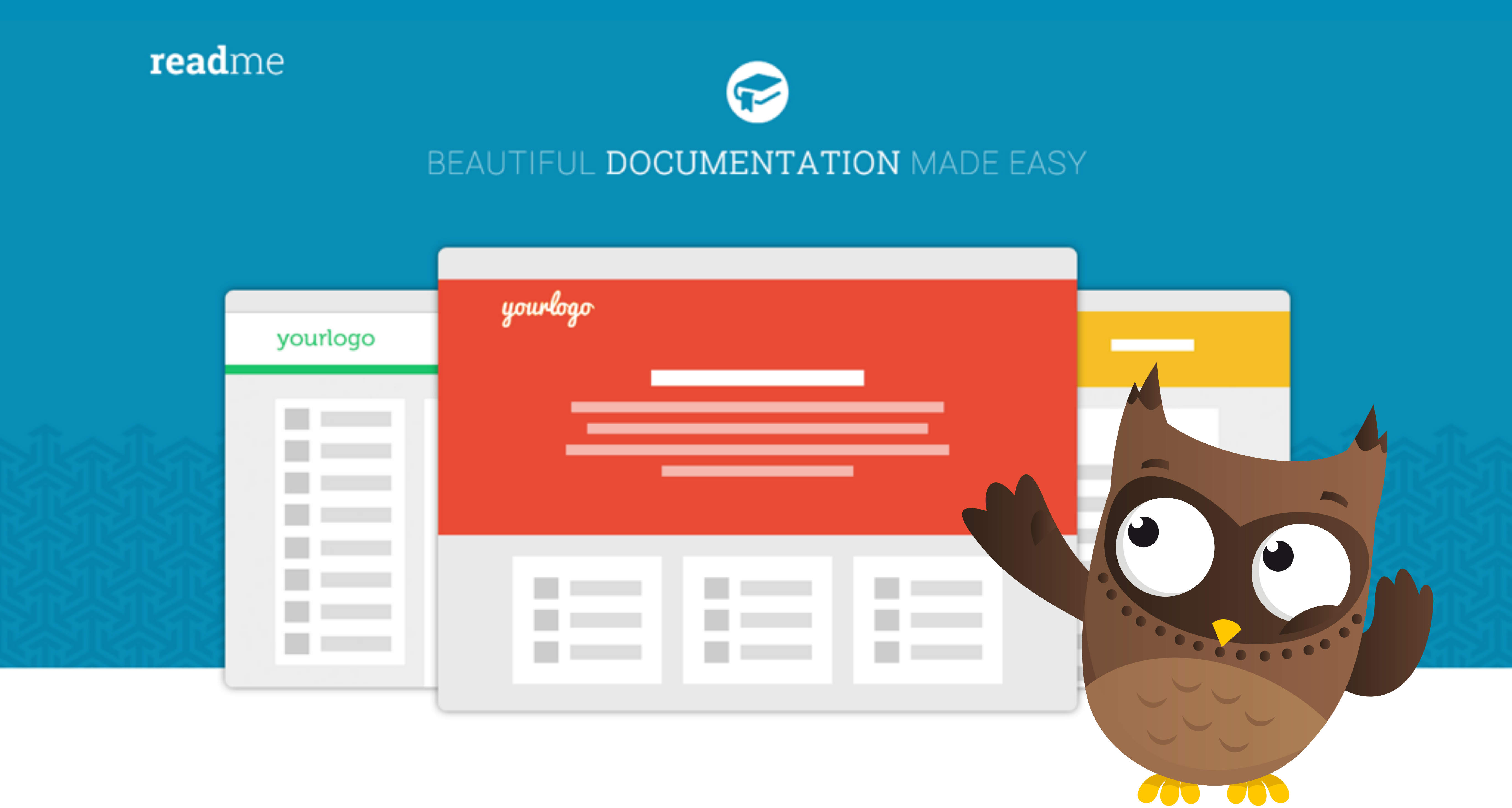
Dynamic Docs

making documentation fit the user



Gregory Koberger

Founder, ReadMe.io
Designer + Developer

@gkoberger

ReadMe.io

Beautiful Documentation
Made Easy

http://readme.io

Owlbert

Our mascot, who will
help out today

@readmeio

A Developers Approach To Docs

Three Levels

Passive

Has no knowledge of the user

Reactive

Responds to an action from the user

Proactive

Pushes content to the user

Level 1: Passive

Passive

Could be contained inside a book

Level 1: Passive

Topical Guides
Why you would do something

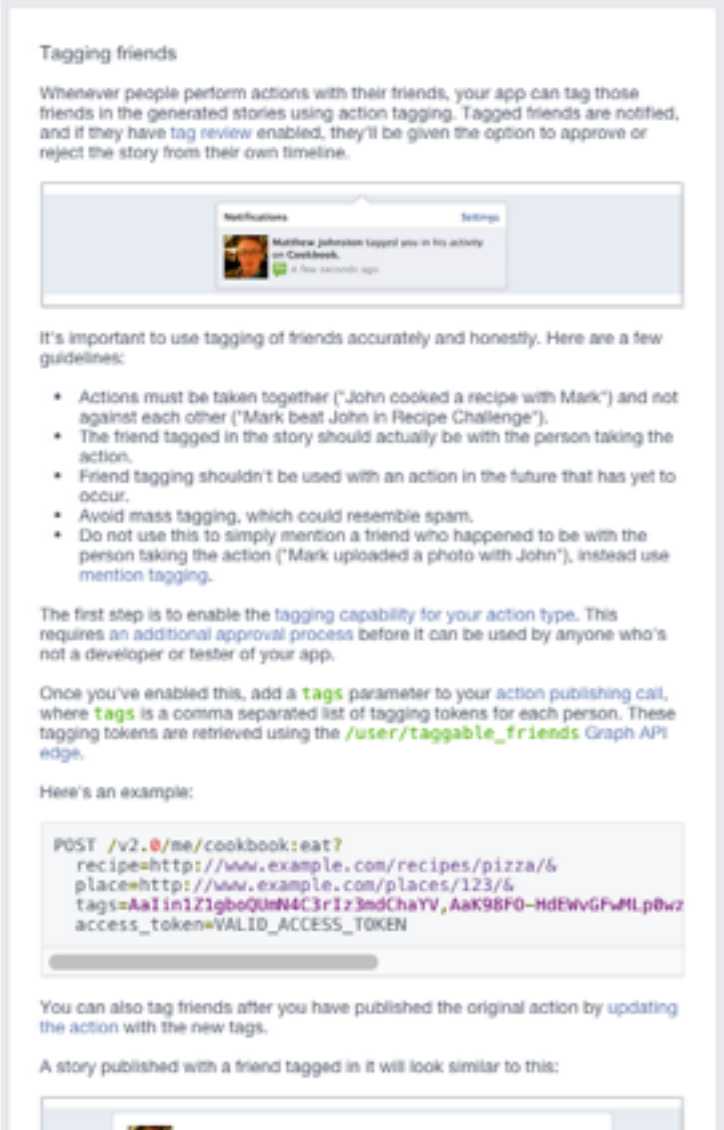
Tutorials
How you would do something

Reference
Deep-dive into the material

Community
Q&A, Support, Collaborative Editing

Level 1.5: Passive(ish)

How to send app invitation to Facebook friends?



Tagging friends

Whenever people perform actions with their friends, your app can tag those friends in the generated stories using action tagging. Tagged friends are notified, and if they have tag review enabled, they'll be given the option to approve or reject the story from their own timeline.

Notifications Settings

Matthew Johnston tagged you in his activity on Cookbook.

A few seconds ago

It's important to use tagging of friends accurately and honestly. Here are a few guidelines:

- Actions must be taken together ("John cooked a recipe with Mark") and not against each other ("Mark beat John in Recipe Challenge").
- The friend tagged in the story should actually be with the person taking the action.
- Friend tagging shouldn't be used with an action in the future that has yet to occur.
- Avoid mass tagging, which could resemble spam.
- Do not use this to simply mention a friend who happened to be with the person taking the action ("Mark uploaded a photo with John"), instead use mention tagging.

The first step is to enable the tagging capability for your action type. This requires an additional approval process before it can be used by anyone who's not a developer or tester of your app.

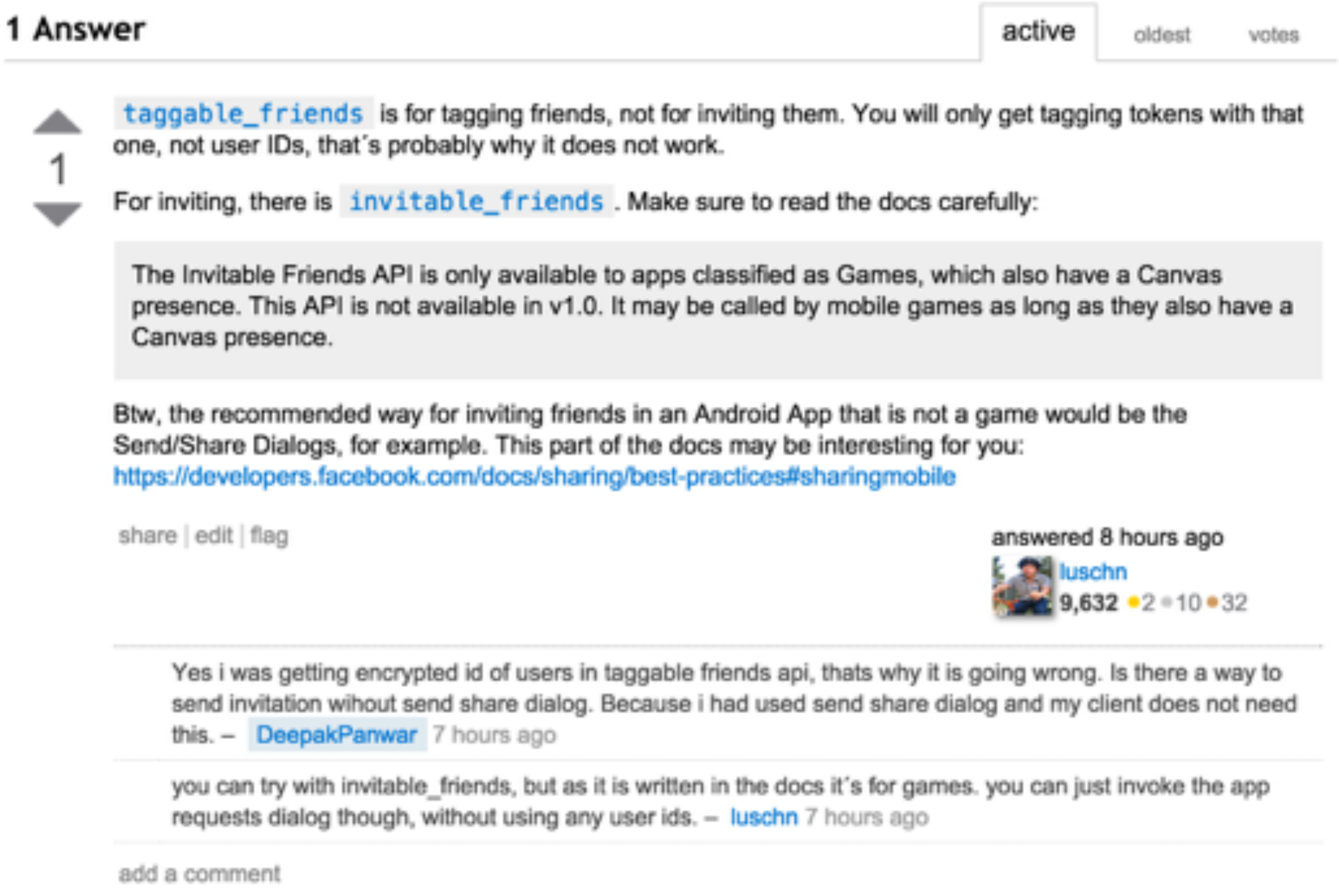
Once you've enabled this, add a tags parameter to your action publishing call, where tags is a comma separated list of tagging tokens for each person. These tagging tokens are retrieved using the /user/taggable_friends Graph API edge.

Here's an example:

```
POST /v2.0/me/cookbook:eat?
  recipe=http://www.example.com/recipes/pizza/&
  place=http://www.example.com/places/123/&
  tags=AaIin1Z1gboQUmN4C3rIz3mdChaYV,AaK9BFO-HdEWvGFwMLp0wz
  access_token=VALID_ACCESS_TOKEN
```

You can also tag friends after you have published the original action by updating the action with the new tags.

A story published with a friend tagged in it will look similar to this:



1 Answer

active oldest votes

1

`taggable_friends` is for tagging friends, not for inviting them. You will only get tagging tokens with that one, not user IDs, that´s probably why it does not work.

For inviting, there is `invitable_friends`. Make sure to read the docs carefully:

> The Invitable Friends API is only available to apps classified as Games, which also have a Canvas presence. This API is not available in v1.0. It may be called by mobile games as long as they also have a Canvas presence.

Btw, the recommended way for inviting friends in an Android App that is not a game would be the Send/Share Dialogs, for example. This part of the docs may be interesting for you: https://developers.facebook.com/docs/sharing/best-practices#sharingmobile

share | edit | flag

answered 8 hours ago
luschn
9,632 •2 •10 •32

Yes i was getting encrypted id of users in taggable friends api, thats why it is going wrong. Is there a way to send invitation wihout send share dialog. Because i had used send share dialog and my client does not need this. – DeepakPanwar 7 hours ago

you can try with invitable_friends, but as it is written in the docs it´s for games. you can just invoke the app requests dialog though, without using any user ids. – luschn 7 hours ago

add a comment

Why Do Docs Need To Be Dynamic?

REDUCING THE COGNITIVE LOAD

8+

to be understood, implemented and **maintained**

Why Do Docs Need To Be Dynamic?

The (current) bane of my existence:
Authentication

Level 2: Reactive

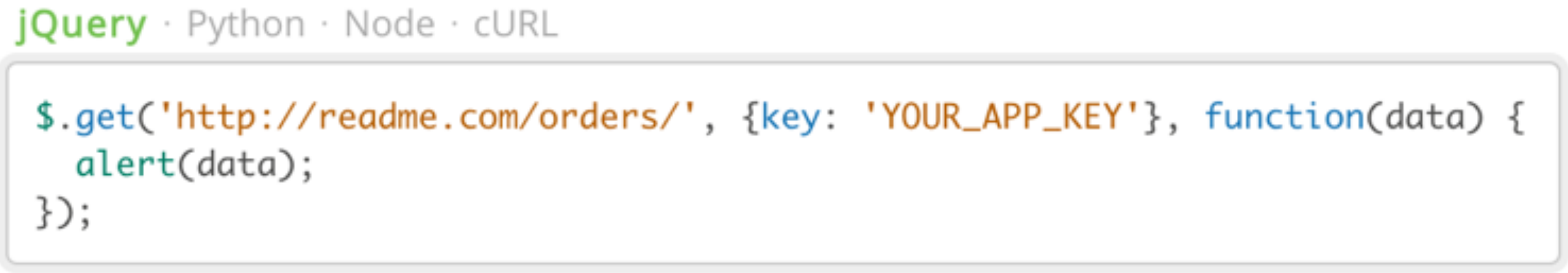
The Premise

Use what you know about your users to reduce their cognitive workload

Level 2: Reactive

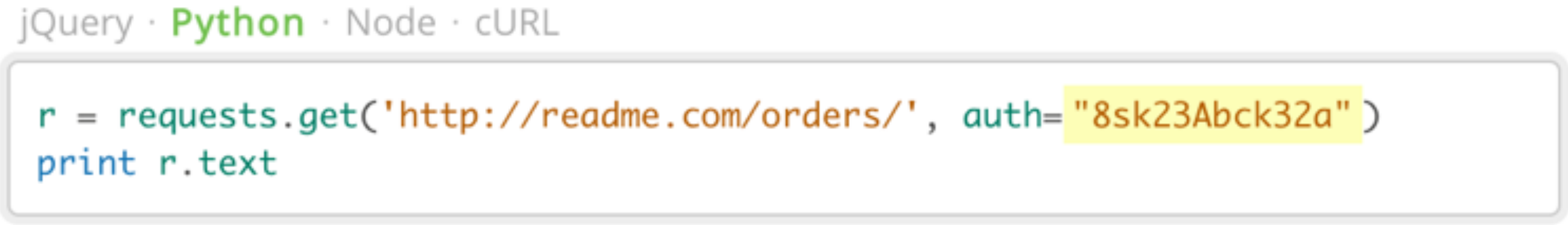
API Keys

before



```
$.get('http://readme.com/orders/', {key: 'YOUR_APP_KEY'}, function(data) {
  alert(data);
});
```

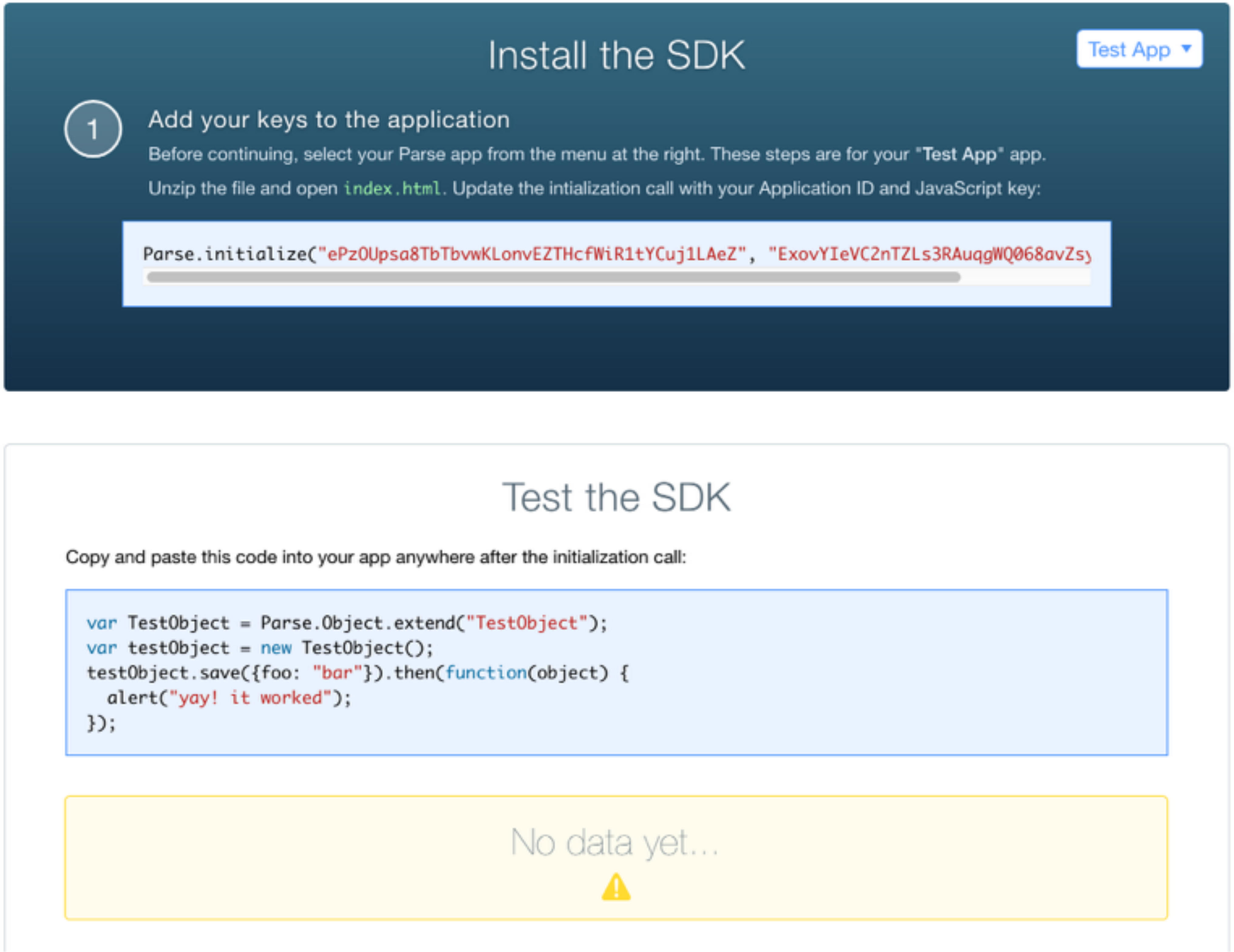
after



```
r = requests.get('http://readme.com/orders/', auth="8sk23Abck32a")
print r.text
```

Level 2: Reactive

Onboarding



Install the SDK

1 Add your keys to the application

Before continuing, select your Parse app from the menu at the right. These steps are for your "**Test App**" app.

Unzip the file and open `index.html`. Update the intialization call with your Application ID and JavaScript key:

```
Parse.initialize("ePzOUpsa8TbTbvwKLonvEZTHcfWiR1tYCuj1LAeZ", "ExovYIeVC2nTZLs3RAuqgWQ068avZsy
```

Test the SDK

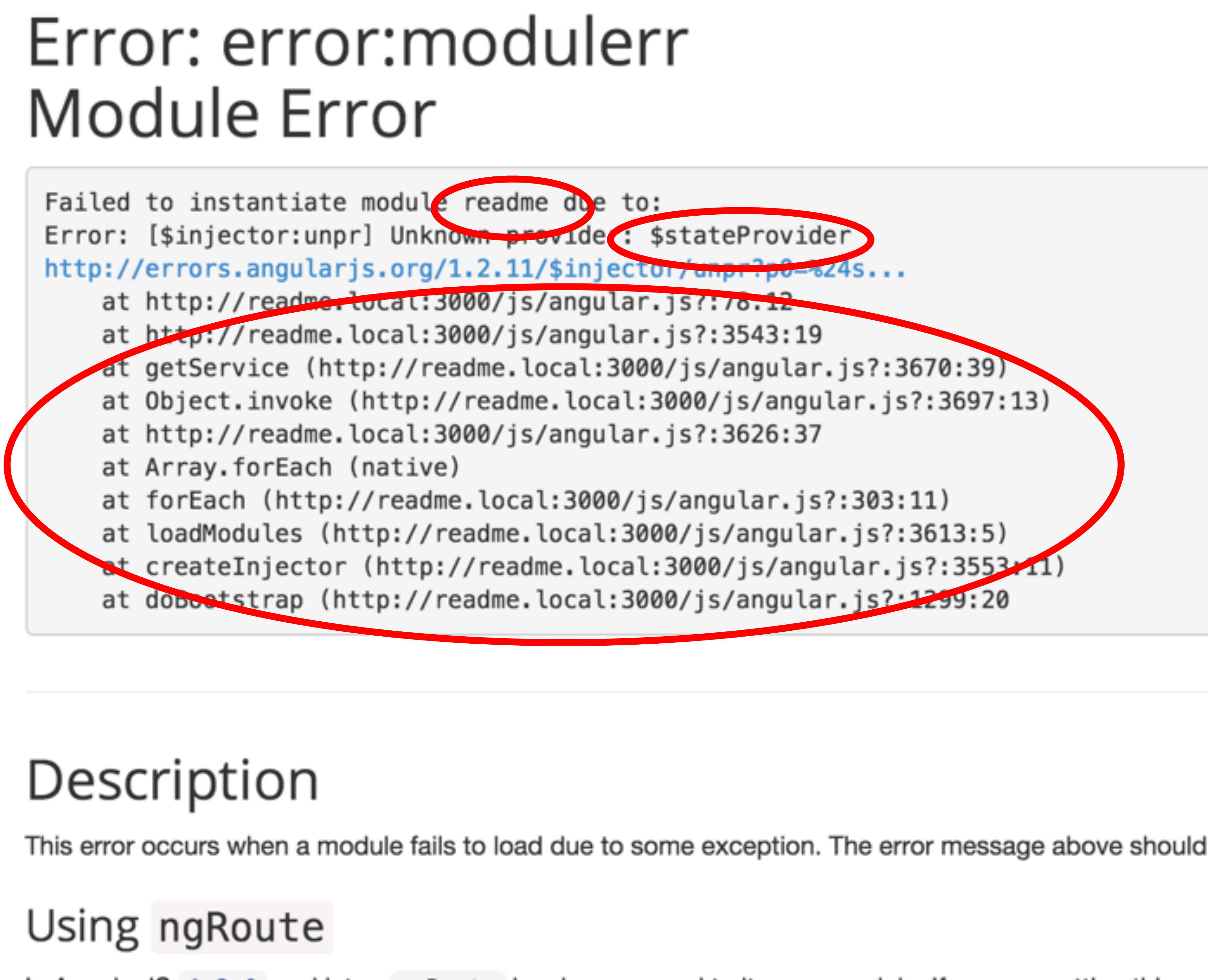
Copy and paste this code into your app anywhere after the initialization call:

```
var TestObject = Parse.Object.extend("TestObject");
var testObject = new TestObject();
testObject.save({foo: "bar"}).then(function(object) {
  alert("yay! it worked");
});
```

No data yet... ⚠

Level 2: Reactive

Customized Errors



Error: error:modulerr
Module Error

```
Failed to instantiate module readme due to:
Error: [$injector:unpr] Unknown provider: $stateProvider
http://errors.angularjs.org/1.2.11/$injector/unpr?p0=%24s...
    at http://readme.local:3000/js/angular.js?:78:12
    at http://readme.local:3000/js/angular.js?:3543:19
    at getService (http://readme.local:3000/js/angular.js?:3670:39)
    at Object.invoke (http://readme.local:3000/js/angular.js?:3697:13)
    at http://readme.local:3000/js/angular.js?:3626:37
    at Array.forEach (native)
    at forEach (http://readme.local:3000/js/angular.js?:303:11)
    at loadModules (http://readme.local:3000/js/angular.js?:3613:5)
    at createInjector (http://readme.local:3000/js/angular.js?:3553:11)
    at doBootstrap (http://readme.local:3000/js/angular.js?:1299:20
```

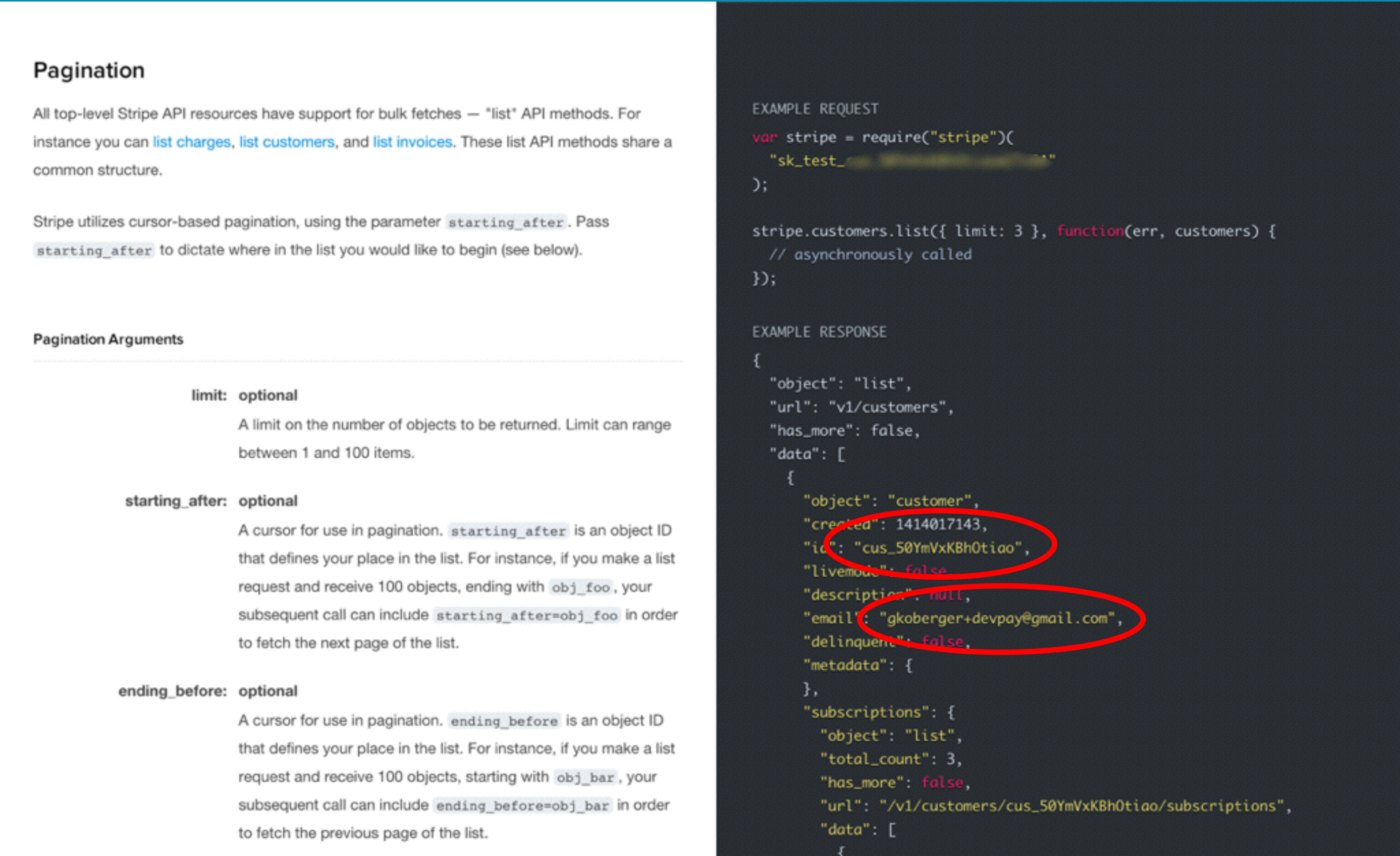
Description

This error occurs when a module fails to load due to some exception. The error message above should provide ad

Using ngRoute

Level 2: Reactive

Actual Requests



Pagination

All top-level Stripe API resources have support for bulk fetches — "list" API methods. For instance you can list charges, list customers, and list invoices. These list API methods share a common structure.

Stripe utilizes cursor-based pagination, using the parameter `starting_after`. Pass `starting_after` to dictate where in the list you would like to begin (see below).

Pagination Arguments

limit: optional

A limit on the number of objects to be returned. Limit can range between 1 and 100 items.

starting_after: optional

A cursor for use in pagination. `starting_after` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, ending with `obj_foo`, your subsequent call can include `starting_after=obj_foo` in order to fetch the next page of the list.

ending_before: optional

A cursor for use in pagination. `ending_before` is an object ID that defines your place in the list. For instance, if you make a list request and receive 100 objects, starting with `obj_bar`, your subsequent call can include `ending_before=obj_bar` in order to fetch the previous page of the list.

EXAMPLE REQUEST

```
var stripe = require("stripe")(
  "sk_test_[redacted]"
);

stripe.customers.list({ limit: 3 }, function(err, customers) {
  // asynchronously called
});
```

EXAMPLE RESPONSE

```
{
  "object": "list",
  "url": "v1/customers",
  "has_more": false,
  "data": [
    {
      "object": "customer",
      "created": 1414017143,
      "id": "cus_50YmVxKBhOtiao",
      "livemode": false,
      "description": null,
      "email": "gkoberger+devpay@gmail.com",
      "delinquent": false,
      "metadata": {
      },
      "subscriptions": {
        "object": "list",
        "total_count": 3,
        "has_more": false,
        "url": "/v1/customers/cus_50YmVxKBhOtiao/subscriptions",
        "data": [
          {
```

Level 3: Proactive

The Premise

Use what you know about your users to reduce their cognitive workload

…..**EVEN MORE!**

Level 3: Proactive

Reaching Out Before There's A Problem

Owlexander

is using…
/users
/orders

Owlivia

is using…
/users
/testing

Level 3: Proactive

Reaching Out Before There's A Problem

Owlexander

is using…

/users 503!

/orders 404!

1337!

Questions?