



Visual regression testing: BackstopJS for the win

Marji Cermak

DevOps Engineer, Co-founder

Marji Cermak



DevOps Engineer, Co-founder – Morpht

Drupal: marji

X: cermakm

Web: www.morpht.com



Visual regression testing

is a method of validating that changes made to an application do not negatively affect the visual appearance of the application.

Visual regression QUIZ

Can you spot any differences in the next two screenshots?

Hint: There are many. Try to flip back and forward between the next two slides.

Test page with no components

This is a test page.

Vivamus suscipit tortor eget felis porttitor volutpat. Praesent sapien massa, convallis a pellentesque nec, egestas non nisi. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec velit neque, auctor sit amet aliquam vel, ullamcorper sit amet ligula. Donec sollicitudin molestie malesuada. Proin eget tortor risus. Donec sollicitudin molestie malesuada.

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec velit neque, auctor sit amet aliquam vel, ullamcorper sit amet ligula. Sed porttitor lectus nibh. Donec sollicitudin molestie malesuada. Proin eget tortor risus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum congue leo eget malesuada.

Curabitur arcu erat, accumsan id imperdiet et, porttitor at sem. Praesent sapien massa, convallis a pellentesque nec, egestas non nisi. Nulla porttitor accumsan tincidunt. Cras ultricies ligula sed magna dictum porta. Quisque velit nisi, pretium ut lacinia in, elementum id enim. Proin eget tortor risus.

Page history

Last updated: 6 April 2021

Test

View modes

Test page

Abbreviation and Technical terms

Components

Test page with no components

Typography

Test Funnelback search results

Test page with no components

This is a test page.

Vivamus suscipit tortor eget felis porttitor volutpat. Praesent sapien massa, convallis a pellentesque nec, egestas non nisi. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec velit neque, auctor sit amet aliquam vel, ullamcorper sit amet ligula. Donec sollicitudin molestie malesuada. Proin eget tortor risus. Donec sollicitudin molestie malesuada.

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec velit neque, auctor sit amet aliquam vel, ullamcorper sit amet ligula. Sed porttitor lectus nibh. Donec sollicitudin molestie malesuada. Proin eget tortor risus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec rutrum congue leo eget malesuada.

Curabitur arcu erat, accumsan id imperdiet et, porttitor at sem. Praesent sapien massa, convallis a pellentesque nec, egestas non nisi. Nulla porttitor accumsan tincidunt. Cras ultricies ligula sed magna dictum porta. Quisque velit nisi, pretium ut lacinia in, elementum id enim. Proin eget tortor risus.

Page history

Last updated: 6 April 2021

Test

[View modes](#)

[Test page](#)

[Abbreviation and Technical terms](#)

[Components](#)

[Test page with no components](#)

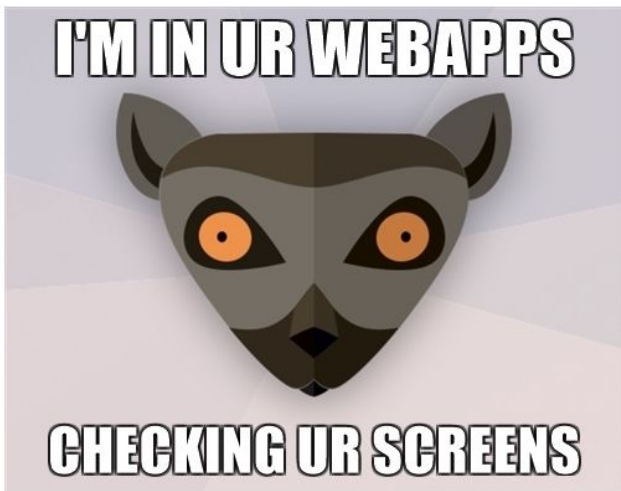
[Typography](#)

[Test Funnelback search results](#)

README Contributing MIT license

npm@latest v6.3.25 Stars 7k contributors 157 downloads 2.3M/year Backstop Reference Test no status

BackstopJS

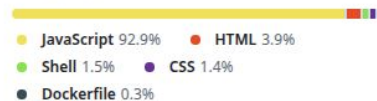


BackstopJS automates visual regression testing of your webapp – comparing screenshots over time.

News

Backstop 6.3.2 -- now with support for Node 20 Many thanks to @dgrebb for this doozy of a maintenance release!

Languages



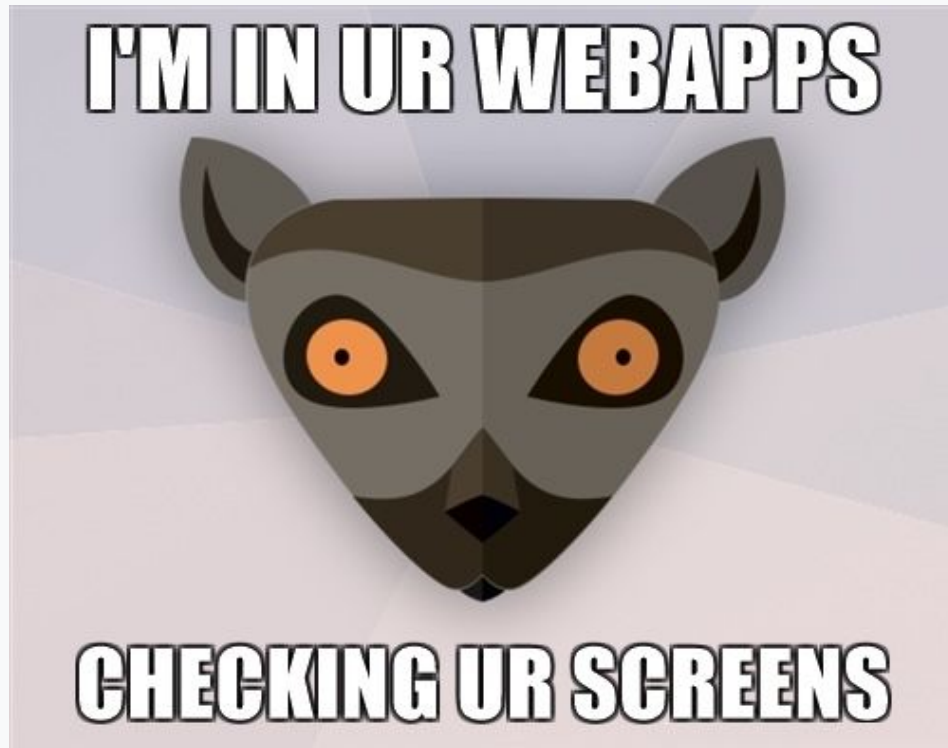
BackstopJS

BackstopJS is an open-source visual regression testing tool.

Screenshot comparisons.

Generates reports showing “before” vs “after” screenshots, highlighting the differences.

Does it look right?



BackstopJS

BackstopJS is an open-source visual regression testing tool.

Screenshot comparisons.

Generates reports showing “before” vs “after” screenshots, highlighting the differences.

Can you spot a difference?



First test ride (1)

Get BackstopJS running

Installation

Global Installation

```
$ npm install -g backstopjs
```

Installation

No need, use a docker image

```
$ docker run --rm --user $(id -u) --shm-size 2048m --add-host="www.google-analytics.com:127.0.0.1"  
--add-host="google-analytics.com:127.0.0.1" -v "$(pwd)":/src backstopjs/backstopjs:6.3.25
```

Installation

No need, use a docker image and define an alias

```
$ docker run --rm --user $(id -u) --shm-size 2048m --add-host="www.google-analytics.com:127.0.0.1"  
--add-host="google-analytics.com:127.0.0.1" -v "$(pwd)":/src backstopjs/backstopjs:6.3.25
```

```
$ alias backstop='docker run --rm --user $(id -u) --shm-size 2048m  
--add-host="www.google-analytics.com:127.0.0.1" --add-host="google-analytics.com:127.0.0.1" -v "$(pwd)":/src  
backstopjs/backstopjs:6.3.25"$@"'
```

Let's run the first test

Simple

```
$ backstop init  
$ backstop reference  
$ backstop test
```

BackstopJS and lagoon / ahoy

Easy to define: [custom/ahoy.yml](#)

```
---
ahoyapi: v2
usage: Project specific ahoy commands.

commands:

  backstop:
    usage: Run a backstopJS container
    cmd: |
      docker run --rm --user $(id -u) --shm-size 2048m --add-host=myproject.docker.amazee.io:host-gateway
      --add-host="www.google-analytics.com:127.0.0.1" --add-host="google-analytics.com:127.0.0.1" -v
      "$(pwd)/tests/backstop":/src backstopjs/backstopjs:6.3.25 "$@" --config=./backstop.config.js
```

BackstopJS and lagoon / ahoy

Have we just integrated backstop to the project? Yes! :-)

```
$ ahoy my backstop
```

```
$ ddev backstop
```

```
$lando backstop
```




Second test ride (2)

reference vs approve

The BackstopJS Workflow

Two ways with reference site

```
$ backstop init  
$ backstop reference  
$ backstop test
```

```
"scenarios": [  
  {  
    "label": "My Homepage",  
    "url": "https://dev.my.page.io/",  
    "referenceUrl": "https://my.page.io/",
```

```
$ backstop init  
$ backstop test  
$ backstop approve
```

```
"scenarios": [  
  {  
    "label": "My Homepage",  
    "url": "https://my.page.io/",  
    "referenceUrl": "",
```

Repeatedly test same site

Current prod is the “reference” state.

New deployment to prod happens.

Prod gets tested, change is found - alarm raised. Either:

Approved - current prod becomes new “reference”. Or fix issue.



Configuring test ride (3)

Scenarios and their properties

Config file

Can be specified (`--config="backstop.json"`)

Can be JS based (`--config="backstop.js"`)

Can be generated (drupal/backstop_generator)

Config file

Scenario properties

- misMatchThreshold (percentage of different pixels allowed to pass) the test)
- hideSelectors (array of selectors set to **visibility: hidden**)
- removeSelectors (array of selectors set to **display: none**)

Config file

Scenario properties

- delay (wait for x milliseconds)
- readyEvent (wait until this string has been logged to the console)
- readySelector (wait until this selector exists before continuing)
- onBeforeScript (used to set up browser state e.g. cookies)



Sophisticated test rides (4)

Various.

Backstop_generator

Drupal module (D11 compatible)

```
$ composer require drupal/backstop_generator  
$ drush en backstop_generator -y
```

https://www.drupal.org/project/backstop_generator

BackstopJS on CI

Periodically test prod - catch a rogue deployment or an edit.

Compare stage and prod (triggered by deployment to stage)

Run on demand against two sites

BackstopJS and logged in

Fill in: backstop_data/engine_scripts/cookies.json

```
[
  {
    "domain": ".govcms.morpht.io",
    "path": "/",
    "name": "SSESSbdf68df33368b928530d13b7716fa9",
    "value": "VGGVG4-RcrtVhp8Ef8Pnj1XX9o1Ur4Xpdfdxx000FJf7-Xme",
    "httpOnly": false,
    "secure": true,
    "sameSite": "Lax"
  }
]
```

Host mapping

myproject.docker.amazee.io or myproject.Indo.site resolve to 127.0.0.1 - beware

```
---
ahoyapi: v2
usage: Project specific ahoy commands.

commands:

  backstop:
    usage: Run a backstopJS container
    cmd: |
      docker run --rm --user $(id -u) --shm-size 2048m --add-host=myproject.docker.amazee.io:host-gateway
      --add-host="www.google-analytics.com:127.0.0.1" --add-host="google-analytics.com:127.0.0.1" -v
      "$(pwd)/tests/backstop":/src backstopjs/backstopjs:6.3.25 "$@" --config=./backstop.config.js
```

Testing via socks proxy

Accessing IP Whitelisted site via sock5 proxy

```
engineOptions: {  
  args: [  
    // To use a socks5 proxy to make test go via a whitelisted IP:  
    // socks5 proxy must be on an address reachable from backstop container  
    '--proxy-server=socks5://172.19.0.1:1080',  
    '--proxy-bypass-list=my.local.site',  
  ],  
},
```


BackstopJS blocked

CDN on prod might block tests.

Make a feature environment from prod and test there.

Use a proxy / whitelisted IP if possible.



Good bye