



# Effortless real time apps in Django

@aaronbassett – rawtech.io

10



rawtech.io



**talk weird**



**@aaronbassett**



Ask Jeeves



The Ask Jeeves homepage features a large search bar at the top with the placeholder "What's on your mind?" Below the search bar are several circular icons with different symbols, such as a graduation cap, a hand, and a gear. To the right of the search bar is a section titled "Ask Jeeves" with a logo and the text "Ask.com". Further down the page are sections for "Browse by Subject" and "Ask Other". A large "ASK" watermark is overlaid across the entire page.



Ask Jeeves  
VIDS

Ask Jeeves for KIDS

Everyday quotations, homework help, and fun!

Do You Know the Answer?  
Where can I earn the ages of  
stars in space?  
Join our Q&A Boards!

What Are People Asking?  
Current questions from Ask

Tour the Web  
Jeeves' To...  
through the...  
click at a time

```
1 (function poll() {
2     new Ajax.Request('/api/' , {
3         method: 'get',
4         onSuccess: function() { ... } ,
5         onFailure: function() { ... }
6     ) ;
7
8     setTimeout(poll, 1000);
9 } ());
```

```
1 (function poll() {  
2     new Ajax.Request('/api/' , {  
3         method: 'get',  
4         timeout: 60000,  
5         onSuccess: function() {  
6             // Do something  
7             poll();  
8         },  
9         onFailure: function() {  
10            // Do something else  
11            poll();  
12        }  
13    });  
14 })();
```

# HACKS

## UPON HACKS

UPON HACKS, UPON HACKS  
UPON HACKS, UPON HACKS, UPON HACKS  
UPON HACKS, UPON HACKS, UPON HACKS, UPON HACKS  
UPON HACKS, UPON HACKS, UPON HACKS, UPON HACKS, UPON HACKS, UPON HACKS  
UPON HACKS, UPON HACKS, UPON HACKS, UPON HACKS, UPON HACKS, UPON HACKS, UPON HACKS



“typical header sizes of 700-800 bytes is common”

*-Google “SPDY: An experimental protocol for a faster web”*



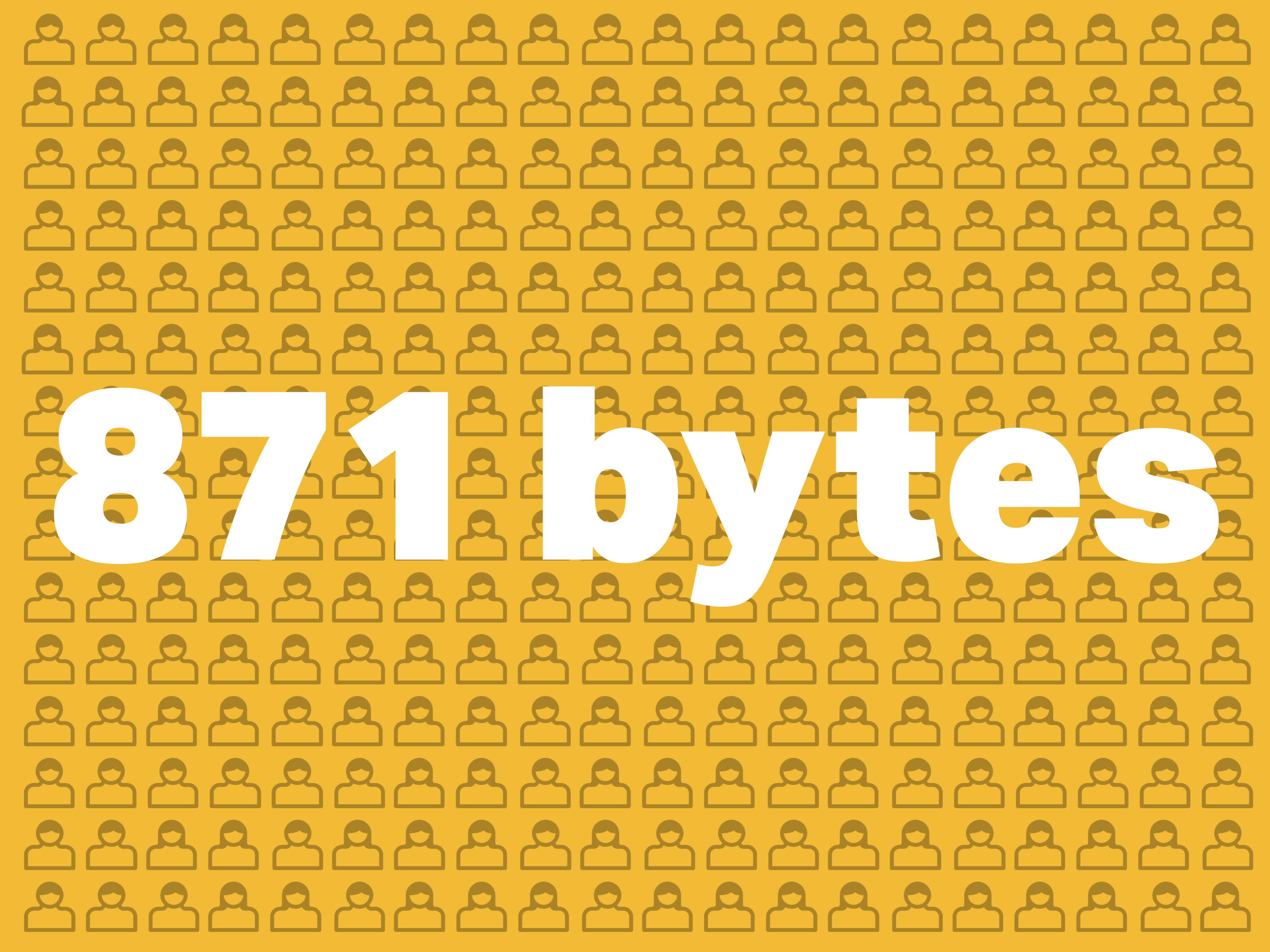
2bytes

0x00

UTF8 DATA

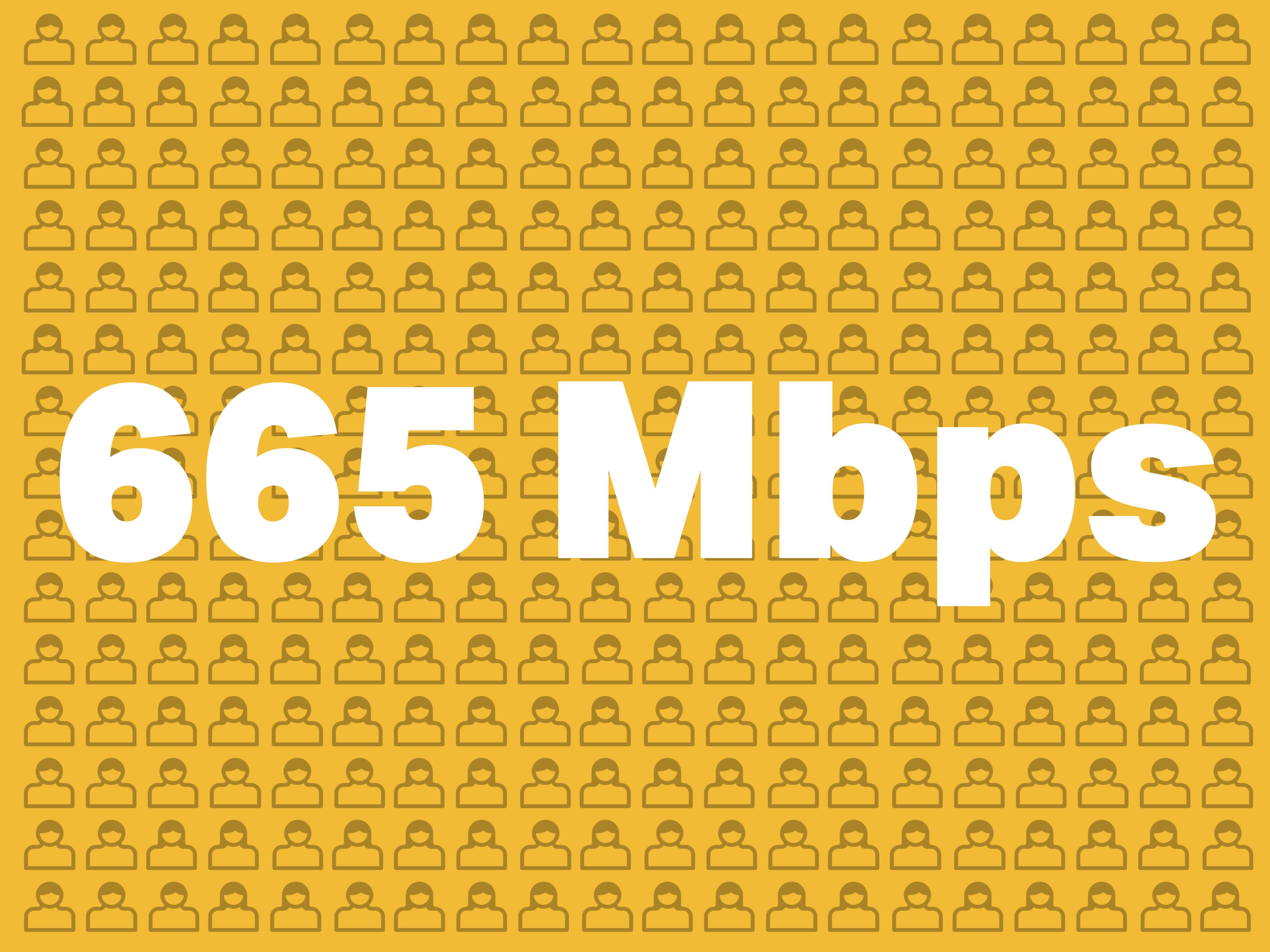
0xFF

100,000



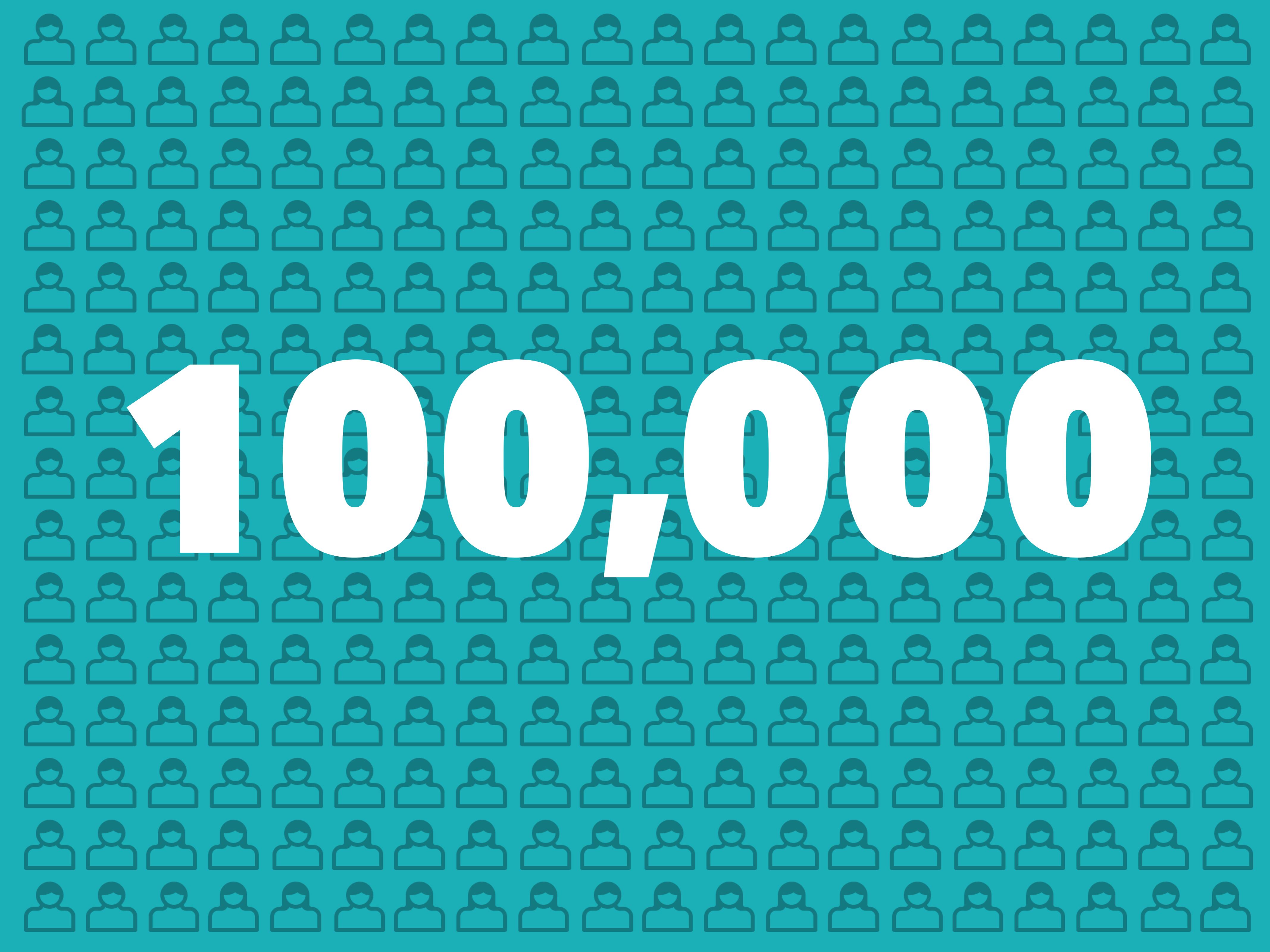
871 bytes

**696,800,000**

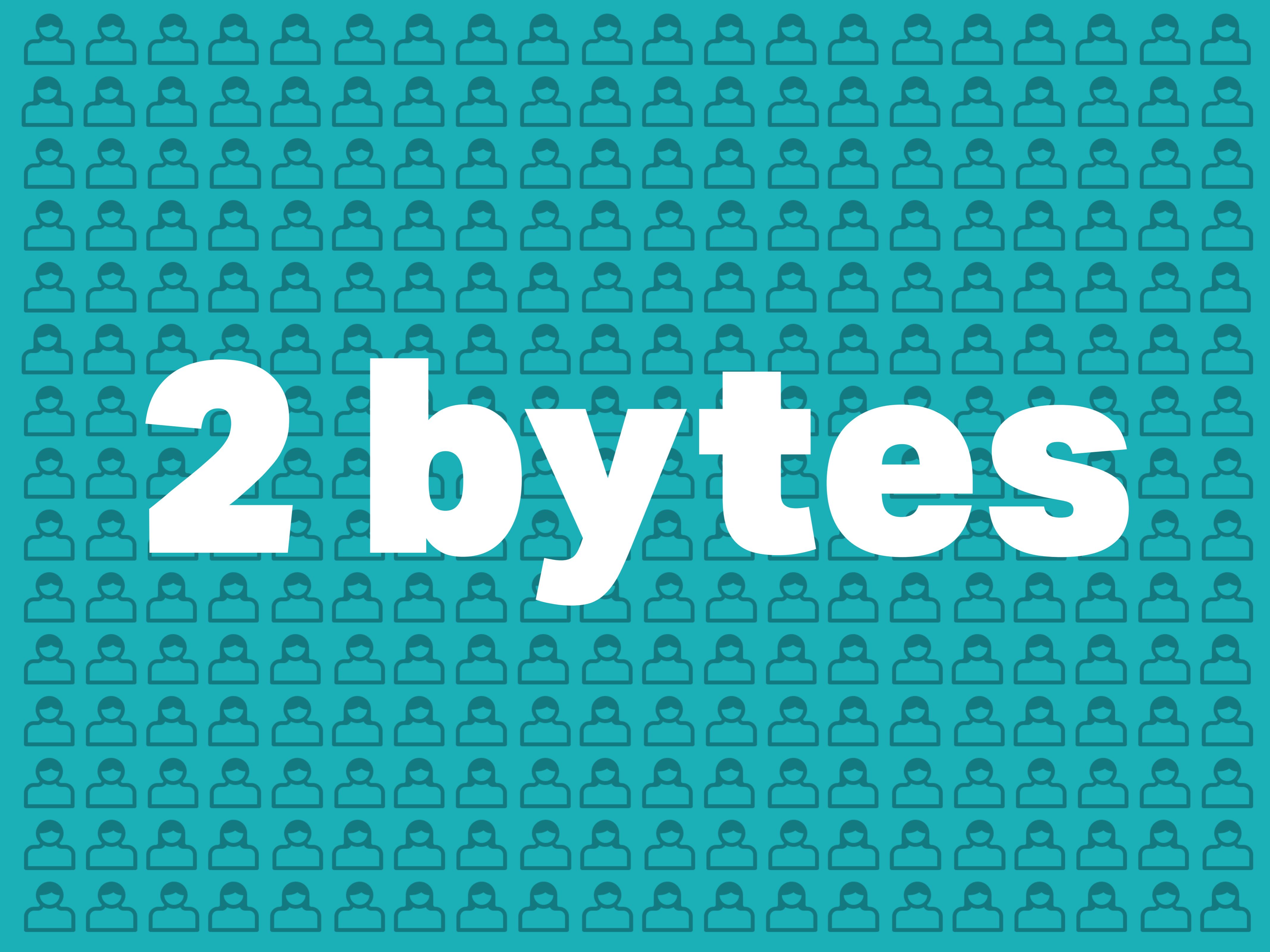


665 Milos

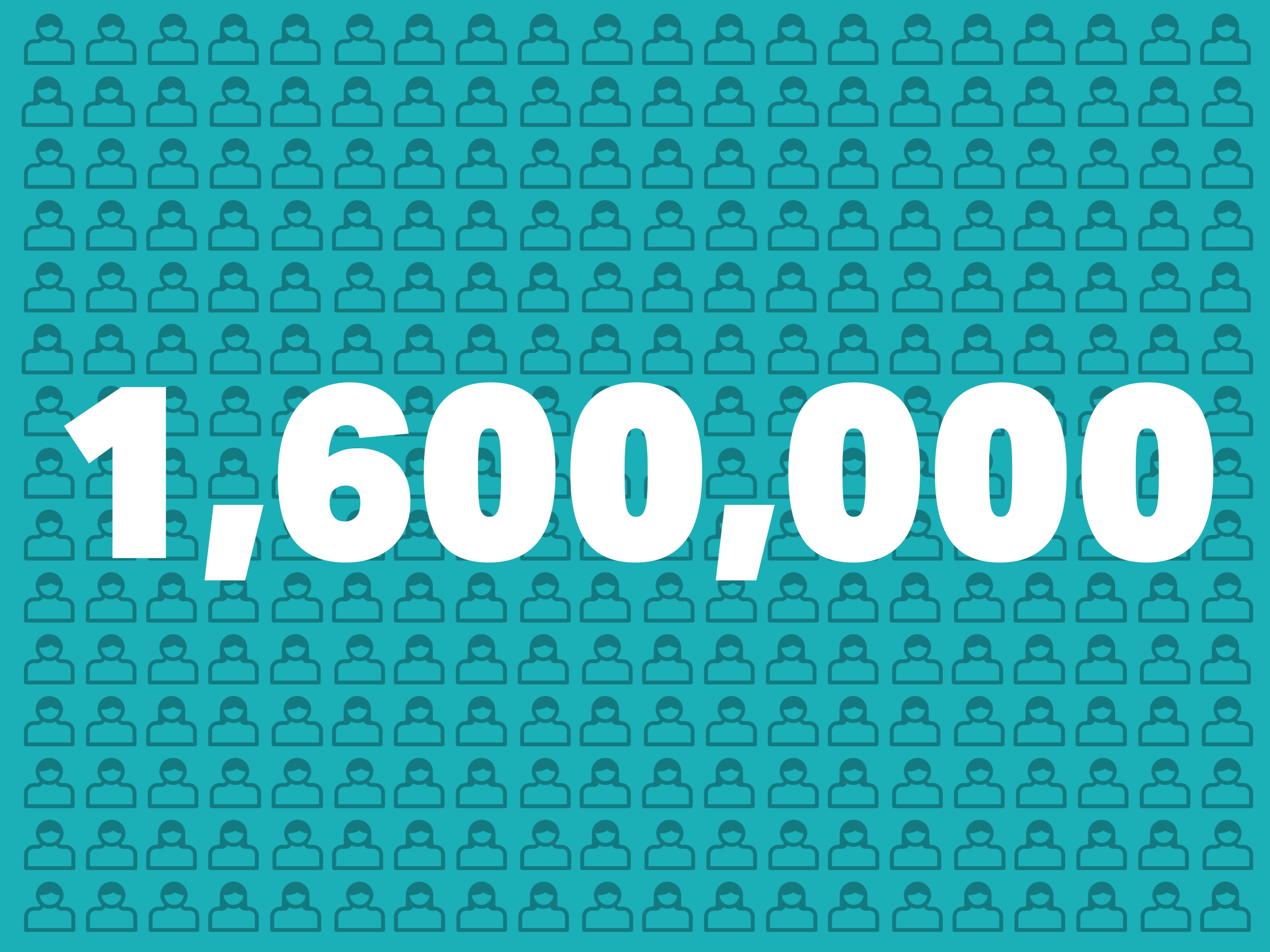
83M8



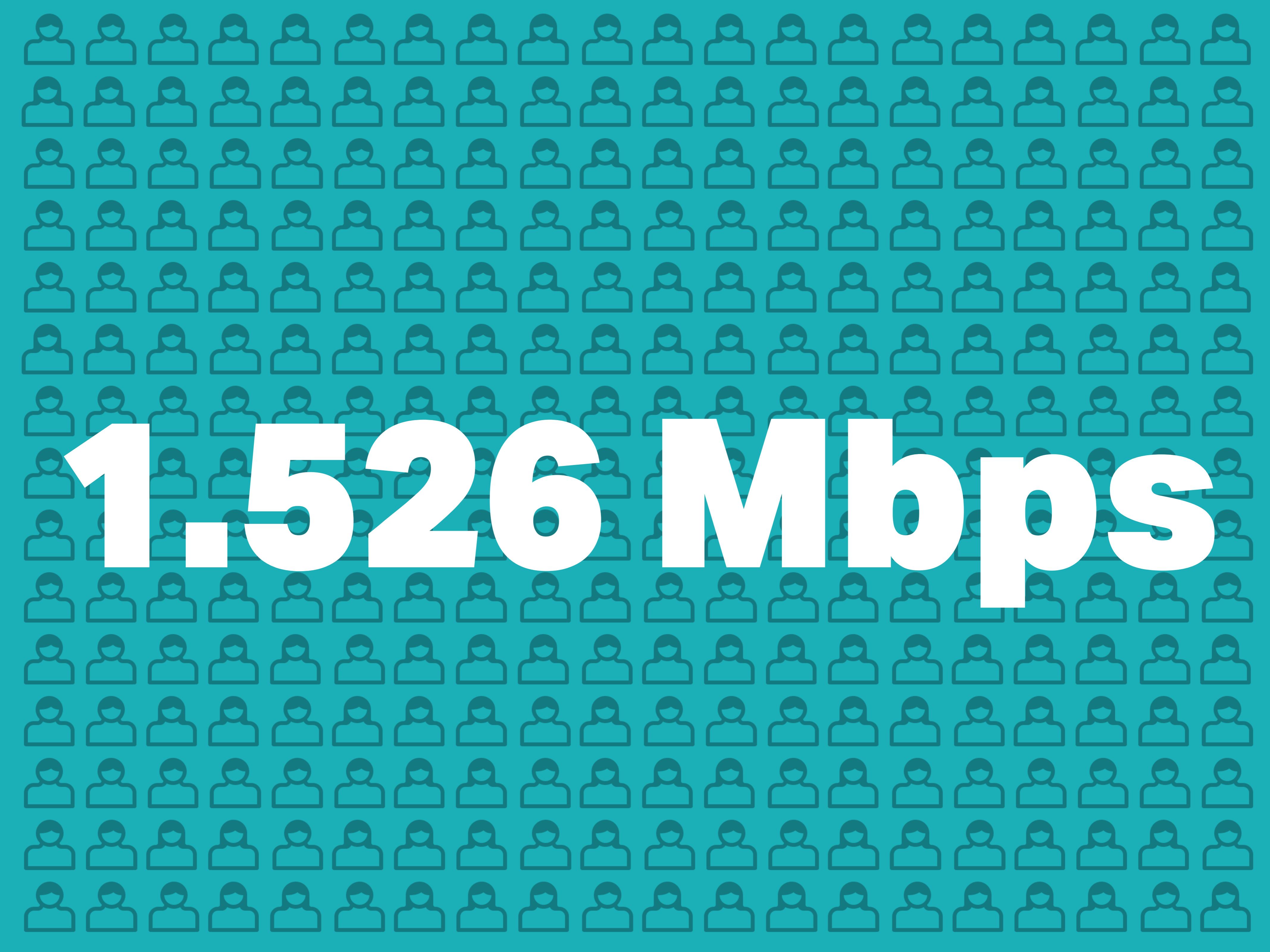
100 000



2bytes



1600,000



**1526 Mbps**

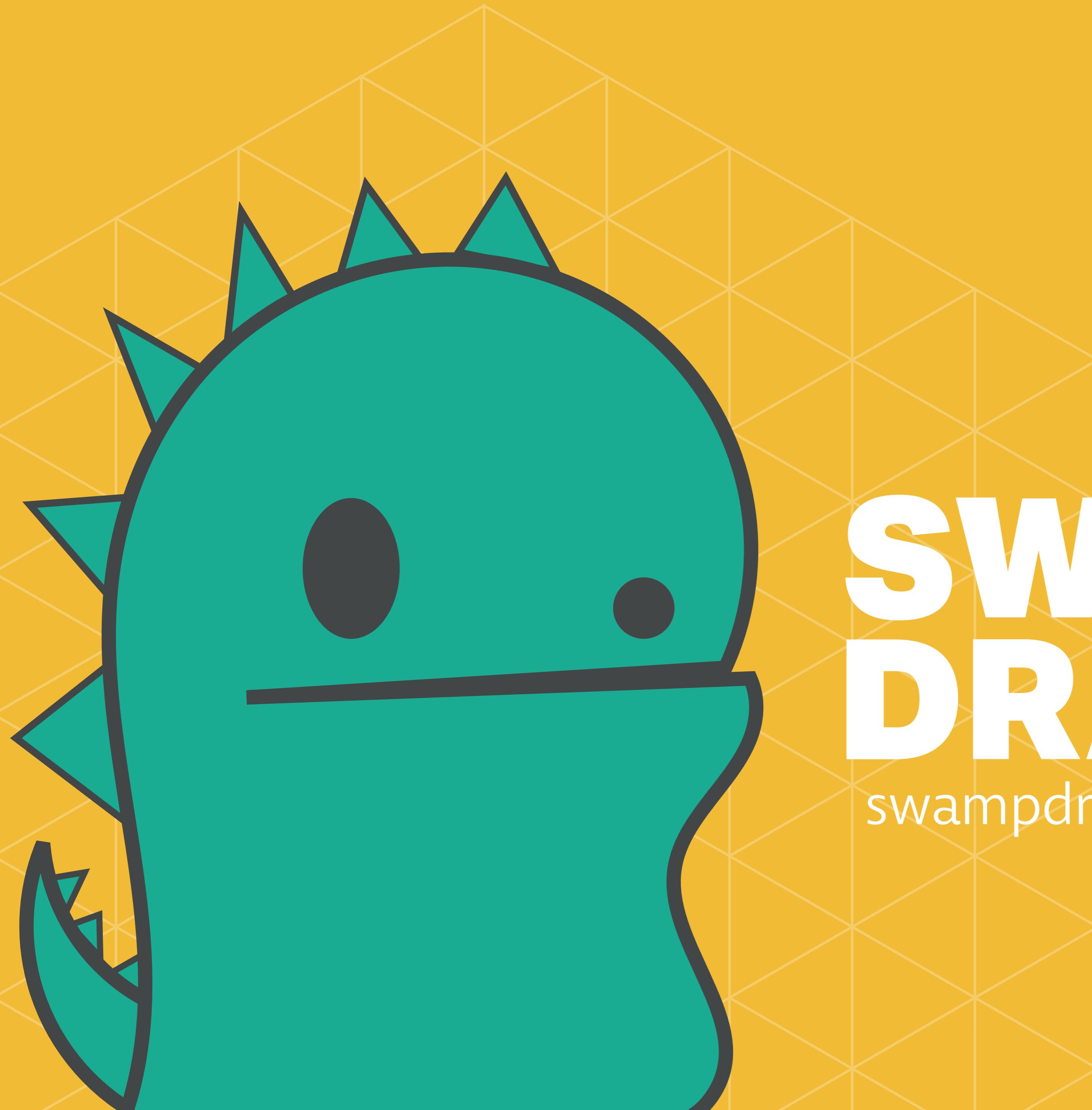
02MB

# POLLING



# WEB SOCKETS





# SWAMP DRAGON

[swampdragon.net](http://swampdragon.net)



**django**



**Tornado**



**redis**



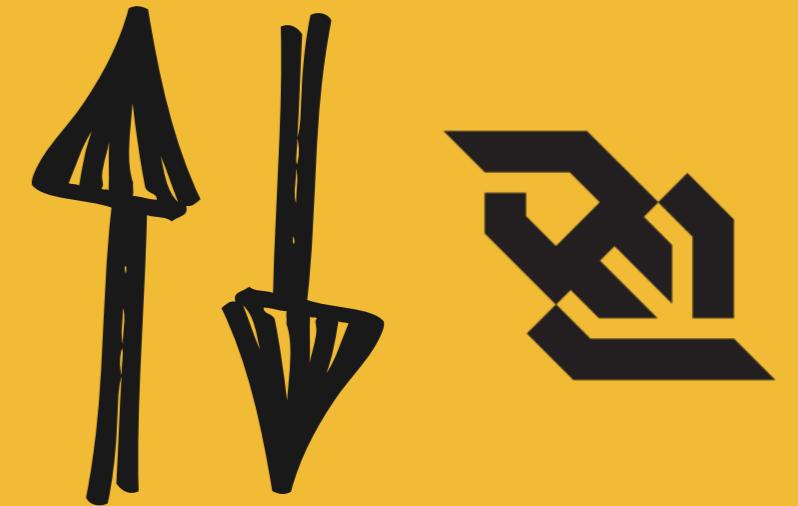
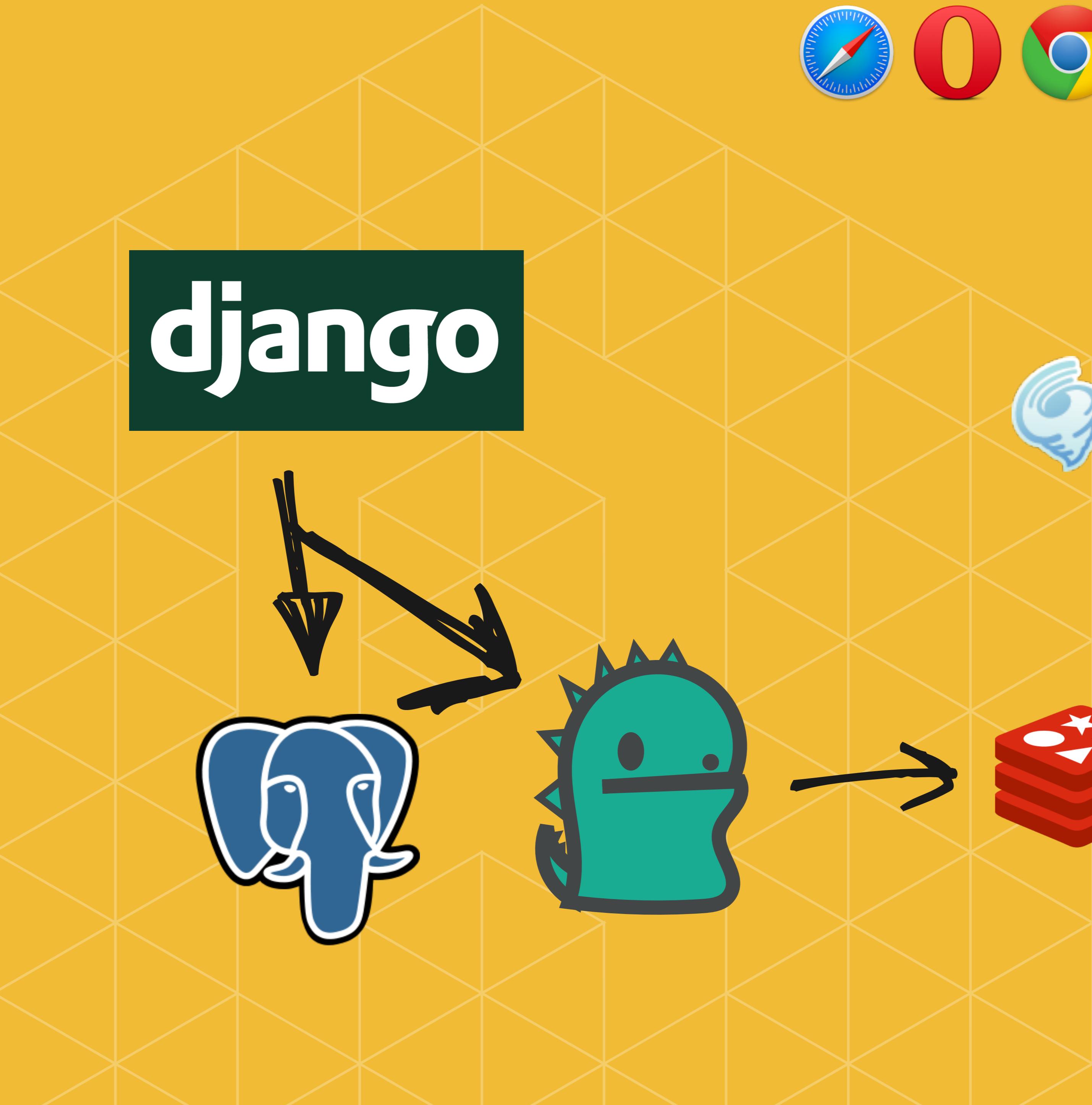
redis



# Tornado



**djang**o



**django**



**Tornado**



**redis**

**create YET another todo app**



**create YET another todo app**



**pip install swampdragon**



**create YET another todo app**



**pip install swampdragon**



**(apt-get | brew) install redis**





dragon  
admin

```
1 #!/usr/bin/env python
2 import os
3 import sys
4
5
6 from swampdragon.swampdragon_server import run_server
7
8 os.environ.setdefault("DJANGO_SETTINGS_MODULE", "<project>.settings")
9
10 host_port = sys.argv[1] if len(sys.argv) > 1 else None
11
12 run_server(host_port=host_port)
```

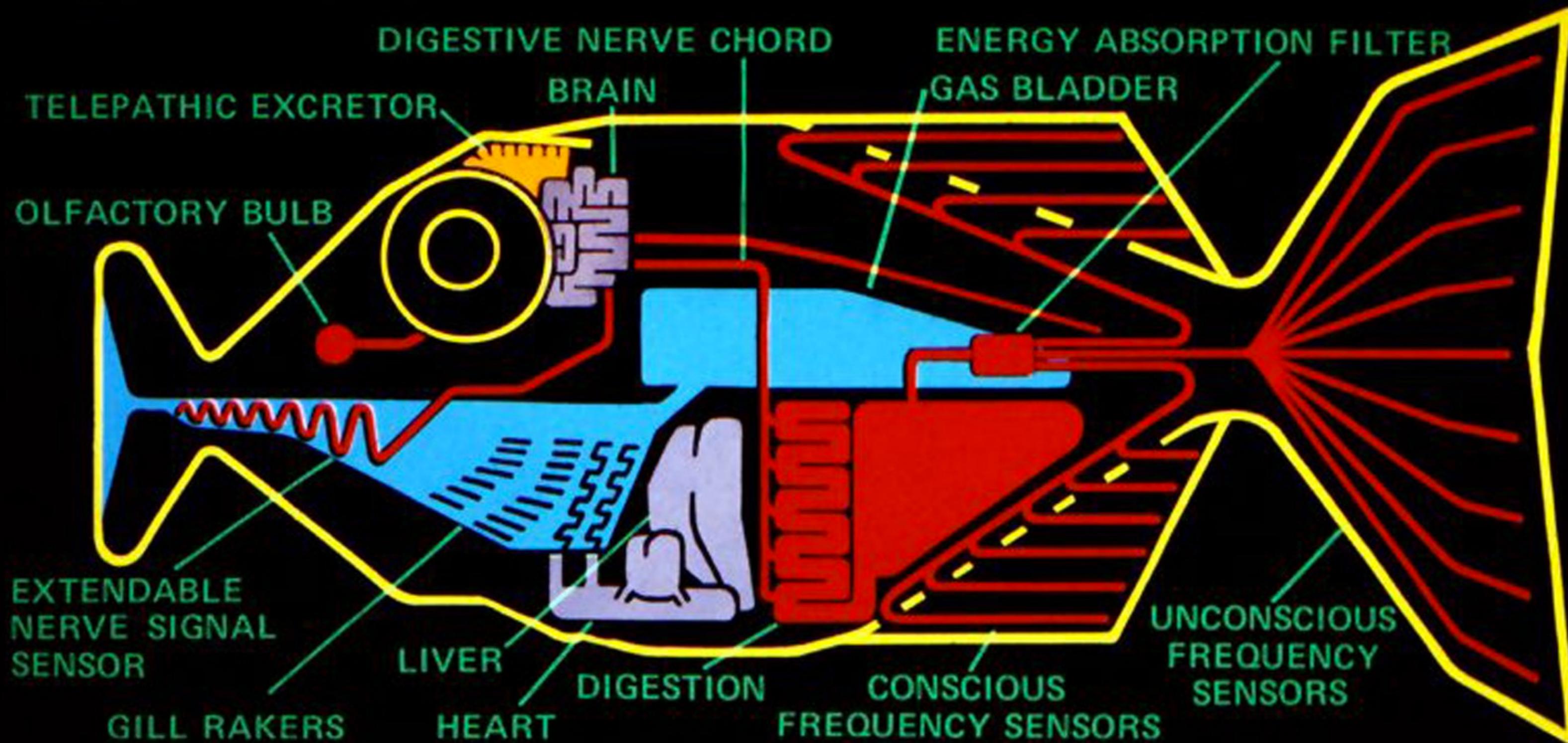
```
1 #!/usr/bin/env python
2 import os
3 import sys
4
5
6 from swampdragon.swampdragon_server import run_server
7
8 os.environ.setdefault("DJANGO_SETTINGS_MODULE", "<project>.settings")
9
10 host_port = sys.argv[1] if len(sys.argv) > 1 else None
11
12 run_server(host_port=host_port)
```

```
1 # -*- coding: utf-8 -*-
2
3 # Django
4 from django.db import models
5 # Swamp Dragon
6 from swampdragon.models import SelfPublishModel
7 from .serializers import TodoListSerializer, TodoItemSerializer
8
9
10 class TodoItem(SelfPublishModel, models.Model):
11     serializer_class = TodoItemSerializer
12
13     todo_list = models.ForeignKey(TodoList)
14     done = models.BooleanField(default=False)
15     text = models.CharField(max_length=100)
16
17     def __unicode__(self):
18         return u"{} ({})".format(
19             text=self.text,
20             status=(u"✓" if self.done else u"✗"))
21
```

```
1 # -*- coding: utf-8 -*-
2
3 # Django
4 from django.db import models
5 # Swamp Dragon
6 from swampdragon.models import SelfPublishModel
7 from .serializers import TodoListSerializer, TodoItemSerializer
8
9
10 class TodoItem(SelfPublishModel, models.Model):
11     serializer_class = TodoItemSerializer
12
13     todo_list = models.ForeignKey(TodoList)
14     done = models.BooleanField(default=False)
15     text = models.CharField(max_length=100)
16
17     def __unicode__(self):
18         return u"{} ({})".format(
19             text=self.text,
20             status=(u"✓" if self.done else u"✗"))
21
```

```
1 # -*- coding: utf-8 -*-
2
3 # Django
4 from django.db import models
5 # Swamp Dragon
6 from swampdragon.models import SelfPublishModel
7 from .serializers import TodoListSerializer, TodoItemSerializer
8
9
10 class TodoItem(SelfPublishModel, models.Model):
11     serializer_class = TodoItemSerializer
12
13     todo_list = models.ForeignKey(TodoList)
14     done = models.BooleanField(default=False)
15     text = models.CharField(max_length=100)
16
17     def __unicode__(self):
18         return u"{} ({})".format(
19             text=self.text,
20             status=(u"✓" if self.done else u"✗"))
21
```

# BABEL FISH



THE BABEL FISH IS SMALL, YELLOW, LEECHLIKE,  
AND PROBABLY THE ODDEST THING IN THE UNIVERSE.  
IT FEEDS ON BRAIN WAVE ENERGY, ABSORBING ALL

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon.serializers.model_serializer import ModelSerializer
5
6
7 class TodoItemSerializer(ModelSerializer):
8     class Meta:
9         model = 'todo.TodoItem'
10        publish_fields = ('done', 'text')
11        update_fields = ('done', )
```

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon.serializers.model_serializer import ModelSerializer
5
6
7 class TodoItemSerializer(ModelSerializer):
8     class Meta:
9         model = 'todo.TodoItem'
10        publish_fields = ('done', 'text')
11        update_fields = ('done', )
```

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon.serializers.model_serializer import ModelSerializer
5
6
7 class TodoItemSerializer(ModelSerializer):
8     class Meta:
9         model = 'todo.TodoItem'
10        publish_fields = ('done', 'text')
11        update_fields = ('done', )
```

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon.serializers.model_serializer import ModelSerializer
5
6
7 class TodoItemSerializer(ModelSerializer):
8     class Meta:
9         model = 'todo.TodoItem'
10        publish_fields = ('done', 'text')
11        update_fields = ('done', )
```

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon.serializers.model_serializer import ModelSerializer
5
6
7 class TodoItemSerializer(ModelSerializer):
8     class Meta:
9         model = 'todo.TodoItem'
10        publish_fields = ('done', 'text')
11        update_fields = ('done', )
```

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon.serializers.model_serializer import ModelSerializer
5
6
7 class TodoItemSerializer(ModelSerializer):
8     class Meta:
9         model = 'todo.TodoItem'
10        publish_fields = ('done', 'text')
11        update_fields = ('done', )
```

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon import route_handler
5 from swampdragon.route_handler import ModelRouter
6
7 from .models import TodoList, TodoItem
8 from .serializers import TodoListSerializer, TodoItemSerializer
9
10
11 class TodoItemRouter(ModelRouter):
12     route_name = 'todo-item'
13     serializer_class = TodoItemSerializer
14     model = TodoItem
15
16     def get_object(self, **kwargs):
17         return self.model.objects.get(pk=kwargs['id'])
18
19     def get_query_set(self, **kwargs):
20         return self.model.objects.filter(todo_list__id=kwargs['list_id'])
21
22
23 route_handler.register(TodoItemRouter)
```

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon import route_handler
5 from swampdragon.route_handler import ModelRouter
6
7 from .models import TodoList, TodoItem
8 from .serializers import TodoListSerializer, TodoItemSerializer
9
10
11 class TodoItemRouter(ModelRouter):
12     route_name = 'todo-item'
13     serializer_class = TodoItemSerializer
14     model = TodoItem
15
16     def get_object(self, **kwargs):
17         return self.model.objects.get(pk=kwargs['id'])
18
19     def get_query_set(self, **kwargs):
20         return self.model.objects.filter(todo_list__id=kwargs['list_id'])
21
22
23 route_handler.register(TodoItemRouter)
```

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon import route_handler
5 from swampdragon.route_handler import ModelRouter
6
7 from .models import TodoList, TodoItem
8 from .serializers import TodoListSerializer, TodoItemSerializer
9
10
11 class TodoItemRouter(ModelRouter):
12     route_name = 'todo-item'
13     serializer_class = TodoItemSerializer
14     model = TodoItem
15
16     def get_object(self, **kwargs):
17         return self.model.objects.get(pk=kwargs['id'])
18
19     def get_query_set(self, **kwargs):
20         return self.model.objects.filter(todo_list__id=kwargs['list_id'])
21
22
23 route_handler.register(TodoItemRouter)
```

get\_list

delete

create

get\_single

update

subscribe

unsubscribe

```
1 # -*- coding: utf-8 -*-
2
3 # Swamp Dragon
4 from swampdragon import route_handler
5 from swampdragon.route_handler import ModelRouter
6
7 from .models import TodoList, TodoItem
8 from .serializers import TodoListSerializer, TodoItemSerializer
9
10
11 class TodoItemRouter(ModelRouter):
12     route_name = 'todo-item'
13     serializer_class = TodoItemSerializer
14     model = TodoItem
15
16     def get_object(self, **kwargs):
17         return self.model.objects.get(pk=kwargs['id'])
18
19     def get_query_set(self, **kwargs):
20         return self.model.objects.filter(todo_list__id=kwargs['list_id'])
21
22
23 route_handler.register(TodoItemRouter)
```

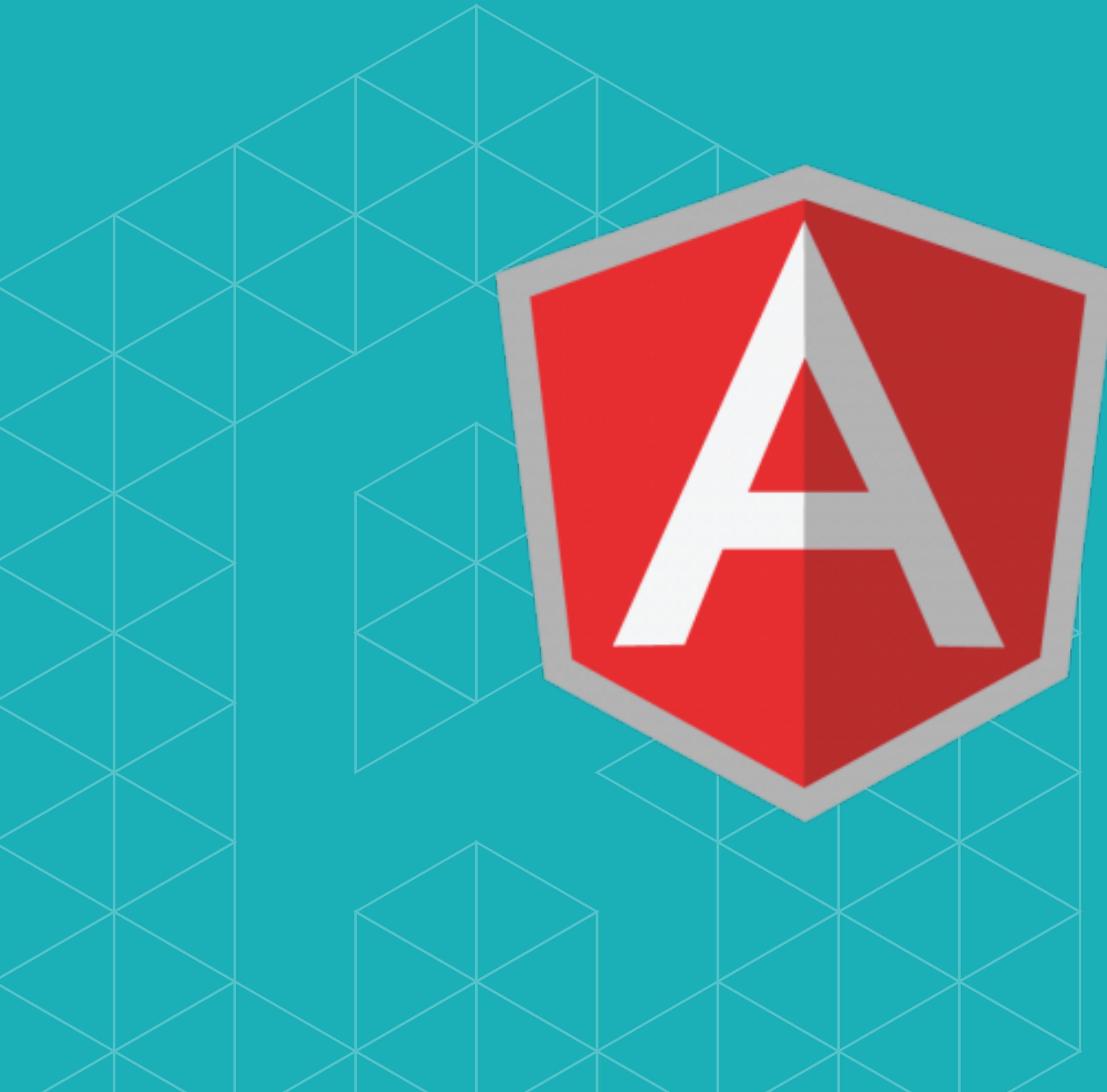
Swamp Dragon - DjangoCon

127.0.0.1:8000

# DjangoCon

Todo list for realtime talk at DjangoCon Europe

- Submit talk proposal
- Panic when talk actually gets accepted
- Break out in a cold sweat
- Break out in a cold sweat...again
- Don't completely mess up the actual talk



```
1 <div class="row" ng-controller="TodoListCtrl">
2   {%
3     verbatim %
4     <h1>{{ todoList.name }}</h1>
5     <p>{{ todoList.description }}</p>
6
7     <div ng-repeat="item in todoItems" class="todos">
8       . . .
9     </div>
10    {%
11      endverbatim %
12    </div>
```

```
1 <div class="row" ng-controller="TodoListCtrl">
2   {%
3     verbatim %
5   <h1>{{ todoList.name }}</h1>
4   <p>{{ todoList.description }}</p>
6   <div ng-repeat="item in todoItems" class="todos">
7     . . .
8   </div>
9   {%
10    endverbatim %
11  </div>
```

```
1 <div class="row" ng-controller="TodoListCtrl">
2   {%
3     verbatim %
4   %}
5   <h1>{{ todoList.name }}</h1>
6   <p>{{ todoList.description }}</p>
7
8   <div ng-repeat="item in todoItems" class="todos">
9     . . .
10    </div>
11  {%
12    endverbatim %
13  }
14 </div>
```

```
1 $dragon.onReady(function() {
2     $dragon.subscribe('todo-item', $scope.channel, {todo_list_id: 1}).then(function(response) {
3         $scope.dataMapper = new DataMapper(response.data);
4     });
5
6     $dragon.getSingle('todo-list', {id:1}).then(function(response) {
7         $scope.todoList = response.data;
8     });
9
10    $dragon.getList('todo-item', {list_id:1}).then(function(response) {
11        $scope.todoItems = response.data;
12    });
13});
```

```
$dragon.subscribe('todo-item', $scope.channel, {todo_list_id: 1})
```

```
$dragon.subscribe('todo-item', $scope.channel, {todo_list_id: 1})
```

```
6 $dragon.getSingle('todo-list', {id:1}).then(function(response) {  
7   $scope.todoList = response.data;  
8 }) ;
```

```
10 $dragon.getList('todo-item', {list_id:1}).then(function(response) {  
11   $scope.todoItems = response.data;  
12 }) ;
```

```
1 $dragon.onReady(function () {  
2     $dragon.subscribe('todo-item', ...);  
3     $dragon.getSingle('todo-list', ...);  
4     $dragon.getList('todo-item', ...);  
5 }) ;
```

```
1 $dragon.onReady(function () {  
2     $dragon.subscribe('todo-item', ...);  
3     $dragon.getSingle('todo-list', ...);  
4     $dragon.getList('todo-item', ...);  
5 }) ;
```

```
1 $dragon.onReady(function () {  
2     $dragon.subscribe('todo-item', ...);  
3     $dragon.getSingle('todo-list', ...);  
4     $dragon.getList('todo-item', ...);  
5 }) ;
```

```
1 $dragon.onReady(function () {  
2     $dragon.subscribe('todo-item', ...);  
3     $dragon.getSingle('todo-list', ...);  
4     $dragon.getList('todo-item', ...);  
5 }) ;
```

```
1 $dragon.onChannelMessage(function(channels, message) {  
2   if (indexOf.call(channels, $scope.channel) > -1) {  
3     $scope.$apply(function() {  
4       $scope.dataMapper.mapData($scope.todoItems, message);  
5     });  
6   }  
7 }) ;
```

```
1 $dragon.onChannelMessage(function(channels, message) {  
2   if (indexOf.call(channels, $scope.channel) > -1) {  
3     $scope.$apply(function() {  
4       $scope.dataMapper.mapData($scope.todoItems, message);  
5     });  
6   }  
7 }) ;
```

```
1 $dragon.onChannelMessage(function(channels, message) {  
2     if (indexOf.call(channels, $scope.channel) > -1) {  
3         $scope.$apply(function() {  
4             $scope.dataMapper.mapData($scope.todoItems, message);  
5         });  
6     }  
7 }) ;
```

```
1 $dragon.onChannelMessage(function(channels, message) {  
2     if (indexOf.call(channels, $scope.channel) > -1) {  
3         $scope.$apply(function() {  
4             $scope.dataMapper.mapData($scope.todoItems, message);  
5         }) ;  
6     }  
7 }) ;
```

# DEMO TIME!

Chrome File Edit View History Bookmarks People Window Help

Aaron

Swamp Dragon - DjangoCon

localhost:8000

# DjangoCon

Todo list for realtime talk at DjangoCon Europe

- Submit talk proposal ✓
- Panic when talk actually gets accepted ✓
- Break out in a cold sweat ✓
- Break out in a cold sweat...again ✓
- Don't completely mess up the actual talk

Select todo item to change

localhost:8000/admin/todo/todoitem/

Action: ----- Go 0 of 5 selected

Todo item

Don't completely mess up the actual talk (x)

Break out in a cold sweat...again (✓)

Break out in a cold sweat (✓)

Panic when talk actually gets accepted (✓)

Submit talk proposal (✓)

5 todo items

Swamp Dragon - DjangoCon Eu...

localhost:8000

# DjangoCon

Todo list for realtime talk at DjangoCon Europe

- Submit talk proposal ✓
- Panic when talk actually gets accepted ✓
- Break out in a cold sweat ✓
- Break out in a cold sweat...again ✓
- Don't completely mess up the actual talk

localhost:8000

# DjangoCon

Todo list for realtime talk at DjangoCon Europe

- Submit talk proposal
- Panic when talk actually gets accepted
- Break out in a cold sweat
- Break out in a cold sweat...again
- Don't completely mess up the actual talk

Aaron



**git.io/vUWyn**



# paaS

Platform as a Service



Firebase – Build Realtime A X Aaron

https://www.firebaseio.com

**Firebase** OVERVIEW GETTING STARTED PRICING DOCS LOGIN SIGN UP

# Build Realtime Apps

A powerful API to store and sync data in realtime.

The screenshot shows the Firebase homepage. At the top, there's a navigation bar with links for Overview, Getting Started, Pricing, Docs, Login, and Sign Up. Below the navigation is a large blue header with the text "Build Realtime Apps" and "A powerful API to store and sync data in realtime.". The main content area features two devices: a desktop computer and a smartphone. Both screens show a feed of posts. On the desktop, a post from "Ali" is visible with the caption "Ali shared a link" and a photo of the Golden Gate Bridge. On the smartphone, a post from "Sara" is visible with the caption "Sara shared a link" and the same photo. A notification overlay on the smartphone screen says "Sara is online". The overall theme is real-time data synchronization across multiple platforms.

Aaron

https://www.pubnub.com

COMPANY CUSTOMERS PRESS CONTACT BLOG GET STARTED LOG IN

PubNub® TUTORIALS PRICING DEVELOPERS HOW IT WORKS USE CASES

# The PubNub Data Stream Network

Build and Scale Realtime Apps for Connected Devices.

GET STARTED

Coca-Cola CBS amazon.com Get Taxi YAHOO! M SAP INSTEON



# databases

Database as a Service



[www.leggetter.co.uk/2013/12/09/choosing-realtime-web-app-tech-stack.html](http://www.leggetter.co.uk/2013/12/09/choosing-realtime-web-app-tech-stack.html)

@leggetter



Pusher: Main — Debug Con Aaron

<https://app.pusher.com/apps/111545/console>

## Main

[« Back to all apps](#)

- [Stats](#)
- [App Keys](#)
- [Debug Console](#)
- [WebHooks](#)
- [Collaborators](#)
- [Settings](#)

## Console

displays all events happening for this app

[Clear](#) [Pause](#)

TYPE	SOCKET	DETAILS	TIME
Waiting for events...			

Events triggered by your application will show up here. To test it out, try opening the '[App keys](#)' page in a new tab.

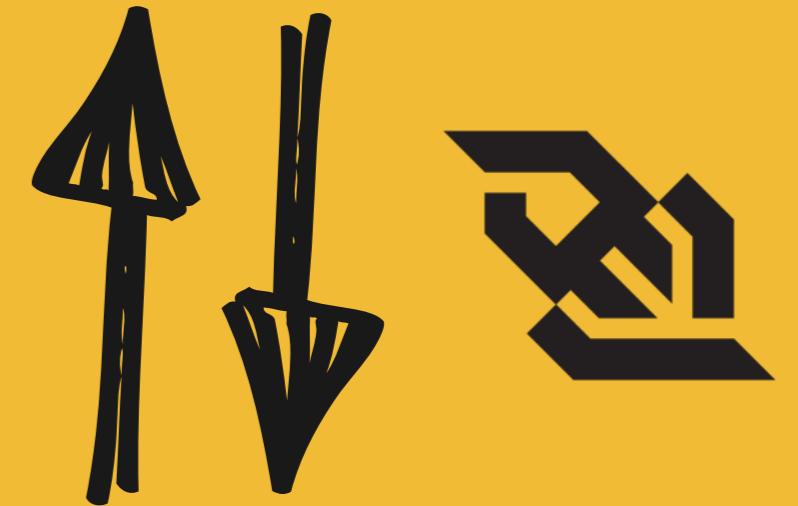
## Create new Event

Channel  
e.g. 'my\_channel'

Event  
e.g. 'user\_created', 'new\_message'

Data  
e.g. '{"name": "Joe", "message\_count": 23}'

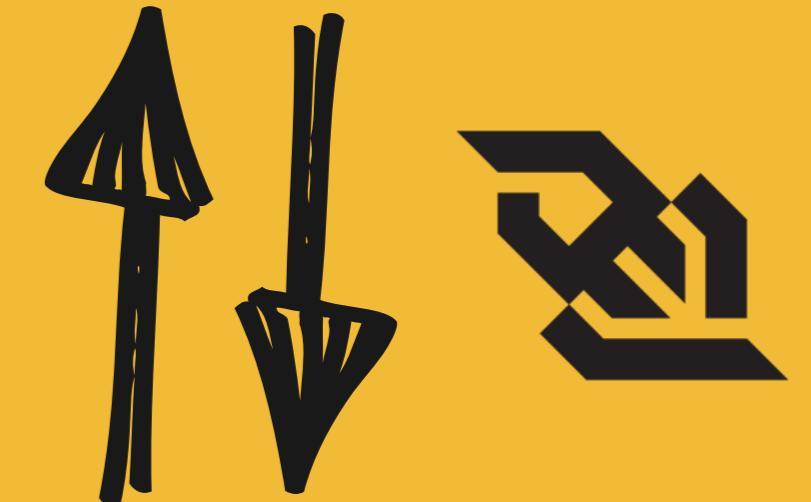
[Send event](#)



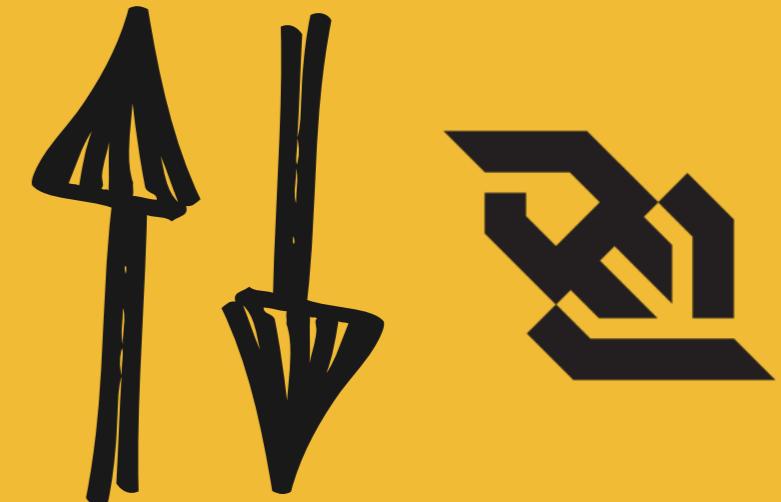
Tornado

redis

# django



# django



```
1 from pusher import Pusher
2
3 pusher = Pusher(
4     app_id=u'111545',
5     key=u'895dec4d68d4e286a48d',
6     secret=u'3cd18f6810c758057767'
7 )
8
9 pusher.trigger(u'channel_name', u'event_name', {u'message': u'hello world'})
```

```
1 from pusher import Pusher  
2  
3 pusher = Pusher(  
4     app_id=u'111545',  
5     key=u'895dec4d68d4e286a48d',  
6     secret=u'3cd18f6810c758057767'  
7 )  
8  
9 pusher.trigger(u'channel_name', u'event_name', {u'message': u'hello world'})
```

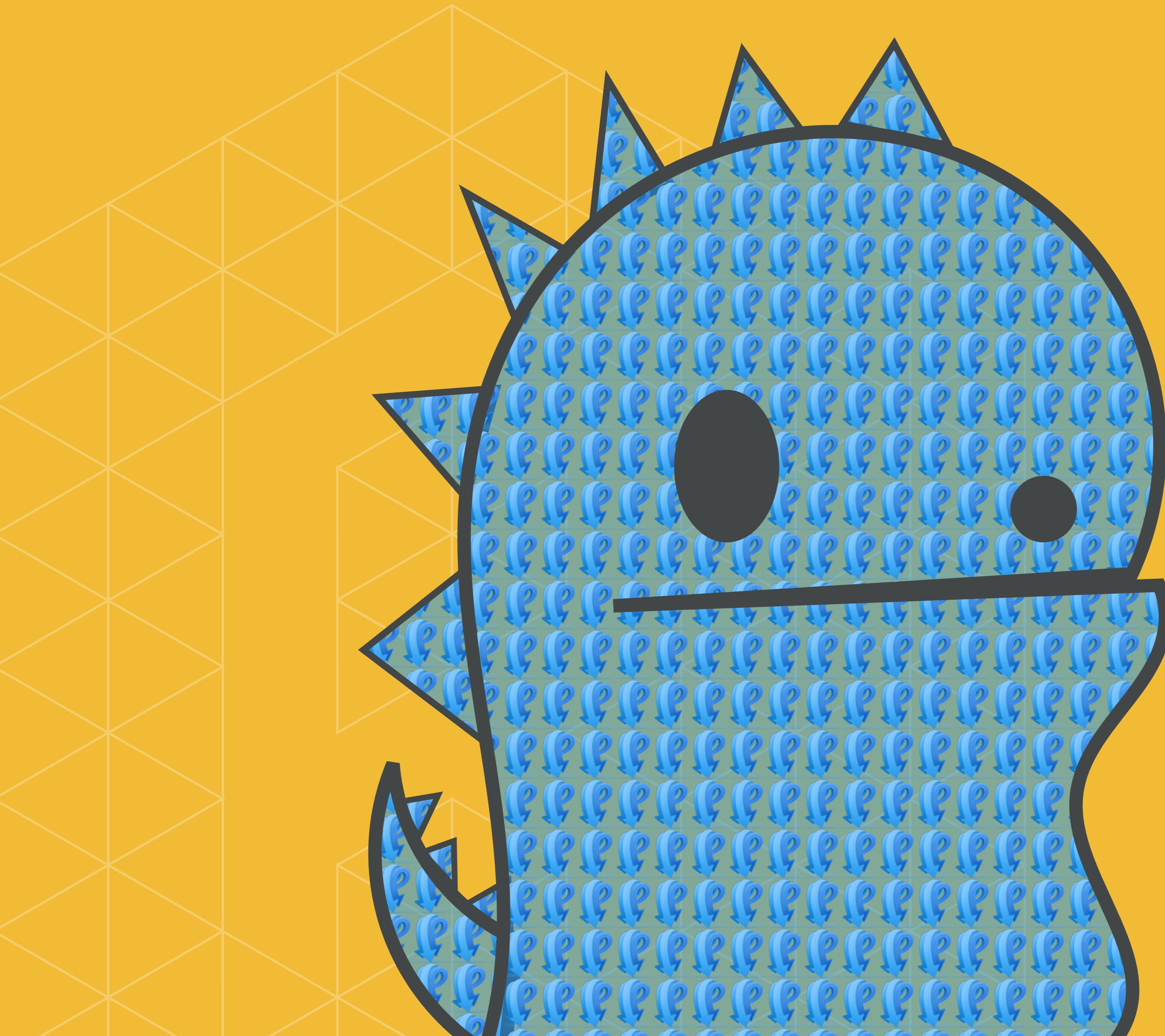
```
1 from pusher import Pusher
2
3 pusher = Pusher(
4     app_id=u'111545',
5     key=u'895dec4d68d4e286a48d',
6     secret=u'3cd18f6810c758057767'
7 )
8
9 pusher.trigger(u'channel_name', u'event_name', {u'message': u'hello world'})
```

```
1 from pusher import Pusher
2
3 pusher = Pusher(
4     app_id=u'111545',
5     key=u'895dec4d68d4e286a48d',
6     secret=u'3cd18f6810c758057767'
7 )
8
9 pusher.trigger(u'channel_name', u'event_name', {u'message': u'hello world'})
```

```
1 from pusher import Pusher
2
3 pusher = Pusher(
4     app_id=u'111545',
5     key=u'895dec4d68d4e286a48d',
6     secret=u'3cd18f6810c758057767'
7 )
8
9 pusher.trigger(u'channel_name', u'event_name', {u'message': u'hello world'})
```

```
1 from pusher import Pusher
2
3 pusher = Pusher(
4     app_id=u'111545',
5     key=u'895dec4d68d4e286a48d',
6     secret=u'3cd18f6810c758057767'
7 )
8
9 pusher.trigger(u'channel_name', u'event_name', {u'message': u'hello world'})
```

```
1 from pusher import Pusher
2
3 pusher = Pusher(
4     app_id=u'111545',
5     key=u'895dec4d68d4e286a48d',
6     secret=u'3cd18f6810c758057767'
7 )
8
9 pusher.trigger(u'channel_name', u'event_name', {u'message': u'hello world'})
```



```
1 class SelfPublishModel(object):
2
3     def _publish(self, action):
4         serializer = self.serializer_class(instance=self)
5         data = serializer.serialize()
6
7         pusher = Pusher(
8             app_id=settings.PUSHER_APP_ID,
9             key=settings.PUSHER_KEY,
10            secret=settings.PUSHER_SECRET
11        )
12
13        pusher.trigger(self.channel_name, action, data)
```

```
1 class SelfPublishModel(object):
2
3     def _publish(self, action):
4         serializer = self.serializer_class(instance=self)
5         data = serializer.serialize()
6
7         pusher = Pusher(
8             app_id=settings.PUSHER_APP_ID,
9             key=settings.PUSHER_KEY,
10            secret=settings.PUSHER_SECRET
11        )
12
13        pusher.trigger(self.channel_name, action, data)
```

```
1 class SelfPublishModel(object):
2
3     def _publish(self, action):
4         serializer = self.serializer_class(instance=self)
5         data = serializer.serialize()
6
7         pusher = Pusher(
8             app_id=settings.PUSHER_APP_ID,
9             key=settings.PUSHER_KEY,
10            secret=settings.PUSHER_SECRET
11        )
12
13         pusher.trigger(self.channel_name, action, data)
```

 Showing **7 changed files** with **49 additions** and **64 deletions**.

[Unified](#) [Split](#)

 <a href="#">README.md</a>	+2 -2	
 <a href="#">talk/server.py</a>	+0 -12	
 <a href="#">talk/talk/settings.py</a>	+4 -4	
 <a href="#">talk/todo/migrations/0001_initial.py</a>	+3 -8	
 <a href="#">talk/todo/mixins.py</a>	+36 -0	
 <a href="#">talk/todo/models.py</a>	+4 -2	
 <a href="#">talk/todo/routers.py</a>	+0 -36	

Aaron

pusher/pusher-angular GitHub, Inc. [US] https://github.com/pusher/pusher-angular

This repository Search Explore Gist Blog Help aaronbassett + ▾ ⌂ ⚙ ⌂

pusher / pusher-angular Watch 22 Star 112 Fork 16

## Pusher Angular Library

33 commits 2 branches 6 releases 6 contributors

branch: master +

### README update

hamchapman authored 15 days ago latest commit a79bf697a8

File	Description	Date
example	Updated example app with readme and removed specific pusher app config	2 months ago
lib	Updated pusher-angular.min.js	15 days ago
public/components	Updated Angular to 1.3.15 for tests	a month ago
tests	Made tests for unbind consistent	15 days ago
.travis.yml	Removed rubbish	7 months ago
LICENSE	v0.1.0 release	8 months ago
README.md	README update	15 days ago
bower.json	Bumped version to 0.1.5	15 days ago

Code Issues Pull requests Wiki Pulse Graphs

SSH clone URL git@github.com:pusher/pusher-angular

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop Download ZIP

```
1 var TodoControllers = angular.module('TodoControllers', []);
2
3 TodoControllers.controller('TodoListCtrl', ['$scope', '$pusher', function
($scope, $pusher) {
4     $scope.todoItems = [];
5
6     var client = new Pusher("895dec4d68d4e286a48d");
7     var pusher = $pusher(client);
8     var todo_items_channel = pusher.subscribe('todo-item');
9     todo_items_channel.bind('updated',
10         function(data) {
11             for(var i=0; i < $scope.todoItems.length; i++) {
12                 if($scope.todoItems[i].id == data.id) {
13                     $scope.todoItems[i] = data;
14                     return;
15                 }
16             }
17         }
18     );
19 }]);
```

```
1 var TodoControllers = angular.module('TodoControllers', []);
2
3 TodoControllers.controller('TodoListCtrl', ['$scope', '$pusher', function
($scope, $pusher) {
4     $scope.todoItems = [];
5
6     var client = new Pusher("895dec4d68d4e286a48d");
7     var pusher = $pusher(client);
8     var todo_items_channel = pusher.subscribe('todo-item');
9     todo_items_channel.bind('updated',
10         function(data) {
11             for(var i=0; i < $scope.todoItems.length; i++) {
12                 if($scope.todoItems[i].id == data.id) {
13                     $scope.todoItems[i] = data;
14                     return;
15                 }
16             }
17         }
18     );
19 }]);
```

```
1 var TodoControllers = angular.module('TodoControllers', []);
2
3 TodoControllers.controller('TodoListCtrl', ['$scope', '$pusher', function
($scope, $pusher) {
4     $scope.todoItems = [];
5
6     var client = new Pusher("895dec4d68d4e286a48d");
7     var pusher = $pusher(client);
8     var todo_items_channel = pusher.subscribe('todo-item');
9     todo_items_channel.bind('updated',
10         function(data) {
11             for(var i=0; i < $scope.todoItems.length; i++) {
12                 if($scope.todoItems[i].id == data.id) {
13                     $scope.todoItems[i] = data;
14                     return;
15                 }
16             }
17         }
18     );
19 }]);
```

```
1 var TodoControllers = angular.module('TodoControllers', []);
2
3 TodoControllers.controller('TodoListCtrl', ['$scope', '$pusher', function
($scope, $pusher) {
4     $scope.todoItems = [];
5
6     var client = new Pusher("895dec4d68d4e286a48d");
7     var pusher = $pusher(client);
8     var todo_items_channel = pusher.subscribe('todo-item');
9     todo_items_channel.bind('updated',
10         function(data) {
11             for(var i=0; i < $scope.todoItems.length; i++) {
12                 if($scope.todoItems[i].id == data.id) {
13                     $scope.todoItems[i] = data;
14                     return;
15                 }
16             }
17         }
18     );
19 }]);
```

Aaron

```
48     {% endblock main_content %}49  </div>50  <!-- /.container -->515253  <!-- Bootstrap core JavaScript54  ===== -->55  <!-- Placed at the end of the document so the pages load faster -->56  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>57  <script src="//netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></script>58  <script type="text/javascript" src="{{ STATIC_URL }}libs/angular-1.2.6/angular.min.js?v=1"></script>59  <script type="text/javascript" src="{{ STATIC_URL }}libs/angular-1.2.6/angular-route.min.js?v=1"></script>60  <script type="text/javascript" src="{{ STATIC_URL }}libs/angular-1.2.6/angular-cookies.min.js?v=1"></script>61  <script type="text/javascript" src="{{ STATIC_URL }}libs/angular-1.2.6/angular-resource.min.js?v=1"></script>6263  {% swampdragon_settings %}64  <script type="text/javascript" src="{{ STATIC_URL }}swampdragon/js/dist/swampdragon.js"></script>65  <script type="text/javascript" src="{{ STATIC_URL }}swampdragon/js/dist/datamapper.js"></script>66  <script type="text/javascript" src="{{ STATIC_URL }}swampdragon/js/angular/services.js?v=1"></script>6768  <script type="text/javascript" src="{{ STATIC_URL }}js/app.js?v=% now 'U' %}></script>69  <script type="text/javascript" src="{{ STATIC_URL }}js/company-controllers.js?v=1"></script>7071  </body>72  </html>
```



# DEMO TIME!

Chrome File Edit View History Bookmarks People Window Help

Aaron

Pusher - DjangoCon Europe x

127.0.0.1:8000

Pusher: Main — Debug Cor x

https://app.pusher.com/ap...

Console displays all events happening for this app

Pause Clear

Type Socket Details Time

## Create new Event

Channel  
e.g. 'my\_channel'

Event  
e.g. 'user\_created', 'new\_message'

Data  
e.g. '{"name": "Joe",  
"message\_count": 23}'

127.0.0.1:8000

Google

127.0.0.1:8000/admin/todo/todoitem/

Django administration

Welcome, aaronbassett. View site / Change password / Log out

Home > Todo > Todo items

Select todo item to change

Add todo item +

0 todo items



**git.io/vker1**



The screenshot shows a customer service application window. At the top, there are several icons: a back arrow, a pencil, a user profile, a flag, a tag, and an envelope. Below the header, the title "Returns" is displayed, followed by two green tags: "50% promo" and "returns". A toolbar with various editing functions (bold, italic, underline, etc.) is visible above the main message area. A context menu is open over a message, with "Returns" highlighted in blue. Other options in the menu include "Shipping" and "Gift Cards". The message itself is from "Mike Adams" and says: "Peter, Can you help Grace while I'm away? Would be good to set up a saved reply for answering these questions in the future. Thanks." The status of the message is "Closed".

>Returns 50% promo x returns x

B I U

Status: Closed Assign to

Save This Reply ... Send Reply

Mike Adams added a note

Peter,

Can you help Grace while I'm away? Would be good to set up a saved reply for answering these questions in the future. Thanks.



**Mike Adams** added a note

Peter,

Can you help Grace while I'm away? Would be good to set up a saved reply for answering these questions in the future. Thanks.

```
1 from pusher import Pusher
2
3 pusher = Pusher(
4     app_id=u'111545',
5     key=u'895dec4d68d4e286a48d',
6     secret=u'3cd18f6810c758057767'
7 )
8
9 pusher.trigger(u'channel_name', u'event_name', {u'message': u'hello world'})
```

```
1 class PusherMixin(object):
2
3     def render_to_response(self, context, **response_kwargs):
4
5         channel = u"{}_{pk}".format(
6             model=self.object._meta.model_name,
7             pk=self.object.pk
8         )
9         event_data = { 'user': self.request.user.username }
10
11     pusher = Pusher(app_id=settings.PUSHER_APP_ID,
12                     key=settings.PUSHER_KEY,
13                     secret=settings.PUSHER_SECRET)
14     pusher.trigger(
15         [channel, ],
16         self.pusher_event_name,
17         event_data
18     )
19
20     return super(PusherMixin, self).render_to_response(context,
**response_kwargs)
```

```
1 class PusherMixin(object):
2
3     def render_to_response(self, context, **response_kwargs):
4
5         channel = u"{}_{pk}".format(
6             model=self.object._meta.model_name,
7             pk=self.object.pk
8         )
9         event_data = { 'user': self.request.user.username }
10
11     pusher = Pusher(app_id=settings.PUSHER_APP_ID,
12                     key=settings.PUSHER_KEY,
13                     secret=settings.PUSHER_SECRET)
14     pusher.trigger(
15         [channel, ],
16         self.pusher_event_name,
17         event_data
18     )
19
20     return super(PusherMixin, self).render_to_response(context,
**response_kwargs)
```

```
1 class PusherMixin(object):
2
3     def render_to_response(self, context, **response_kwargs):
4
5         channel = u"{}_{pk}".format(
6             model=self.object._meta.model_name,
7             pk=self.object.pk
8         )
9         event_data = { 'user': self.request.user.username }
10
11     pusher = Pusher(app_id=settings.PUSHER_APP_ID,
12                     key=settings.PUSHER_KEY,
13                     secret=settings.PUSHER_SECRET)
14     pusher.trigger(
15         [channel, ],
16         self.pusher_event_name,
17         event_data
18     )
19
20     return super(PusherMixin, self).render_to_response(context,
**response_kwargs)
```

```
1 class PusherMixin(object):
2
3     def render_to_response(self, context, **response_kwargs):
4
5         channel = u"{}_{pk}".format(
6             model=self.object._meta.model_name,
7             pk=self.object.pk
8         )
9
10        event_data = { 'user': self.request.user.username }
11
12        pusher = Pusher(app_id=settings.PUSHER_APP_ID,
13                        key=settings.PUSHER_KEY,
14                        secret=settings.PUSHER_SECRET)
15
16        pusher.trigger(
17            [channel, ],
18            self.pusher_event_name,
19            event_data
20        )
21
22        return super(PusherMixin, self).render_to_response(context,
23 **response_kwargs)
```

```
1 class PusherMixin(object):
2
3     def render_to_response(self, context, **response_kwargs):
4
5         channel = u"{}_{pk}".format(
6             model=self.object._meta.model_name,
7             pk=self.object.pk
8         )
9         event_data = {'user': self.request.user.username}
10
11         pusher = Pusher(app_id=settings.PUSHER_APP_ID,
12                         key=settings.PUSHER_KEY,
13                         secret=settings.PUSHER_SECRET)
14         pusher.trigger(
15             [channel, ],
16             self.pusher_event_name,
17             event_data
18         )
19
20     return super(PusherMixin, self).render_to_response(context,
**response_kwargs)
```

```
1 var pusher = new Pusher('{{ settings.PUSHER_KEY }}');
2 var channel = pusher.subscribe('model_{{ object.pk }}');
3 channel.bind('update', function(data) {
4   alert(data.user + " has begun updating this object");
5 });
```

```
1 var pusher = new Pusher('{{ settings.PUSHER_KEY }}');
2 var channel = pusher.subscribe('model_{{ object.pk }}');
3 channel.bind('update', function(data) {
4   alert(data.user + " has begun updating this object");
5 });
```

```
1 var pusher = new Pusher('{{ settings.PUSHER_KEY }}');
2 var channel = pusher.subscribe('model_{{ object.pk }}');
3 channel.bind('update', function(data) {
4   alert(data.user + " has begun updating this object");
5 });
```

```
1 var pusher = new Pusher('{{ settings.PUSHER_KEY }}');
2 var channel = pusher.subscribe('model_{{ object.pk }}');
3 channel.bind('update', function(data) {
4   alert(data.user + " has begun updating this object");
5 });
```

Aaron

django-pusherable 0.1.0 : F

Python Software Foundation [US] <https://pypi.python.org/pypi/django-pusherable>

python

» Package Index > django-pusherable > 0.1.0

PACKAGE INDEX »

- Browse packages
- Package submission
- List trove classifiers
- List packages
- RSS (latest 40 updates)
- RSS (newest 40 packages)
- Python 3 Packages
- PyPI Tutorial
- PyPI Security
- PyPI Support
- PyPI Bug Reports
- PyPI Discussion
- PyPI Developer Info

ABOUT »

NEWS »

DOCUMENTATION »

DOWNLOAD »

COMMUNITY »

FOUNDATION »

CORE DEVELOPMENT »

## django-pusherable 0.1.0

*Real time object access notifications via Pusher*

=====

django-pusherable

=====

Real-time object access notifications via Pusher\_

.. \_Pusher: <https://pusher.com>

Installation

-----

Install django-pusherable::

pip install django-pusherable

Configuration

-----

Then add `pusherable` to your `INSTALLED\_APPS`. You will also need to add your Pusher app credentials to `settings.py`. These are available on your app keys page.::

```
PUSHER_APP_ID = u"\""
PUSHER_KEY = \"\""
```

Initial ideas on how to make this a little more robust #5

**Open** chrisjones-brack3t wants to merge 10 commits into pusher:master from chrisjones-brack3t:feature/ideas

Conversation 10 Commits 10 Files changed 2 +151 -51

Labels None yet

Milestone No milestone

Assignee No one—assign yourself

Notifications Unsubscribe You're receiving notifications because you were mentioned.

5 participants

Lock pull request

Initial ideas on how to make this a little more robust

I haven't included any tests nor is this complete. This is more for presenting a different direction for the mixins. I've pulled some ideas and code from my own project [django-braces](#).





git.io/vk9uu



@aaronbassett