Working with Webhooks

Lorna Mitchell, Nexmo

What is a Webhook?

An HTTP POST request.

Why Webhooks?

Event-driven HTTP = Webhooks

Webhooks in the Wild

Slack Integrations

github APP 5:37 PM

[nexmo-python-quickstart] New branch "account-examples" was pushed by lornajane

[nexmo-community/nexmo-python-quickstart] Pull request submitted by lornajane

#46 Add get-balance and configure-account building blocks

We've added a couple of building blocks for account, so I thought I'd add python examples to get us started.

GitHub Builds

All checks have failed

1 failing check

continuous-integration/travis-ci/pr — The Travis CI build failed

This branch has no conflicts with the base branch

Only those with write access to this repository can merge pull requests.

Webhook Use Cases

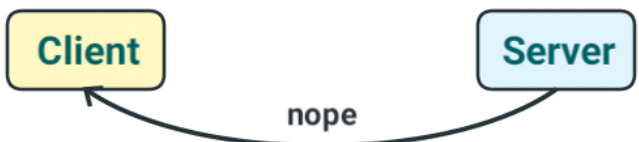
- Notify of events
- Deliver data when available
- Broadcast to multiple receivers as-it-happens

How APIs Work

How APIs Work

How APIs Work

How APIs Work



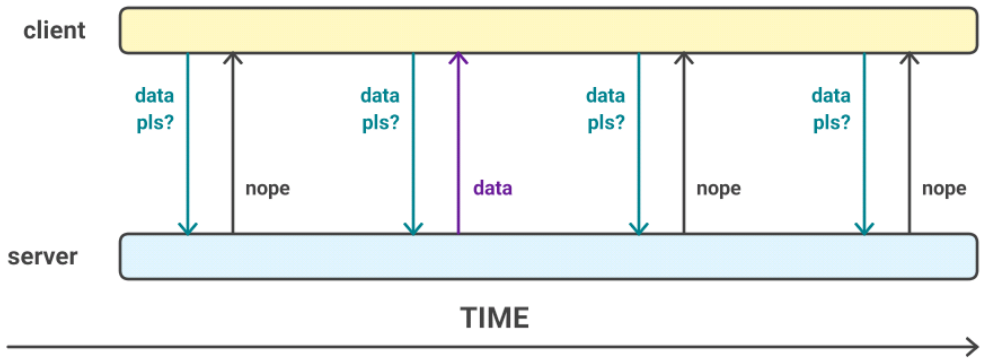
How Webhooks Work

How Webhooks Work

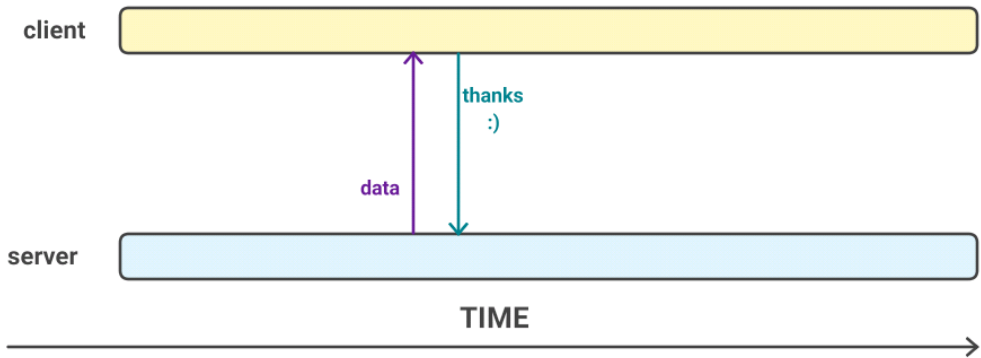
How Webhooks Work

What About Time?

APIs Over Time



Webhooks Over Time



Where To Webhook To?

Webhooks Need Pre-arrangement

With APIs, the client calls the server.

With Webhooks, the client has to register with the server, to get data later.

Designing Webhooks: Example

A GitHub "push" event contains nested data.

```
"ref": "refs/tags/simple-tag",
"before": "a10867b14bb761a232cd80139fbd4c0d33264240",
"after": "0000000000000000000000000000000000000000",
"compare": "https://github.com/Codertocat/Hello-World/compare/a10867b14bb
"commits": [ ],
"repository": {
    "owner": { },
    "html_url": "https://github.com/Codertocat/Hello-World"
},
"sender": { }
```

Designing Webhooks

Think about how your users will use the API.

- What's their context/technology/device?
- What are they doing when they call the endpoint?
- What are they likely to do next?

Receiving Webhooks

Warning: minor tangent ahead

Ngrok for Testing Webhooks

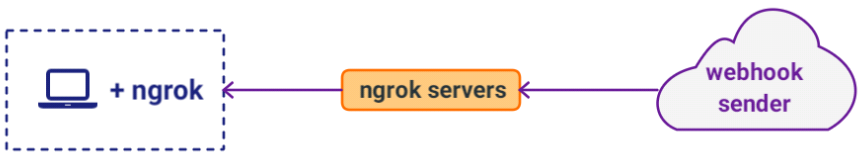
https://ngrok.com/ - secure tunnel to your dev platform

Use this tool to:

- webhook into code running locally
- inspect the request and response of the webhook
- replay requests and see the responses

Ngrok for Testing Webhooks

Start the tunnel on your laptop: receive a public URL



Example: Nexmo SMS

When you register a phone number and receive an SMS, your application receives a webhook.

I'm running open endpoints on the internet and accepting data, now what?

Webhook Security

When working with webhooks:

- be aware of attack vectors
- always use SSL
- consider shared secrets and hashing
- all good HTTP security practices apply

Nexmo SMS Security

Nexmo can sign messages using a shared secret.

The PHP library https://github.com/nexmo/nexmo-php can do this for you.

```
$signature = new \Nexmo\Client\Signature($_GET,
    SIGNATURE_SECRET, 'sha256');

$isValid = $signature->check($_GET['sig']);
```

Receiving Webhooks: Best Practice

It's just an HTTP request! Advice:

- DO: accept, store and acknowledge quickly
- DON'T: process before acknowledging

Using Queues in PHP Applications

Queues protect you against bursty traffic.

Queues separate work from webservers.

- This example uses https://beanstalkd.github.io/ and Laravel
- Other good alternatives: Redis or Laravel Horizon, Amazon SQS, RabbitMQ

SlimPHP and Beanstalkd

```php
$container = $app->getContainer();

$container['queue'] = function ($container) {
    return new \Pheanstalk\Pheanstalk(
        getenv('QUEUE_HOST'),
        getenv('QUEUE_PORT')
    );
};
```

Pro-tip: Try `vlucas\phpdotenv` for dev platform environment vars.

SlimPHP and Beanstalkd

```
13 $app->get('/webhooks/inbound-sms', function ($request, $response, $arg
14     $params = $request->getQueryParams();
15
16     $data = ["event" => "message",
17         "text" => $params['text'],
18         "receivedAt" => date("U"),
19         "payload" => $params];
20     error_log("New message: " . $params['text']);
21     $this->queue->useTube('sms')
22         ->put(json_encode($data));
23 });
```

Laravel and Beanstalkd

```
1  use App\Jobs\InboundSms;
2  Route::get('inbound-sms', function(Request $request) {
3      // get incoming parameters (includes GET and POST)
4      $params = $request->input();
5      $data = ["event" => "message", "text" => $params['text'],
6          "receivedAt" => date("U"), "payload" => $params];
7      error_log("New message: " . $params['text']);
8
9      InboundSms::dispatch($data);
10     return "OK";
11 });
```

So The Data is In a Queue. Now What?

Let's talk about Workers

Workers are long-running scripts that process a series of jobs.

Workers need to be independent:

- if things go wrong, exit
- separate tool to monitor/restart as needed
- beware long-running process hazards
- everything processed "at least once" (but maybe more than once, and in any order...)

Processing SMSes with PHP

```
13 while($job = $queue->reserve()) {
14     $received = json_decode($job->getData(), true);
15     error_log($received['text']);
16
17     // delegate to the class that will do the work
18     $worker->process($received);
19
20     $queue->delete($job);
21 }
```

Processing SMSes with PHP

```
 9 class Worker {
10     public function process($data) {
11         $signature = new \Nexmo\Client\Signature( $data['payload'],
12             getenv('NEXMO_API_SIGNATURE_SECRET'),
13             'sha256');
14         if ($signature->check($data['payload']['sig'])) {
15             $this->write_to_file($this->outfile, $data['text']);
16             return true;
17         }
18         return false;
19     }
```

Processing SMSes with Laravel

```
1  Route::get('inbound-sms', function(Request $request)
2  {
3      $params = $request->input();
4      $data = ["event" => "message", "text" => $params['text'],
5          "receivedAt" => date("U"), "payload" => $params];
6      error_log("New message: " . $params['text']);
7
8      InboundSms::dispatch($data);
9      return "OK";
10 });
```

Publishing Webhooks

Sending webhooks is an implementation of Publish/Subscribe pattern.

- Put desired webhooks into a queue, and send at a steady rate.
- Everything we discussed about queues and workers applies.
- DO NOT send from live webserver.

Webhooks

... are awesome :)

Webhooks in Your Applications

- Use them WHEN you want to notify other systems
- Examples of HOW to use webhooks hopefully gave you some ideas
- Webhooks are HTTP: we already understand this

Thanks!

- Feedback please! https://joind.in/
- PHP Web Services from O'Reilly
- Nexmo: https://nexmo.com
- Me: https://lornajane.net
- Ngrok: https://ngrok.com/
- Code: https://github.com/lornajane/incoming-sms-beanstalkd-php

Bonus slides

Serverless Webhook Endpoints

Serverless technology:

- Functions as a Service
- Scalable: ideal for bursty workloads
- Pay-as-you-go, and with free tiers
- PHP supported on some platforms (they all support NodeJS)

Code: https://github.com/lornajane/fortunes-by-sms

Serverless PHP SMS Reply Code Example

```
 4 function main(array $params) : array {
 5     $cookie = getCookies()[random_int(0, 430)];
 6     parse_str($params['__ow_query'], $query);
 7
 8     $url = "https://rest.nexmo.com/sms/json";
 9     $options = ["http" => [ "method" => "POST", "content" => json_enco
10     $context = stream_context_create($options);
11     $response = file_get_contents($url, false, $context);
12     return ["body" => $cookie];
13 }
```

Serverless PHP SMS Reply Deployment

Zip the code (and dependencies if needed), then deploy

```
rm -f sms-fortune.zip
zip -rq sms-fortune.zip index.php

ibmcloud fn action update sms-fortune/incoming-php \
    --kind php:7.2 --web raw sms-fortune.zip
```

Link your Nexmo number to the URL of this action, share and enjoy.

Here's one I made earlier: 201-579-9898