

AMPBench, an AMP URL Validation and Troubleshooting Application

Debug AMP pages with AMPBench, an open source app from the AMP Project.

Pieter Greyling - Partner Engineer/Senior Developer Advocate at Google - 25 October 2016

<https://medium.com/@greyling/ampbench-an-amp-url-validation-and-troubleshooting-application-d33ee83df604#.lqdq7098v>

What does AMPBench do?

[AMPBench is an open source web application and service](#) that validates AMP pages and their associated Structured Data at a user provided page URL.

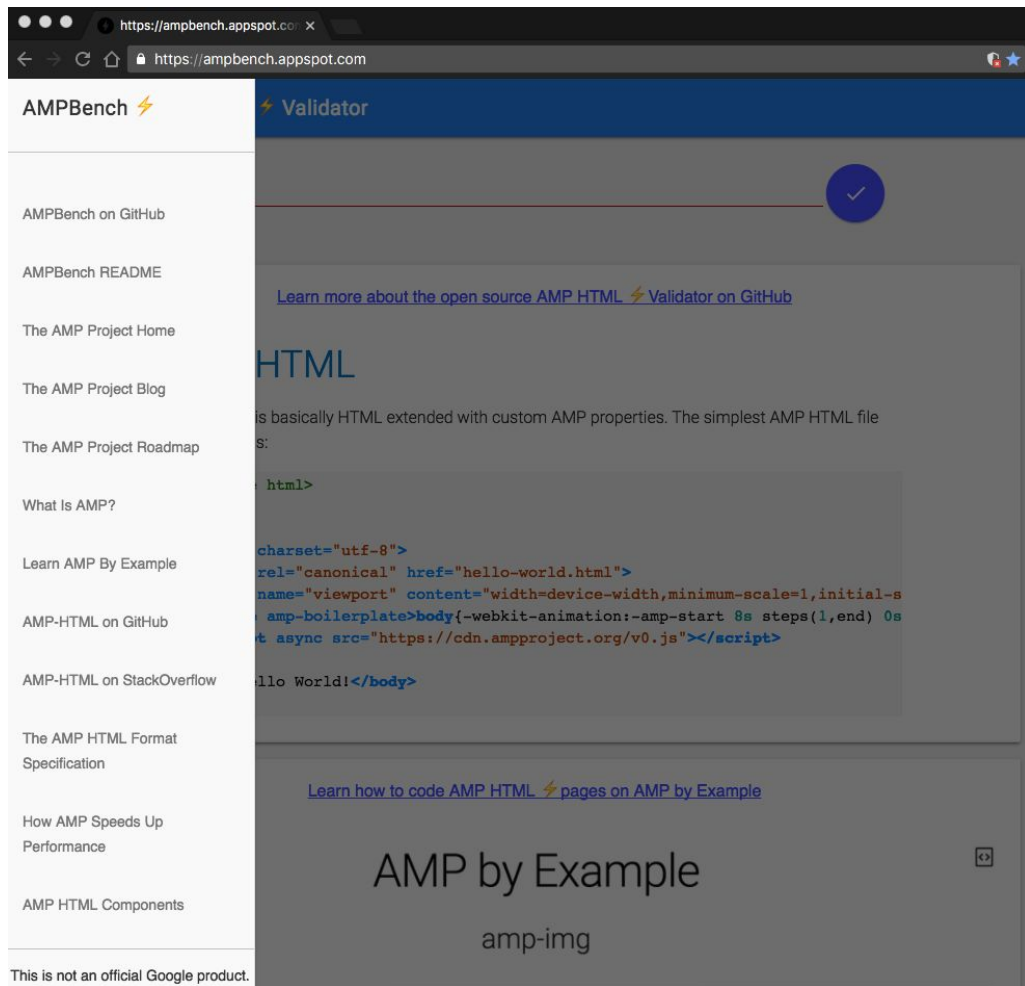
The source code for the service is open for contributions on GitHub at:

[ampbench - AMPBench: AMP URL validation and troubleshooting tools - github.com](#)

A live instance of AMPBench is currently hosted on a [Google Cloud Platform Node.js App Engine Flexible Environment](#) at: <https://ampbench.appspot.com/>

The latter can be used by the community for ad-hoc testing of AMP pages on the Web. Later in this article I will document how to run (or host) a development or production instance of AMPBench. This documentation is also available on the [AMPBench README at GitHub](#).

AMPBench is written in JavaScript ([ES6/2015](#)) with [Node.js](#), is licensed under the [Apache 2.0 LICENSE](#), is not an official Google product, and is part of the open source [AMP Project](#).



During AMP URL validation, it builds referable, support-friendly sharable URLs such as the following:

- <https://ampbench.appspot.com/validate?url=https://ampbyexample.com/>


<https://ampbyexample.com/>AMP Checks **[PASS]**

AMP Check	Status	Results [@2016-10-21T21:21:46.486Z][validator-signature:fe2d79fbb047e8c8]
AMP URL	PASS	https://ampbyexample.com/
Canonical Link	PASS	https://ampbyexample.com/
AMP HTML Link in Canonical page?	INFO	[Canonical URL is reachable][AMP page Canonical link points to the AMP page itself: is this a standalone AMP?]
AMP HTML Link in AMP page?	PASS	AMP page appears free of AMP-HTML rel link(s)
AMP HTML Required Markup Issues [0]	PASS	AMP HTML required markup content appears to be valid
Indexed in Google AMP Cache?	PASS	[PASS] [HTTP: 200] URL is indexed in the Google AMP Cache: https://cdn.ampproject.org/c/s/ampbyexample.com/ [PASS] [HTTP: 200] URL is reachable in the Google AMP Viewer: https://www.google.com/amp/s/ampbyexample.com/


Validation checks point to relevant, contextual documentation

Each section of the AMPBench user interface attempts to guide users to helpful documentation that is relevant for the particular aspect of the AMP URL that is being verified.

AMP Checks [PASS]	
AMP Check	Status
AMP URL	PASS
Canonical Link	PASS
AMP HTML Link in Canonical page?	WARNING
AMP HTML Link in AMP page?	PASS
AMP HTML Required Markup Issues [0]	PASS
Indexed in Google AMP Cache?	PASS
Response Times (milliseconds)	WARNING
Is access allowed for Googlebots? (robots.txt)	PASS
Robots Meta Tag	PASS
X-Robots-Tag Header	PASS
Redirects via User Agent Googlebot Smartphone	PASS



[LEARN](#)
[BUILD](#)
[NEWS](#)
[CONTRIBUTE](#)
[SUPPORT](#)

 English

Tutorials

- Create Your First AMP Page

Guides

- Styling & Layout
- Include Images & Video
- Include Iframes
- Include Third-Party Content
- Make Your Page Discoverable
- Validate AMP Pages
- Configure Analytics

Reference

- Components / Tags
- AMP HTML Specification
- Experimental features
- AMP Validation Errors

Make Your Page Discoverable

In some cases, you might want to have both a non-AMP and an AMP version of the same page, for example, a news article. Consider this: If Google Search finds the non-AMP version of that page, how does it know there's an AMP version of it?

Linking pages with <link>

In order to solve this problem, we add information about the AMP page to the non-AMP page and vice versa, in the form of `<link>` tags in the `<head>`.

Add the following to the non-AMP page:

```
<link rel="amphtml" href="https://www.example.com/url/to/amp/document.html">
```


And this to the AMP page:

```
<link rel="canonical" href="https://www.example.com/url/to/full/document.html">
```

Contents

- Linking pages with <link>
- What if I only have one page?
- Integrate with third-party platforms through additional metadata
- Use Schema.org for most search engines
- Other metadata for even more platforms

AMP Checks [PASS]	
AMP Check	Status
AMP URL	PASS
Canonical Link	PASS
AMP HTML Link in Canonical page?	WARNING
AMP HTML Link in AMP page?	PASS
AMP HTML Required Markup Issues [0]	PASS
Indexed in Google AMP Cache?	PASS
Response Times (milliseconds)	WARNING
Is access allowed for Googlebots? (robots.txt)	PASS
Robots Meta Tag	PASS
X-Robots-Tag Header	PASS
Redirects via User Agent Googlebot Smartphone	PASS



AMP BY EXAMPLE

Using the Google AMP Cache

Introduction

The Google AMP Cache stores valid AMPs and provides a consistently fast access to AMPs.

AMP Cache URL Format

Loading your AMPs via the Google AMP Cache is easy. The AMP Cache URL is composed based on the source URL's scheme (HTTP or HTTPS):

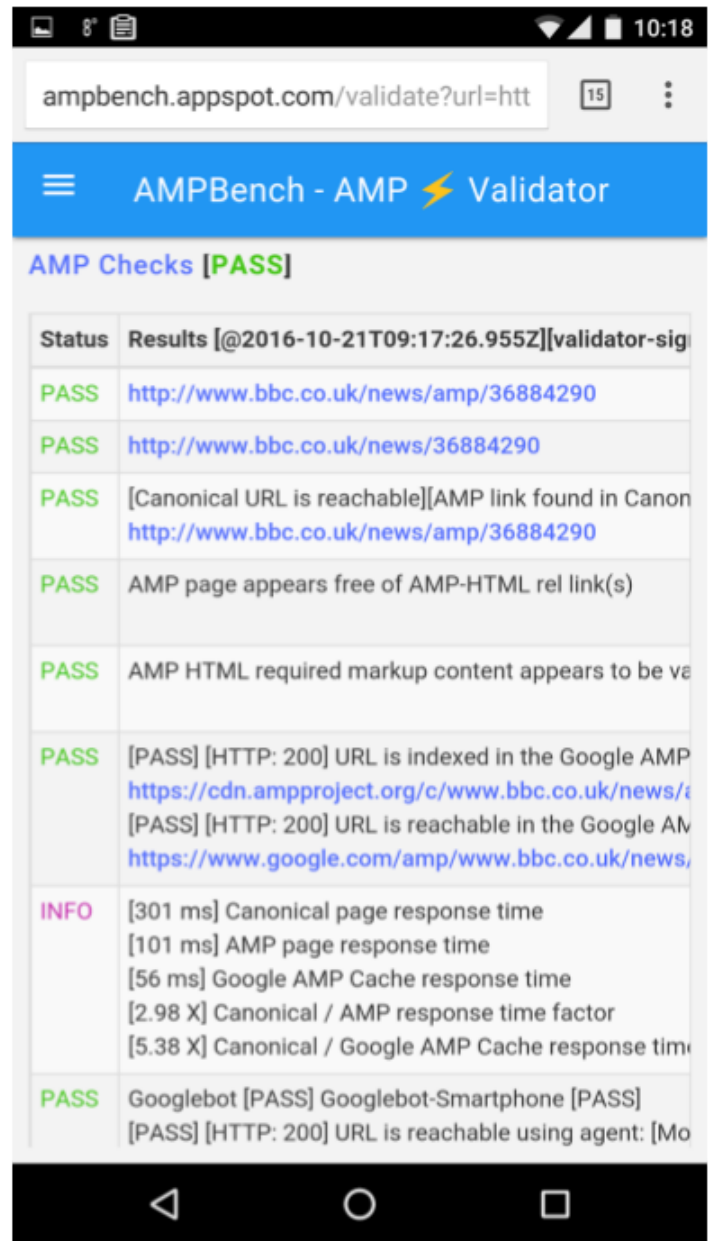
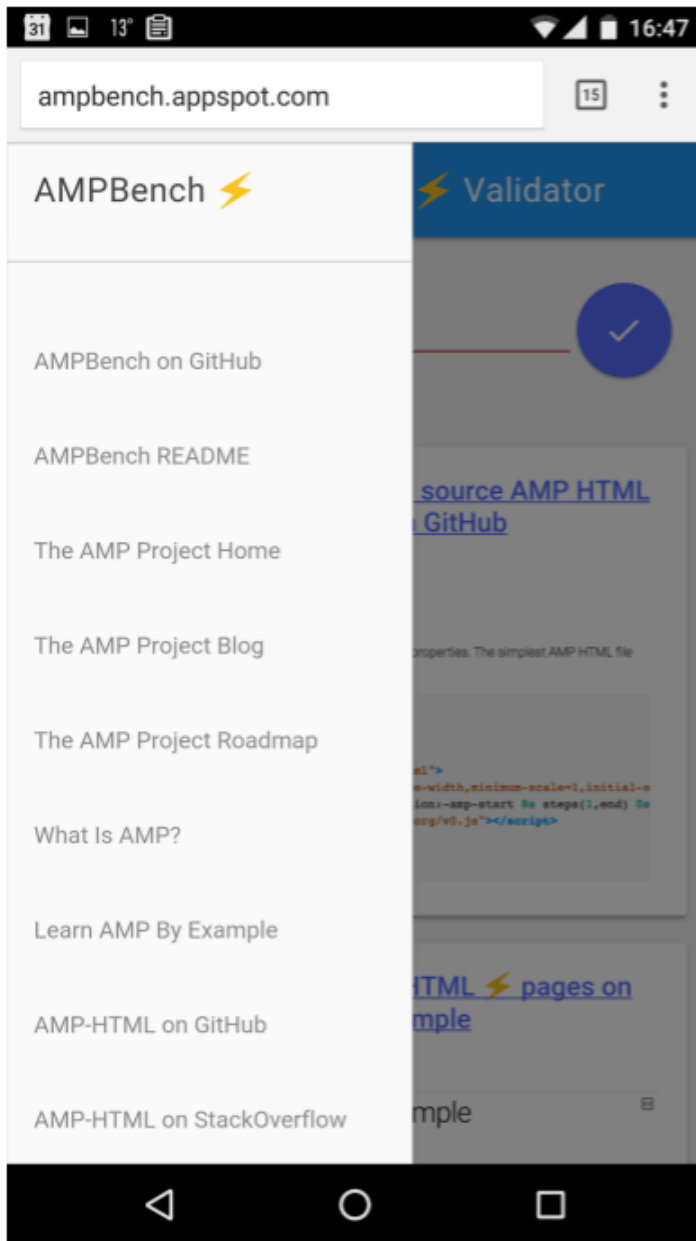
- for HTTP prepend `cdn.ampproject.org/c/`. Example: `https://cdn.ampproject.org/c/www.bbc.co.uk/news/amp/35838735`
- for HTTPS prepend `cdn.ampproject.org/c/s/`. Example: `https://cdn.ampproject.org/c/s/ampbyexample.com`

It is OK for URLs to include parameters in the query string, simply include these in the AMP Cache URL as well.

Example: `https://cdn.ampproject.org/c/s/ampbyexample.com/g?value=Hello%20World`

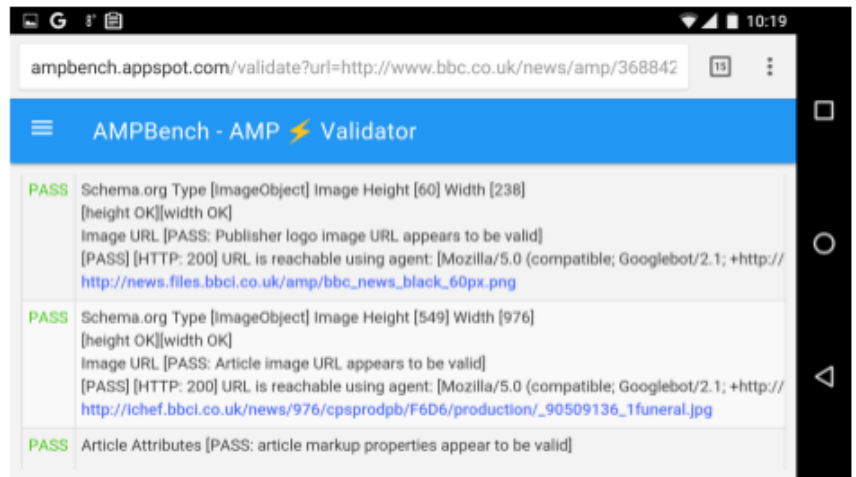
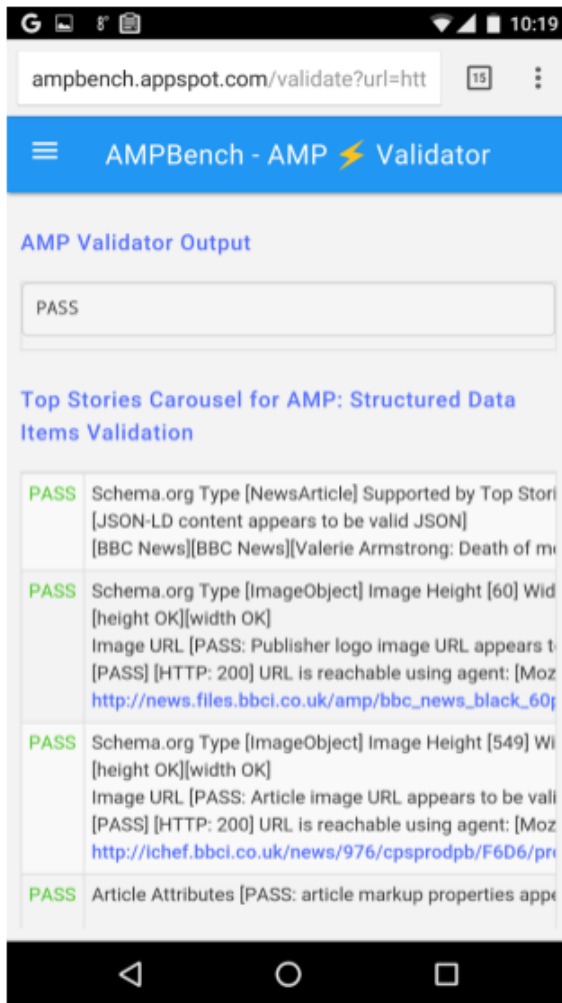
Material Design UI

The user interface of AMPBench is built with [Material Design Lite](#).



Responsive results tables support troubleshooting on the move

In order to allow wide format validation results to be comfortably viewed on mobile devices, AMPBench renders output with [the Bootstrap framework](#) in [responsive, scrollable tables](#). This means all results can be inspected by simply scrolling the viewport horizontally.



Getting the code and running it on your system (or host)

Install [Node.js](#) on your system. This can be done [by downloading](#) or [by using a package manager](#) or [by using NVM](#). Then do the following from a terminal command-line session:

```
$ git clone https://github.com/ampproject/ampbench.git
$ cd ampbench
$ npm update
$ npm run start
```

Or, to watch code changes and automatically restart the server in development mode:

```
$ npm run monitor
```

Then try navigating to these links from your web browser:

- <http://localhost:8080>
- <http://localhost:8080/version>
- <http://localhost:8080/validate?url=https://ampbyexample.com>
- <http://localhost:8080/api?url=https://ampbyexample.com>

And try these from the command-line:

```
$ curl http://localhost:8080/version
$ curl http://localhost:8080/raw?url=https://ampbyexample.com
$ curl http://localhost:8080/api?url=https://ampbyexample.com
$ curl http://localhost:8080/api2?url=https://ampbyexample.com
```

Experimental AMPBench JSON APIs

AMPBench supports some simple, experimental JSON APIs.

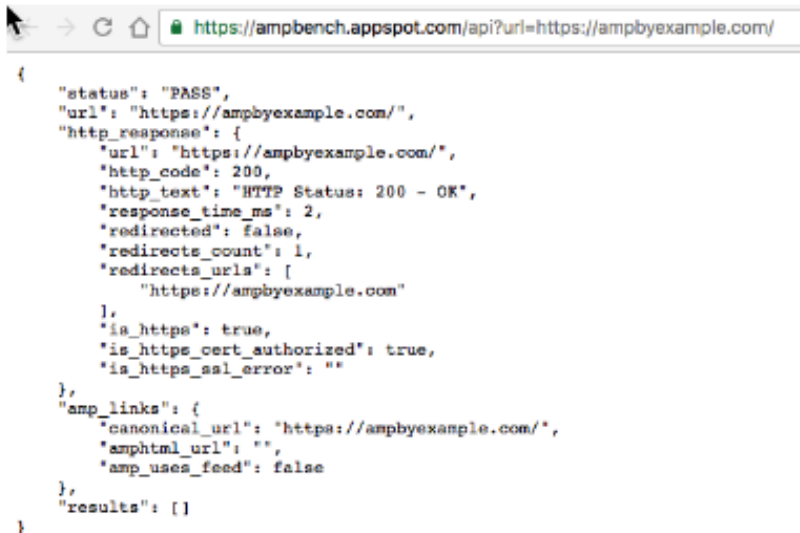
Try these from your web browser:

- <https://ampbench.appspot.com/api?url=https://ampbyexample.com>
- <https://ampbench.appspot.com/api2?url=https://ampbyexample.com>

Or from the command-line:

```
$ curl https://ampbench.appspot.com/api?url=https://ampbyexample.com
$ curl https://ampbench.appspot.com/api2?url=https://ampbyexample.com
```

These should produce output similar to the following:



```
{
  "status": "PASS",
  "url": "https://ampbyexample.com/",
  "http_response": {
    "url": "https://ampbyexample.com/",
    "http_code": 200,
    "http_text": "HTTP Status: 200 - OK",
    "response_time_ms": 2,
    "redirected": false,
    "redirects_count": 1,
    "redirects_urls": [
      "https://ampbyexample.com"
    ],
    "is_https": true,
    "is_https_cert_authorized": true,
    "is_https_ssl_error": ""
  },
  "amp_links": {
    "canonical_url": "https://ampbyexample.com/",
    "amphtml_url": "",
    "amp_uses_feed": false
  },
  "results": []
}
```



```
{
  "status": "PASS",
  "url": "https://ampbyexample.com/",
  "http_response": {
    "url": "https://ampbyexample.com/",
    "http_code": 200,
    "http_text": "HTTP Status: 200 - OK",
    "response_time_ms": 6,
    "redirected": false,
    "redirects_count": 1,
    "redirects_urls": [
      "https://ampbyexample.com"
    ],
    "is_https": true,
    "is_https_cert_authorized": true,
    "is_https_ssl_error": ""
  },
  "amp_links": {
    "canonical_url": "https://ampbyexample.com/",
    "amphtml_url": "",
    "amp_uses_feed": false
  },
  "results": []
}
```

Requesting an AMPHTML Validator reload

The AMPHTML validator is cached at <https://cdn.ampproject.org/v0/validator.js>.

AMPBench loads a locally cached copy of this file into memory using the [amphtml-validator Node.js npm package](#) which supports a [validation API](#).

With the `../command_force_validator_update` command we can ask AMPBench to update the locally cached validator source file from the official cache location and then reload the validator code into memory. It will check if a different version, based on a checksum of the contents, is available before doing this.

It can be invoked like this:

- http://localhost:8080/command_force_validator_update

Or, from a terminal session like this:

```
$ curl http://localhost:8080/command_force_validator_update
```

Utilities

AMPBench includes some useful debug utility commands that can in some cases help with troubleshooting, such as when a full validation fails on a URL by returning unexpected server responses.

The `/debug...` commands attempt to follow fetch requests and display relevant request and response details in a similar spirit to the `curl -I [--head]...` utility.

Use these as follows in the browser:

- <https://ampbench.appspot.com/debug?url=https://ampbyexample.com>

and:

- https://ampbench.appspot.com/debug_curl?url=https://ampbyexample.com

Or, with the command-line compatible `_cli` equivalents, in a terminal session:

```
$ curl https://ampbench.appspot.com/debug_cli?url=https://ampbyexample.com
```

and:

```
$ curl https://ampbench.appspot.com/debug_curl_cli?url=https://ampbyexample.com
```

The `/debug` and `/debug_cli` versions use a smartphone HTTP User Agent, whilst the `/debug_curl` and `/debug_curl_cli` variants use the `curl` (desktop and server-side) User Agent. The applied User Agent string is displayed in the output and can be seen in the resulting HTTP request headers as in the following examples.

For `/debug...`:

```
{"User-Agent":"Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2725.0 Mobile Safari/537.36","host":"ampbyexample.com"}
```

and for `/debug_curl...`:

```
{"User-Agent":"curl/7.43.0","host":"ampbyexample.com"}
```

An example of `/debug_curl_cli` output can be seen here:


```

=> GET: https://allrecipes.com/recipe/11749/chicken-pesto-a-la-lisa/amp/

{"User-Agent": UA_CURL}

-----

=> REQUEST: https://allrecipes.com/recipe/11749/chicken-pesto-a-la-lisa/amp/

{"User-Agent": "curl/7.43.0", "host": "allrecipes.com"}

-----

=> REDIRECT: 302 https://allrecipes.com/recipe/11749/chicken-pesto-a-la-lisa/

{"cache-control": "private", "content-length": "18479", "content-type": "text/html; charset=utf-8", "location": "/recipe/11749/chicken-pesto-a-la-lisa/", "expires": "Sat, 22-Oct-2016 17:31:52 GMT", "path": "/", "HttpOnly": true, "ARSiteUser": "1-94eb5496-fad5-4326-97eb-13d40fa54318", "domain": "allrecipes.com", "pressedSession": "6wAAAB+LCAAAAAABABFjzEOwjAMRe+SmUiUpoWysYA6sCGmLA1lwUKJKztBQoi706pAx++n72erl1UXojtCHTqyavvPh4TtkK3KdGXAFaYqdde0hTb5qt0G6xG8A5Yb9iOZBLtA4ekpybc+TskHjIL/zrrd0cfwwCbC9MNwnfoAn4SDN0sAAAA=", "domain": "allrecipes.com", "path": "/", "HttpOnly": true, "ARToken": "P6k/U+2F1ECWIwEIuLJz4wSO/rUsym4rPrym5KW0yoI7TWk88v3gRQy9xb77ZrSc", "domain": "allrecipes.com", "expires": "Sat, 22-Oct-2016 17:31:52 GMT", "path": "/", "TS01c363ed39eca5b09401b109a11ff48f2f6f96b0d9a39671a86", "Path": "/", "TS01cf248e=0123291da4c681a6a573d83410f74525b92af61569a2efca6cdc34570e700bd39ed11085718ff1ea6a077d979dbdf7035606a4009fe2e5b18b4142c092f326aa4e76b74bd04937be2020da7d69ae03702fc0b62da0", "path": "/", "domain": "allrecipes.com", "isable-session-affinity": "true", "access-control-allow-origin": "http://images.media-allrecipes.com", "date": "Fri, 21 Oct 2016 17:31:51 GMT", "connection": "close", "7f67e16a6b9dc25c43075009e4992086490f16fcc974c5746a4579cc1c87867311f; Path=/"}}

-----

=> REQUEST: https://allrecipes.com/recipe/11749/chicken-pesto-a-la-lisa/

{"User-Agent": "curl/7.43.0", "referer": "https://allrecipes.com/recipe/11749/chicken-pesto-a-la-lisa/amp/", "host": "allrecipes.com"}

-----

=> REDIRECT: 301 http://allrecipes.com/recipe/11749/chicken-pesto-a-la-lisa/

{"content-length": "182", "content-type": "text/html; charset=UTF-8", "location": "http://allrecipes.com/recipe/11749/chicken-pesto-a-la-lisa/", "expires": "Sat, 22-Oct-2016 17:31:52 GMT", "path": "/", "HttpOnly": true, "ARSiteUser": "1-94eb5496-fad5-4326-97eb-13d40fa54318", "domain": "allrecipes.com", "pressedSession": "6wAAAB+LCAAAAAABABFjzEOwjAMRe+SmUiUpoWysYA6sCGmLA1lwUKJKztBQoi706pAx++n72erl1UXojtCHTqyavvPh4TtkK3KdGXAFaYqdde0hTb5qt0G6xG8A5Yb9iOZBLtA4ekpybc+TskHjIL/zrrd0cfwwCbC9MNwnfoAn4SDN0sAAAA=", "domain": "allrecipes.com", "path": "/", "HttpOnly": true, "ARToken": "P6k/U+2F1ECWIwpmI527pRnm1NMCHAKlkHELe8R8BA4At3ZW35N1DEHwBJzh2JWr", "domain": "allrecipes.com", "expires": "Sat, 22-Oct-2016 17:32:03 GMT", "path": "/", "LastViewedRecipes": "11749", "domain": "allrecipes.com", "path": "/", "TS01c363ed39eca5b09401b109a11ff48f2f6f96b0d9a39671a86", "Path": "/", "TS01cf248e=0123291da49c505b8d4c815b748684b778e0d1a3961c3cc8e530681cb7049f24e439308773302998ac9ed56feeb44da075b3b95a283003c6b81affc9bb68eb599f3cbfa44210a8267", "path": "/", "domain": "allrecipes.com", "isable-session-affinity": "true", "access-control-allow-origin": "http://images.media-allrecipes.com", "date": "Fri, 21 Oct 2016 17:32:02 GMT", "connection": "close", "7f67e16a6b9dc25c43075009e4992086490f16fcc974c5746a4579cc1c87867311f; Path=/"}}

-----

=> RESPONSE: 200

{"cache-control": "private", "content-length": "287093", "content-type": "text/html; charset=utf-8", "set-cookie": ["FirstImpression=False; path=/; HttpOnly", "ARSiteUser=1-db5354c6-15e1-409e-a797-656b15fc8af8; domain=allrecipes.com; expires=Fri, 20-Oct-2017 20:55:49 GMT; path=/; HttpOnly", "ARToken=P6k/U+2F1ECWIwpmI527pRnm1NMCHAKlkHELe8R8BA4At3ZW35N1DEHwBJzh2JWr", "domain=allrecipes.com; expires=Sat, 22-Oct-2016 17:32:03 GMT; path=/; HttpOnly", "LastViewedRecipes=11749; domain=allrecipes.com; path=/; HttpOnly", "TS01c363ed39eca5b09401b109a11ff48f2f6f96b0d9a39671a86", "Path=/; HttpOnly", "TS01cf248e=0123291da49c505b8d4c815b748684b778e0d1a3961c3cc8e530681cb7049f24e439308773302998ac9ed56feeb44da075b3b95a283003c6b81affc9bb68eb599f3cbfa44210a8267", "path=/; domain=allrecipes.com"], "x-aspnetmvc-version": "5.0", "x-powered-by": "ASP.NET", "arr-disable-session-affinity": "true", "access-control-allow-origin": "http://images.media-allrecipes.com", "date": "Fri, 21 Oct 2016 17:32:02 GMT", "connection": "close", "7f67e16a6b9dc25c43075009e4992086490f16fcc974c5746a4579cc1c87867311f; Path=/"}}

-----

```

AMPBench is open for contributions

Like the [AMP Project as a whole](#), [AMPBench](#) welcomes [GitHub Issues](#) and [Pull Requests](#), or [any other contributions](#) the community deems useful.