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

(D 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

×

X

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 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. ... may also agree security (e.g. shared secret) at this stage.





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



Shared Secrets

Share a secret in advance, then transmit all the fields and a signature hash created using the secret.







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 SlimPHP
- Other good alternatives: Redis or Laravel Horizon, Amazon SQS, RabbitMQ





SlimPHP and Beanstalkd

```
1 $container = $app->getContainer();
2
3 $container['queue'] = function ($container) {
4    return new \Pheanstalk\Pheanstalk(
5       getenv('QUEUE_HOST'),
6       getenv('QUEUE_PORT')
7    );
8 };
```

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->getOuervParams():
15
16
       $data = ["event" => "message",
17
           "text" => $params['text'].
           "receivedAt" => date("U").
18
19
           "payload" => $params]:
       error_log("New message: " . $params['text']);
20
21
       $this->queue->useTube('sms')
22
           ->put(json_encode($data));
23 });
```



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 SMS with PHP

```
14 $queue->watch("sms"):
15
16 while($job = $queue->reserve()) {
17
       $received = json_decode($job->getData(), true);
18
       error log($received['text']):
19
20
       // delegate to the class that will do the work
21
       $worker->process($received):
22
23
       $queue->delete($job);
24 }
```



Processing SMS with PHP

1	<pre>class SecureWorker extends Worker {</pre>
2	<pre>public function process(\$data) {</pre>
3	<pre>if (\$this->validateMessageSignature(\$data)) {</pre>
4	<pre>\$this->write_to_file(\$this->outfile, \$data['text']);</pre>
5	<pre>return true; }</pre>
6	}
7	<pre>protected function validateMessageSignature(\$data) {</pre>
8	<pre>\$signature = new \Nexmo\Client\Signature(</pre>
9	<pre>\$data['payload'], getenv('NEXMO_API_SIGNATURE_SECRET'), 's</pre>
10	<pre>return \$signature->check(\$data['payload']['sig']);</pre>
11	}



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!

- PHP Web Services from O'Reilly
- Me: https://lornajane.net
- Nexmo: https://nexmo.com
- Ngrok: https://ngrok.com/



Code: https://github.com/lornajane/incoming-sms-heanstalkd-php

