

OWASP Juice Shop

Achieving sustainability for open source projects



8th to 12th
of May
2017

Waterfront
Conference
Center

OWASP
AppSec
Belgium

https://www.owasp.org/index.php/OWASP_Juice_Shop_Project

Presentation by Björn Kimminich / @bkimminich

Flattr

0

Like 49

Tweet

Follow @owasp_juiceshop

Follow @bkimminich

Follow @bkimminich

148

Star

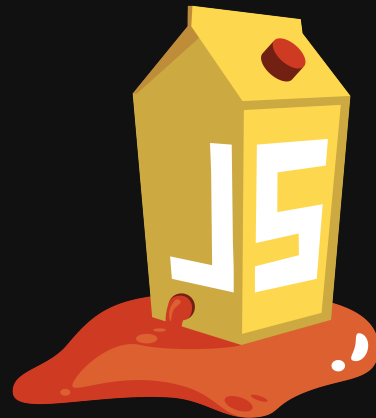
335

Chapter One

WAT?

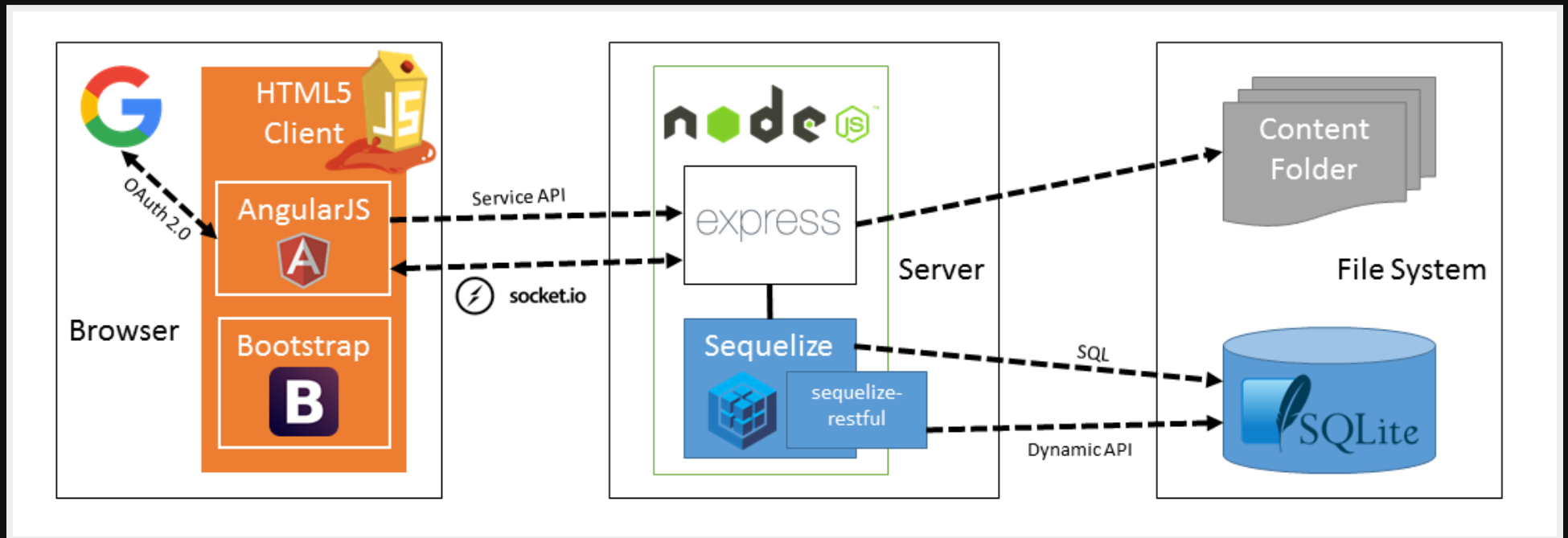
OWASP Juice Shop...

...is an intentionally insecure webapp written in Node.js, Express and AngularJS. It contains 38+ hacking challenges of varying difficulty tracked on a score board.



Modern Web-Architecture

Javascript all the way from UI to REST API



OWASP Juice Shop CTF...

...is a commandline tool written in Node.js and published on NPM.js. It helps you to set up and host CTF events on the [CTFd](#) platform with the Juice Shop challenges.



```
<!-- uncomment this and the following slide **ONLY** if...
```



```
<!-- ...Juice Shop got promoted to Lab Project
during the Project Summit prior to this AppSecEU! :-D
```



The next OWASP Connector should make this official! (But you heard it here first!)



Chapter Two

Open Source Antipatterns

Barren README

An empty or lackluster front-page radiates the impression that nobody takes serious care of the project.

Bad Example

OWASP Juice Shop

OWASP Juice Shop is an intentionally insecure web app for security trainings written entirely in Javascript which encompasses the entire OWASP Top Ten and other severe security flaws.

Setup

```
npm start
```




or

```
docker run -d -p 3000:3000 bkimminich/juice-shop
```

Copyright © by Bjoern Kimminich 2014-2017.

Worse Example

```
1 lines (1 sloc) | 12 Bytes
```

Raw Blame History   

```
juice-shop
```

Worst Example

Add a README with an overview of your project.

Add a README


Good Example

The screenshot shows the GitHub repository for OWASP Juice Shop. It includes the repository name, description, and a list of main selling points. The description states that the application contains 30 challenges of varying difficulty. The main selling points include: easy-to-install, self-contained, self-healing, gamification, CTF support, and re-branding. Below the text is a diagram of the application architecture showing the flow from a browser to a server and then to a database. The page also features a 'Preview' section with a 'Deploy on Heroku' button and a 'Setup' section with instructions for running the application locally.

This screenshot displays the Docker installation instructions for OWASP Juice Shop. It provides a step-by-step guide for installing and running the application using Docker. The instructions include: 1. Installing Docker, 2. Searching for the image, 3. Clicking on the Open icon, 4. Installing a Docker daemon on Windows, 5. Downloading the image, 6. Unpacking the image, 7. Running the application, 8. Accessing the application. It also includes a section for 'Packaged Distributions' and 'Amazon EC2 Instance' with specific commands for running the application on AWS. A table lists various Docker images and their corresponding packaged distributions. The page concludes with a 'Slide Deck' section containing links to an introduction slide and the full slide deck.


This screenshot shows the 'Troubleshooting' and 'Contributing' sections of the OWASP Juice Shop documentation. The 'Troubleshooting' section provides instructions on how to seek help, including a link to the GitHub issue tracker. The 'Contributing' section outlines the process of contributing to the project, including how to create a pull request, how to write a blog post, and how to create a new challenge. It also includes a 'References' section with links to related resources and a 'References' section with links to related resources. The page features a large, stylized illustration of a yellow juice carton with the number '15' on it, set against a red background.

Better(?) Example

find packages Greetings, bkimminich

★ juice-shop-ctf-cli public

Command line client to generate INSERT statements for CTFd with the OWASP Juice Shop challenges



OWASP Juice Shop CTF

owasp incubator release v0.4.0

Follow 161

build passing coverage 97% code climate 4.0 bitHound 98 dependencies up-to-date

The NPM package `juice-shop-ctf-cli` lets you create a list of INSERT statements for the CTFd database that will populate the platform for a Capture the Flag event using OWASP Juice Shop.

```
c:\Data>juice-shop-ctf
Generate INSERT statements for CTFd with the OWASP Juice Shop challenges
Juice Shop URL to retrieve challenges? https://juice-shop.herokuapp.com
Secret key corp URL to ctf.key file? https://raw.githubusercontent.com/bkimminich/juice-shop/master/ctf.key
DELETE all CTFd Challenges before INSERT statements? Yes
SELECT all CTFd Challenges after INSERT statements? Yes

SQL written to C:\Data\insert-ctfd-challenges.sql

For a step-by-step guide to apply the INSERT statements to CTFd, please refer to
https://github.com/bkimminich/juice-shop-ctf#populating-the-ctfd-database
```

Installation

downloads 122/month downloads 122

```
npm i -g juice-shop-ctf-cli
```

Unleash awesomeness

Private packages, team management tools, and powerful integrations. [Get started with npm Orgs](#)

npm i juice-shop-ctf-cli

how? learn more

bkimminich published 3 days ago

0.4.0 is the latest of 4 releases

github.com/bkimminich/juice-shop-ctf

owasp.org/index.php/OWASP_Juice_Shop_Pro...

MIT

Collaborators

edit

Stats

3 downloads in the last day

44 downloads in the last week

122 downloads in the last month

Usage

Open a command line and run:

```
juice-shop-ctf
```

Then simply follow the instructions of the command line tool.

Populating the CTFd database

Apply the generated `INSERT-ctfd-challenges.sql` following the steps describing your own CTFd setup.

Default setup (including SQLite database)

- Get CTFd with `git clone https://github.com/CTFd/CTFd.git`.
- Perform steps 1 and 3 from the CTFd installation instructions.
- Use your favourite SQLite client to connect to the CTFd database and execute the INSERT statements you created.
- Browse to your CTFd instance (i.e. by default `https://127.0.0.1:8000`) and create an admin user and CTF name.

docker-compose setup (including MySQL container)

- Setup Docker host and Docker compose.
- Follow step 2.4 from the CTFd Docker setup to download the source code, create containers and start them.
- After running `docker-compose up` from previous step, you should be able to browse to your CTFd instance (i.e. `localhost:8000` by default) and create an admin user and CTF name.
- Once you have done this, run `docker-compose down` or use CTRL-C to shut down CTFd. Note: Unlike a usual Docker container, data will persist even afterwards.
- Add the following section to the `docker-compose.yml` file and then run `docker-compose up` again:

```
ports:
  - "3306:3306"
```

- Use your favourite MySQL client to connect to the CTFd database (default credentials are root with no password) and execute the INSERT statements you created.
- Browse back to your CTFd instance (i.e. and check everything has worked correctly).
- If everything has worked, do another `docker-compose down`, remove the ports section you added to `docker-compose.yml`, and then do `docker-compose up` again and you are ready to go!

Other setups (PostgreSQL or MySQL)

- Perform any of the CTFd database setups.
- Launch your CTFd instance and perform the remaining setup similar to the SQLite default setup.

Troubleshooting

If you need help with the application setup please check the Troubleshooting section below or post your specific problem or question in the official Github chat.

- If using Docker/Toolbox on Windows make sure that you also enable port forwarding for all required ports from Host 127.0.0.1:XXXX to 0.0.0.0:XXXX for TCP in the default VM's network adapter in VirtualBox. For CTFd you need ports 8000 permanently and 3306 during setup.

Contributing

Found a bug? Got an idea for enhancement? Improvement for cheating prevention? Feel free to create an issue or post your ideas in the chat. Pull requests are also highly welcome - please refer to [CONTRIBUTING.md](#) for details.

Donations

PayPal

PayPal donations via above button go to the OWASP Foundations and are earmarked for "Juice Shop". This is the preferred way to support the project.

Others

bitHound


Contributors

Ordered by date of first contribution. Auto-generated on Wed, 02 Feb 2017 20:02:38 GMT.

- Björn Kimminich aka [b3kimminich](#)
- Josh Grossman aka [jgthoth](#)

Licensing

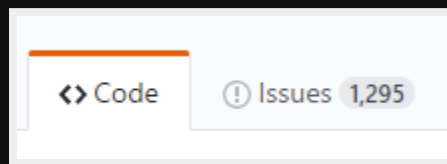
This program is free software: you can redistribute it and/or modify it under the terms of the MIT license. OWASP Juice Shop and any contributions are Copyright © by Björn Kimminich 2016-2017.



Pile of Issues

Open issues pile up in the ticketing system, most of them unanswered and unattended.

A pile of unattended issues



Really a 🍌-pile of unattended issues

The screenshot displays a GitHub Issues page with a list of seven issues. At the top, it shows '1,295 Open' and '6,151 Closed' issues. The issues are sorted by 'Oldest' as indicated by the dropdown menu. The issues are as follows:

- Issue #254: opened on 10 Sep 2011, labeled 'feature-request' and 'notifications'.
- Issue #483: opened on 28 Mar 2012, labeled 'bug'.
- Issue #532: opened on 8 May 2012, labeled 'feature-request'.
- Issue #761: opened on 14 Nov 2012, labeled 'feature-request'.
- Issue #866: opened on 21 Jan 2013, labeled 'bug'.
- Issue #887: opened on 29 Jan 2013, labeled 'feature-request'.
- Issue #909: opened on 4 Feb 2013, labeled 'feature-request' and 'notifications'.

The 'Sort by' dropdown menu is open, showing the following options:

- Newest
- Oldest
- Most commented
- Least commented
- Recently updated
- Least recently updated
- Most reactions

At the bottom of the dropdown menu, there are reaction icons: thumbs up, thumbs down, smiley face, ice cream cone, frowny face, and heart.

Counter: Use **labels** properly

The screenshot displays a Jira issue list with the following items:

- 25 Open** ✓ 138 Closed
- Author ▾ Labels ▾ Milestones ▾ Assignee ▾ Sort ▾
- Crashes if invalid json is posted** 0 - Backlog **blocked** **bug** 1
#131 opened on 2 Nov 2015 by mschwartau
- Crashes if posting with x-www-form-urlencoded** 0 - Backlog **blocked** **bug** 4
#152 opened on 14 Jan 2016 by volkert
- Migration to latest test framework versions** 0 - Backlog **blocked** **testing** 1
#164 opened on 27 Jul 2016 by bkimminich TechStack Update
- Migration to Angular 2** 0 - Backlog **help wanted** 1
#165 opened on 27 Jul 2016 by bkimminich TechStack Update
- Migration to latest Sequelize** 0 - Backlog **help wanted** **technical debt** 3
#167 opened on 28 Jul 2016 by bkimminich TechStack Update
- Add Server-side JS Injection flaw** 0 - Backlog **challenge** 1
#175 opened on 11 Aug 2016 by bkimminich Challenge Pack 2...
- Release files from CI-process not part of SourceForge releases** 0 - Backlog **blocked** **bug** 2
continuous delivery
#179 opened on 25 Aug 2016 by bkimminich
- Add another flavor of DOM-based XSS** 0 - Backlog **challenge** 1
#217 opened on 11 Oct 2016 by bkimminich Challenge Pack 2...
- Exploit lack of proper default \$translateSanitization** 0 - Backlog **challenge** 1
#218 opened on 11 Oct 2016 by bkimminich Challenge Pack 2...
- Add mutation testing for server-side tests** 0 - Backlog **help wanted** **question** **testing** 1
#219 opened on 12 Oct 2016 by bkimminich
- Add socket.io-based challenge** 0 - Backlog **challenge** 2
#220 opened on 12 Oct 2016 by bkimminich Challenge Pack 2...

Agile Counter: Kanban Board

The screenshot displays a Kanban board for the 'juice-shop' project. The board is organized into four columns: Backlog, Ready, Working, and Done. Each column contains a list of tasks with their respective IDs, progress bars, and status indicators.

Backlog:

- #131: Crashes if invalid json is posted. 1 comment. **Marked as blocked**.
- #152: Crashes if posting with x-www-form-urlencoded. 4 comments. **Marked as blocked**.
- #164: Migration to latest test framework versions. 1 comment. **Marked as blocked**.
- #175: Add Server-side JS Injection flaw.
- #165: Migration to Angular 2.
- #167: Migration to latest Sequelize. 3 comments.
- #179: Release files from CI-process not part of SourceForge releases. 2 comments. **Marked as blocked**.

Ready:

- #294: Migrate to Angular 1.6. 0 comments. #165.

Working:

- #305: Show vulnerability page/location/tipp in scoreboard. 9 comments.
- #309: Feature: add auto-save for solved challenges. 4 comments. #307.

Done:

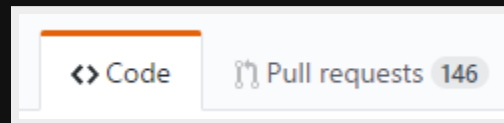
- #277: Customization of the juice shop. 8 comments. **Closed**. Archive.
- #301: Docker crashed when using SQLi on sqlite_master table. 7 comments. **Closed**. Archive.
- #302: After docker crashed there's no way to restore challenges. 2 comments. juice-shop. 7 #301. **Closed**. Archive.
- #303: Corrected contribution links. 1 comment. **Closed**. Archive.
- v3.0.0 #304: **Closed**. Archive.
- #306: Add "2 Hour Hacking: Juice Shop" in LA. **Closed**. Archive.
- #307: Feature: Add refresh button to score-board. 4 comments.

<https://huboard.com/bkimminich/juice-shop>

PR Disaster

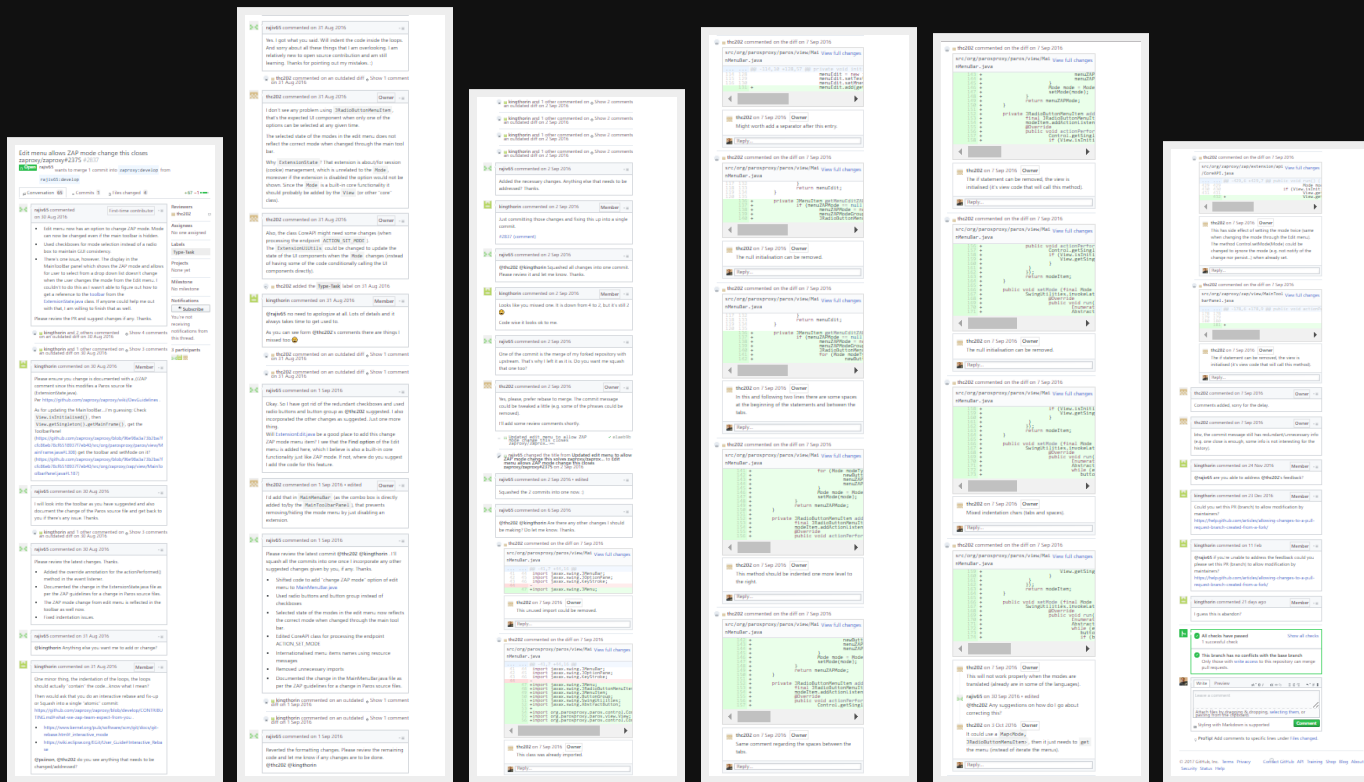
Pull Requests are routinely ignored or flogged to death.

Ignored Pull Requests



Example for a PR flogged to death*

<https://github.com/zaproxy/zaproxy/pull/2837>



*Disclaimer: This example should **by no means** imply that the ZAP team is routinely flogging PRs to death! On the contrary, they offer a helping hand to even those who totally ignore their **written contribution requirements!**

Good Counter: Contribution Guide (ZAP)

Guidelines for Pull Request (PR) submission and processing:

What should you, the author of a pull request, expect from us (ZAP Team)?

- How much time (maximum) until the first feedback? 1 week.
- And following iterations? 1 week.
- This is a guideline we should normally be able to hit. If it's been more than a week and you haven't heard then please feel free to add a comment to your PR and @ mention the team (@zaproxxy/team-zaproxxy).

What we (ZAP Team), expect from you?

- "Atomic commits" (logical changes to be in a single commit). Please don't group disjointed changes into a single commit/PR.
- Descriptive commits (subject and message):
- For example: <https://github.com/spring-projects/spring-framework/blob/master/CONTRIBUTING.md#format-commit-messages>
- Discussion about the changes:
- Should be done in/on the PR or via the Dev Group and a link to that Dev Group thread added to the PR comments. (i.e.: Shared information is important, if something happens via IRC or private email please ensure a summary makes it to the PR.)
- Discussion will be kept in the pull request unless off topic.
- No merge commits. Please, rebase.
- Rebase if the branch has conflicts.
- How much time will a pull request be left open?
- This isn't static, one or more members of the ZAP Team will reach out (using @ mentions in PR comments) once or twice in order to get things back on track. If no input is received after a month or two then the PR will be closed. Total stale time will likely be 2 to 3 months.
- Close with a message such as: "The pull request was closed because of lack of activity (as per CONTRIBUTING guidelines)". Labeled as "Stale".
- If the contribution is deemed important or still valuable the code may be:
- Manually merged (if possible).
- Retrieved by another member of the team, fixed up and resubmitted. In which case the commit message (PR message) should contain a reference to the original submission.

<https://github.com/zaproxxy/zaproxxy/blob/develop/CONTRIBUTING.md#good-counter-guidelines-for-pull-request-pr-submission-and-processing>

Contribution Guide (Juice Shop)

Git-Flow

This repository is maintained in a simplified [Git-Flow](#) fashion: All active development happens on the `develop` branch while `master` is used to deploy stable versions to the [Heroku demo instance](#) and later create tagged releases from.

Pull Requests

Using Git-Flow means that PRs have the highest chance of getting accepted and merged when you open them on the `develop` branch of your fork. That allows for some post-merge changes by the team without directly compromising the `master` branch, which is supposed to hold always be in a release-ready state.

Unit & Integration Tests

There is a full suite containing

- independent unit tests for the client-side code
- integration tests for the server-side API

These tests verify if the normal use cases of the application work. All server-side vulnerabilities are also tested.

```
npm test
```

JavaScript Standard Style Guide

Since v2.7.0 the `npm test` script verifies code compliance with the `standard` style before running the tests. If PRs deviate from this coding style, they will now immediately fail their build and will not be merged until compliant.



In case your PR is failing from style guide issues try running `standard --fix` over your code - this will fix all syntax or code style issues automatically without breaking your code. You might need to `npm i -g standard` first.

End-to-end Tests

The e2e test suite verifies if all client- and server-side vulnerabilities are exploitable. It passes only when all challenges are solvable on the score board.

<https://github.com/bkimminich/juice-shop/blob/master/CONTRIBUTING.md>

Auto Counter: -enforced Coding Style

JavaScript Standard Style Guide

Since v2.7.0 the `npm test` script verifies code compliance with the `standard` style before running the tests. If PRs deviate from this coding style, they will now immediately fail their build and will not be merged until compliant.



JS Standard
Code Style

In case your PR is failing from style guide issues try running `standard --fix` over your code - this will fix all syntax or code style issues automatically without breaking your code. You might need to `npm i -g standard` first.

<http://standardjs.com>

Solecistic Versioning

The **Semantic Versioning** syntax (MAJOR.MINOR.PATCH) is used while its semantics are not adhered to.

Bad Example: AngularJS 1.x

Migrating an App to a newer version

 [Improve this Doc](#)

Minor version releases in AngularJS introduce several breaking changes that may require changes to your application's source code; for instance from 1.0 to 1.2 and from 1.2 to 1.3.

Although we try to avoid breaking changes, there are some cases where it is unavoidable:

- AngularJS has undergone thorough security reviews to make applications safer by default, which drives many of these changes.
- Several new features, especially animations, would not be possible without a few changes.
- Finally, some outstanding bugs were best fixed by changing an existing API.

Contents

[Migrating from 1.5 to 1.6](#)

[Migrating from 1.4 to 1.5](#)

[Migrating from 1.3 to 1.4](#)

[Migrating from 1.2 to 1.3](#)

[Migrating from 1.0 to 1.2](#)

Good Example: AngularJS 2+

Starting with the 2.0.0 release of Angular, we've adopted the following development processes:

- We use semantic versioning for signaling the content of Angular releases.
- We have moved to time-based release cycles so that you can plan ahead.
- We have a deprecation policy so that you know how to get notified of API changes ahead of time.
- We have clarified the distinction between stable and experimental APIs.
- We have clarified the scope of our Public API surface.

Semantic Versioning

SemVer means that our version numbers are meaningful. Patch releases will not change the functionality, minor releases will contain only additive changes, and breaking changes are reserved for major releases.

Juice Shop takes it serious


For a tiny incompatible configuration-file change Juice Shop went from 2.26.0 to 3.0.0

v3.0.0


78eee33


Verified

v3.0.0 Edit

 bkimminich released this 10 days ago · 3 commits to master since this release

Incompatible changes

-  simplified and extended configuration mechanism (see [CUSTOMIZATION.md](#))
 - properties `application.logo`, `application.favicon` and all `products.image` now handle files and URLs
 - removed obsolete properties for logo, favicon and product image URLs



 Custom YAML configuration files from v2.x might not work with v3.0.0! Refer to [CUSTOMIZATION.md](#) to migrate to latest configuration specs. For a `yourconfig.yml` run `NODE_ENV=yourconfig npm run protractor` to verify all challenges will work in customized mode.

Major Zero

Not even the original author thinks that the project is mature enough to release a 1.x version from it.

Example: z85-cli

Ninja Power Manifesto npm Enterprise features pricing documentation support

 find packages sign up or log in 

★ **z85-cli** public
Command line client for ZeroMQ Base-85 encoding

Getting Started

Install the module with:

```
npm install -g z85-cli
```

Documentation

Encoding

```
z85 --encode [-e] <value>
```

Decoding

```
z85 --decode [-d] <value>
```


Specification


Please refer to [32/Z85 - ZeroMQ Base-85 Encoding Algorithm](#).

License

Copyright (c) 2014-2016 Bjoern Kimminich Licensed under the MIT license.


We ♥ your boss
Selectively mirror the npm registry inside your firewall. Filter packages based on security, licensing, code quality and more. Build awesome stuff faster.
[Try npm Enterprise for free...](#)

 `npm install -g z85-cli`
[how? learn more](#)


 **bkimminich** published 8 months ago

0.1.9 is the latest of 9 releases

github.com/bkimminich/z85-cli

MIT 

Collaborators list



Stats

0 downloads in the last day

9 downloads in the last week

44 downloads in the last month

[One open issue](#) on GitHub

[No open pull requests](#) on GitHub

[Try it out](#)

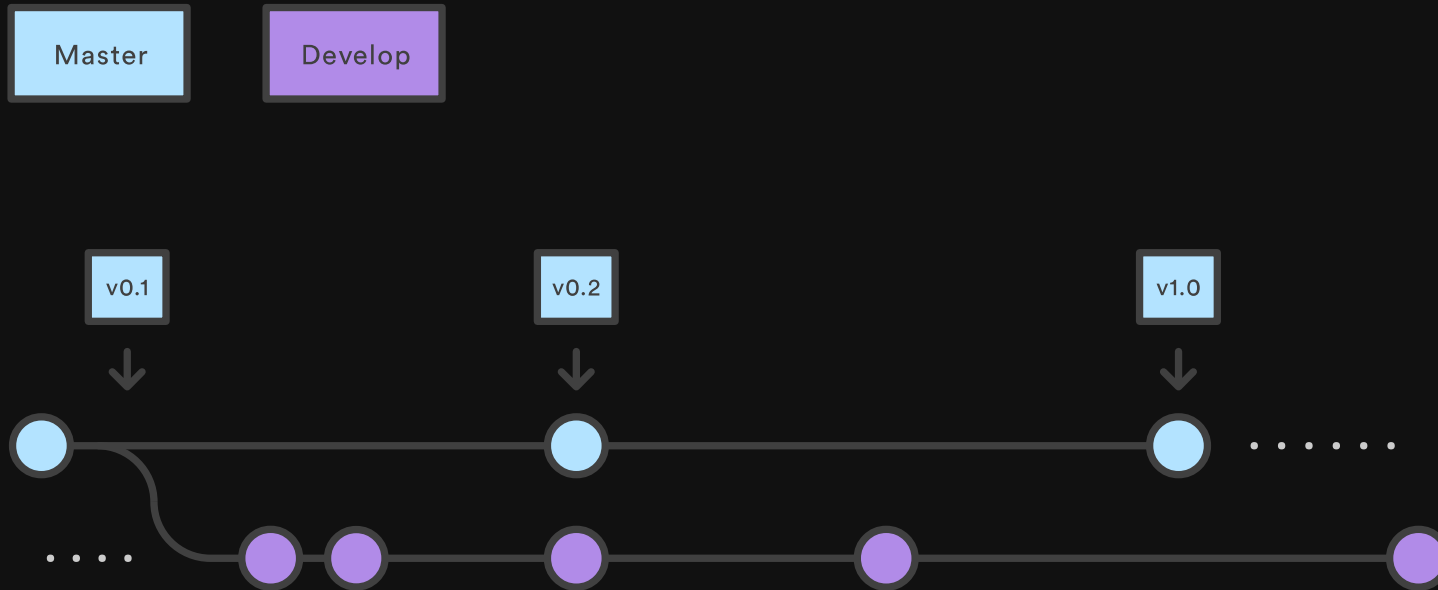
Chapter Three

My little OSS Sustainability Toolbox

Git Flow

The master branch stores the official release history, while development happens on the `develop` branch.

Git Flow % Feature Branches



Clean Code



Clean Javascript! 😊

```
'use strict'

var path = require('path')
var utils = require('../lib/utils')
var challenges = require('../data/datacache').challenges

exports = module.exports = function servePremiumContent () {
  return function (req, res) {
    if (utils.notSolved(challenges.premiumPaywallChallenge)) {
      utils.solve(challenges.premiumPaywallChallenge)
    }
    res.sendFile(path.resolve(__dirname, '../app/private/under-construction.gif'))
  }
}
```

Well, never mind...

```
fs.copy('app/index.template.html', 'app/index.html', {overwrite: true}, function () {
  if (config.get('application.logo')) {
    var logo = config.get('application.logo')
    if (utils.startsWith(logo, 'http')) {
      var logoPath = logo
      logo = decodeURIComponent(logo.substring(logo.lastIndexOf('/') + 1))
      utils.downloadToFile(logoPath, 'app/public/images/' + logo)
    }
    var logoImageTag = '<img class="navbar-brand navbar-logo" src="/public/images/'
    replace({ regex: /<img class="navbar-brand navbar-logo" (.*)=""/, replacement
  }
  if (config.get('application.theme')) {
    var themeCss = 'bower_components/bootswatch/' + config.get('application.theme')
    replace({ regex: /bower_components\/bootswatch\/.*\/bootstrap\.min\.css/, repla
  }
})
```

...it's still Javascript at the end of the day! 🤖

Test Automation



 **Protractor** *frisby.js*
end to end testing for AngularJS



KARMA



Jasmine



Example: UI Unit Test


```
it('should hold anonymous placeholder for email if current user is not logged in',
  inject(function () {
    $httpBackend.whenGET('/rest/user/whoami').respond(200, {user: {}})

    $httpBackend.flush()

    expect(scope.userEmail).toBe('anonymous')
  })
)
```

Example: API Integration Test

```
frisby.create('GET password change without passing any passwords')  
  .get(REST_URL + '/user/change-password')  
  .expectStatus(401)  
  .expectBodyContains('Password cannot be empty')  
  .toss()
```

No-idea-why-this--happens-snippet-injection Vulnerability: Did anyone notice how above code was totally butchered by highlight.js beginning at the opening single quote in line 1?

Example: Challenge E2E Test

```
describe('challenge "loginAdmin"', function () {
  it('should log in Admin with SQLI attack on email field using "\' or 1=1--"', function () {
    email.sendKeys('\'' or 1=1--')
    password.sendKeys('a')
    loginButton.click()

    expect(browser.getLocationAbsUrl()).toMatch(/\/search/)
  })
  protractor.expect.challengeSolved({challenge: 'Login Admin'})
})
```

Example: Checking if a challenge is solved

```
protractor.expect = {
  challengeSolved: function (context) {
    describe('(shared)', function () {
      beforeEach(function () {
        browser.get('/#/score-board')
      })

      it("challenge '" + context.challenge + "' should be solved on score board", function() {
        expect(element(by.id(context.challenge + '.solved')).getAttribute('class')).not
          expect(element(by.id(context.challenge + '.notSolved')).getAttribute('class')).
      })
    })
  }
}
```

What is a realistic goal for *Test Coverage*?

100%

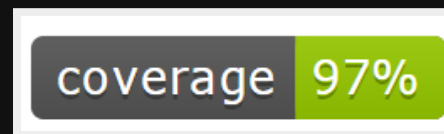
Juice Shop Test Coverage

Having all tests pass after a commit means, it's safe to merge PRs or publish a new release!

Juice Shop



CTF Extension

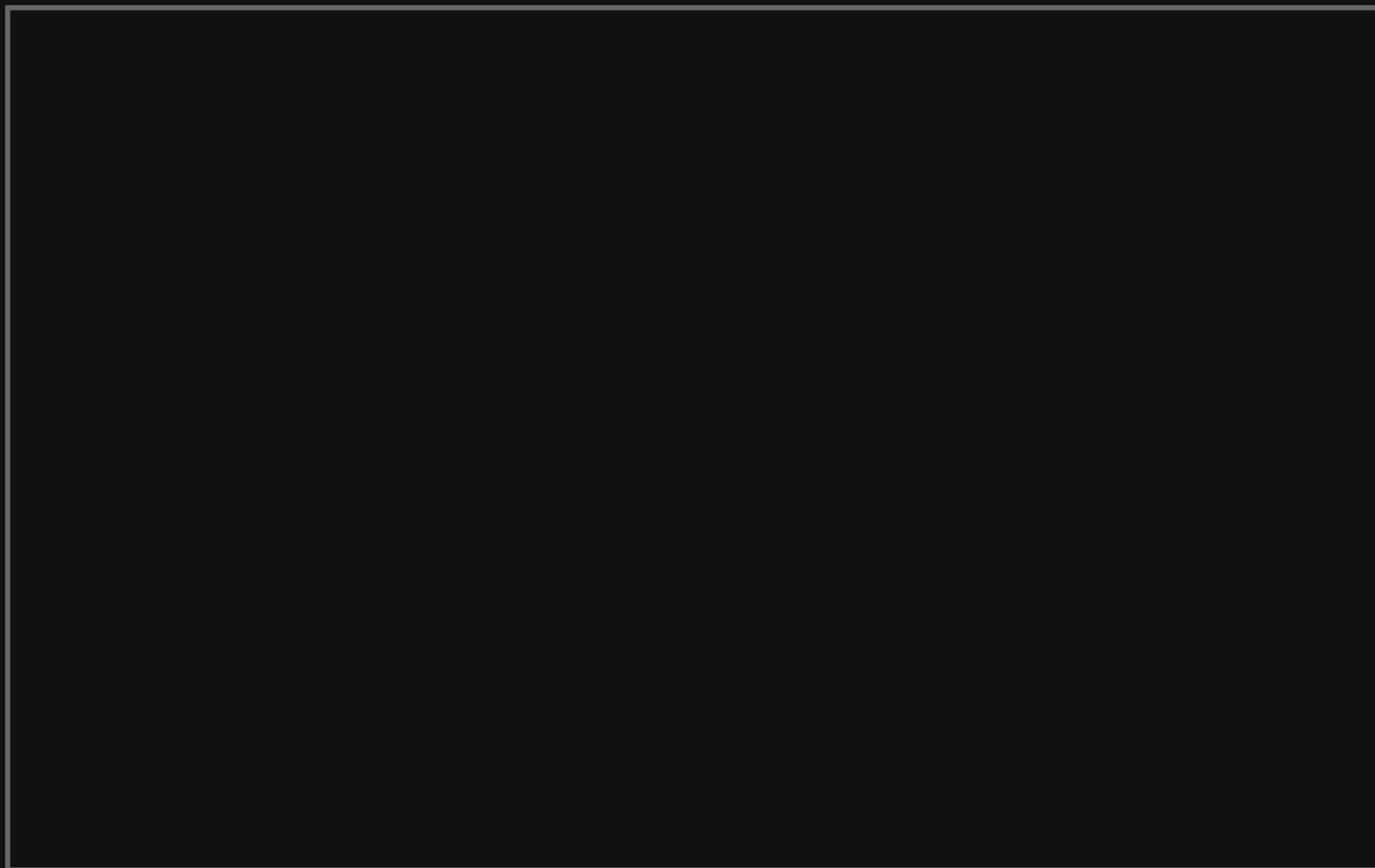


Mutation Testing



Reality check: Do your tests *actually test stuff*?

Example: Report for Juice Shop CTF



A realistic goal for *Mutation Coverage*?

100%

Juice Shop Mutation Coverage

Juice Shop

CTF Extension

92%* 98%

*of the UI unit tests!



CI/CD



Chapter undefined

Live Dual Release

What could possibly go wrong?

Release Dashboard

Juice Shop

build

passing



build passing

release

v3.0.1

CTF Extension

build

passing

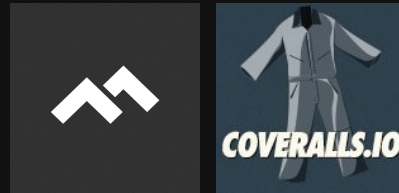
release

v1.0.1

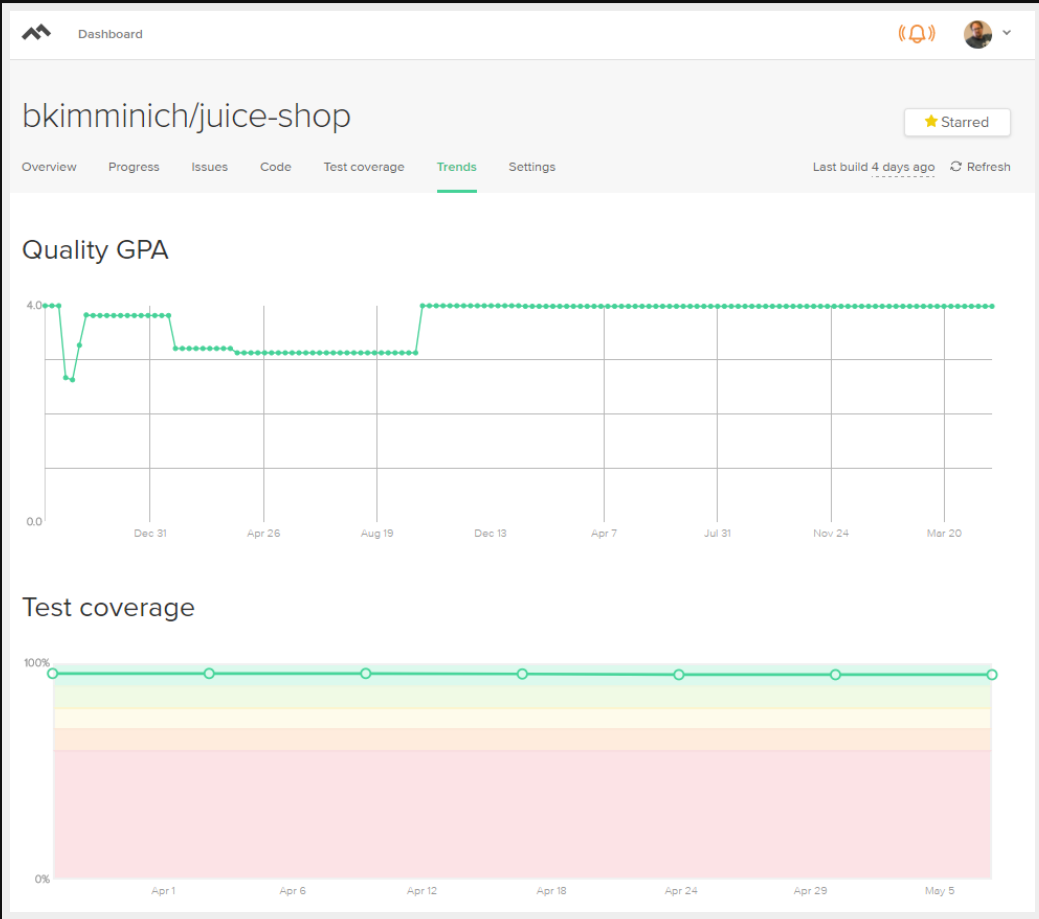
Chapter 3 (continued)

My little OSS Sustainability Toolbox

Quality Metrics



Example: Code Climate Stats



Example: bitHound Dashboard

bitHound BITHOUND 101 PRICING POPULAR BLOG

Find Public Repos on GitHub... Björn Kimminich

bkimminich / juice-shop
OWASP Juice Shop is an intentionally insecure webapp for security trainings written entirely in Javascript which encompasses the entire...

DASHBOARD DEPENDENCIES HACKS FILES SETTINGS

95

DEPENDENCIES ANALYSIS

93 86 PACKAGES 13 PRIORITY 3 MUTED 4 INSECURE 0 DISALLOWED 0 DEPRECATED 26 OUTDATED

PRIORITY DEPENDENCIES [LEARN MORE](#)

replace
Required: ~0.3 | Stable: 0.3.0 | MIT
INSECURE (1 DEEP)

karma [DEV DEPENDENCY](#)
Required: ~1.1 | Stable: 1.6.0 | MIT
INSECURE (3 DEEP) **OUTDATED** (7 MONTHS)

protractor [DEV DEPENDENCY](#)
Required: ~4.0 | Stable: 5.1.1 | MIT
INSECURE (3 DEEP) **OUTDATED** (5 MONTHS) **RISKY UPGRADE**

[SEE ALL 13 PRIORITY DEPENDENCIES](#)

CODE ANALYSIS

96 126 ANALYZED FILES 0 PRIORITY FILES 56 TEST FILES 0 BLACKLISTED FILES

master (2dd493c) analyzed 2 days ago

COMMUNITY RATING

(1)

★★★★★

YOUR RATING ★★★★★

OPEN SOURCE PROJECT

README	README.md
LICENSE	MIT
CHANGELOG	tags
LINT CONFIG	standard
BUILD SYSTEM	Grunt
BITHOUND CONFIG	.bithoundrc

[SHARE RESULTS](#)

[OVERALL BADGE](#)

Example: Coveralls at CTF-Extension

COVERALLS

- ← Back to Repo
- Travis Build #76
- 28026c90 on github
- Prev Build on develop (#75)
- Next Build on develop (#77)
- Delete

BKIMMINICH / JUICE-SHOP-CTF / 76

83%

MASTER: 97%

REPO ADDED
17 FEB 2017


TOTAL FILES
11

BUILDS
55

BADGE
coverage 97%

TOKEN
vU7o1e1uF90eDK4sM1dvy7M017E0TBE9X

COMMITTED 4 MAY 20 - 13:22 COVERAGE DECREASED (-3.7%) TO 82.927%

BUILD #	BUILD TYPE	COMMITTED BY	COMMIT MESSAGE	RUN DETAILS
# 76	push travis-ci	 bkimminich	Refactor and test hint creation	63.17 hits per line 102 of 123 relevant lines covered (82.93%) 12 of 22 branches covered (54.55%)

JOBS

COVERAGE	JOB	FILES COVERED	RAN
↓ 82.93	76.2 (2)	11	04 May 2017 TRAVIS JOB 76.2
↓ 82.93	76.3 (3)	11	04 May 2017 TRAVIS JOB 76.3

SOURCE FILES ON BUILD 76

ALL 11 CHANGED 2 SOURCE CHANGED 2 COVERAGE CHANGED 2 SEARCH:

COVERAGE	FILE	LINE	RELEVANT	COVERED	MISSED	HITS/LINE	BRANCH HITS	BRANCH MISSES
↑ 70.97	lib/generateSql.js	63	48	35	13	80.0	9	5 - 4

And now for the most important metric of all...





Truck Factor

The number of team members that would have to leave the project to effectively kill it.

What is a realistic goal for *Truck Factor*?



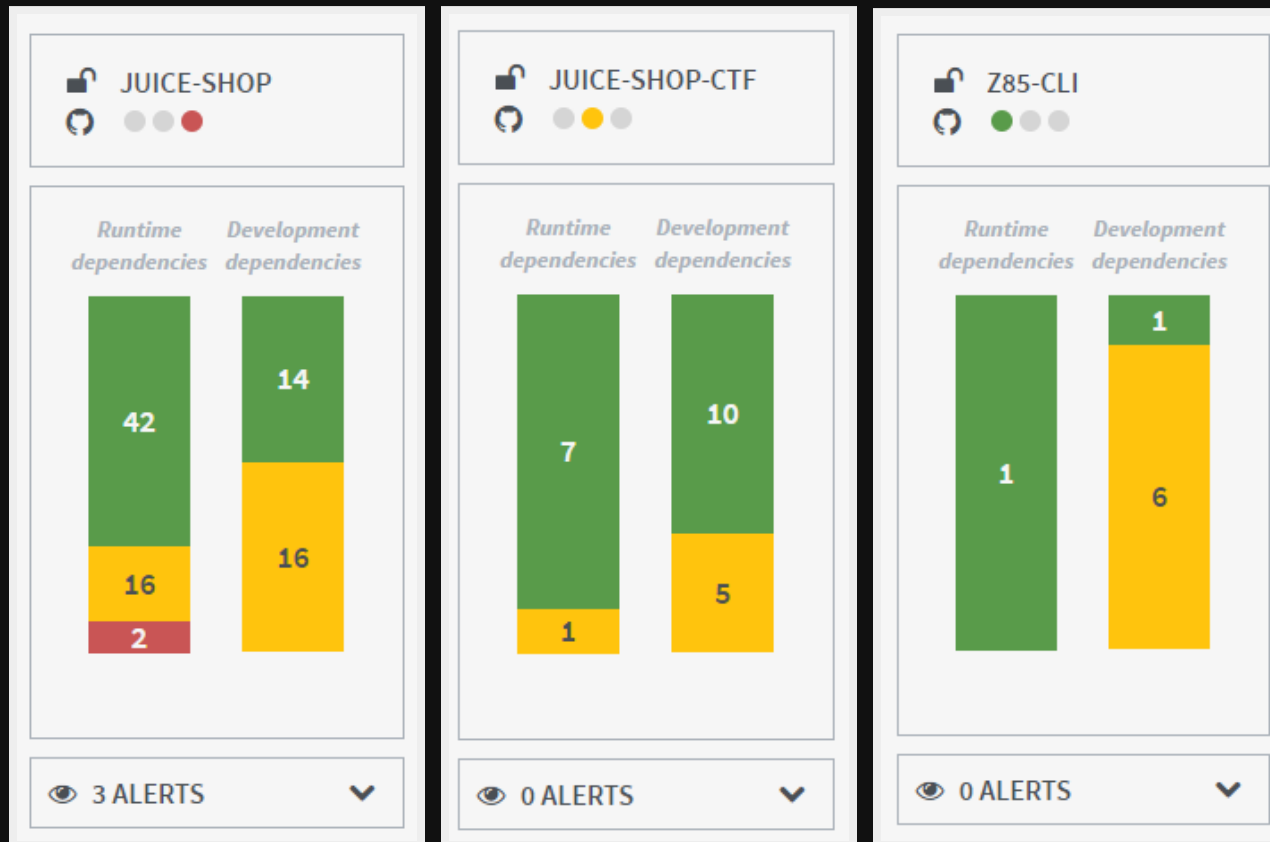
Meaning: If the whole team leaves, a new developer can take over and would be able to maintain & continue working on the project!

Dependency Control



gemnasium

Example: Gemnasium Dashboard



Chapter 4.1

New Open Source Antipatterns

Badge Barrage

The front-page is overcrowded with (mostly useless) information and status badges.

Overexaggerated Example



OWASP Juice Shop

owasp incubator

release v2.26.0

Follow

180

build passing build passing coverage 95% code climate 4.0 bitHound 94 contributors 14 Hu Board code style standard

dependencies insecure devDependencies up to date dependencies insecure dev dependencies insecure nsp no known vulns

heroku deployed Deploy to Heroku docker build automated docker pulls 5k downloads 292 total downloads 993/total

goodreads write review Donate Flattr tips \$0.05/week bitcoin 1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm

dash Xr556RzuwX6hg5EGpkybbv5RanJoZN17kW ether 0x0f933ab9fcaa782d0279c300d73750e1311eae6 license MIT License

The most trustworthy online shop out there. (@dschadow)

OWASP Juice Shop is an intentionally insecure web app for security trainings written entirely in Javascript which encompasses the entire [OWASP Top Ten](#) and other severe security flaws.

Bad Example

153 lines (112 sloc) | 8.4 KB

Raw Blame History

Contributing

contributors 15 Hu Board code style standard

build passing build passing coverage 95% code climate 4.0 bitHound 95

Found a bug? Crashed the app? Broken challenge? Found a vulnerability that is not on the Score Board?

Feel free to [create an issue](#) or [post your ideas in the chat](#)! Pull requests are also highly welcome - please follow the guidelines below to make sure your PR can be merged and doesn't break anything.

Code & Dependency Analysis Results

Provider	Status
Gemnasium	dependencies update!
David-DM	dependencies insecure devDependencies out of date
BitHound	dependencies insecure dev dependencies insecure
Node Security	nsp no known vulns

Git-Flow

This repository is maintained in a simplified [Git-Flow](#) fashion: All active development happens on the `develop` branch while `master` is used to deploy stable versions to the [Heroku demo instance](#) and later create tagged releases from

Coin Flip CI

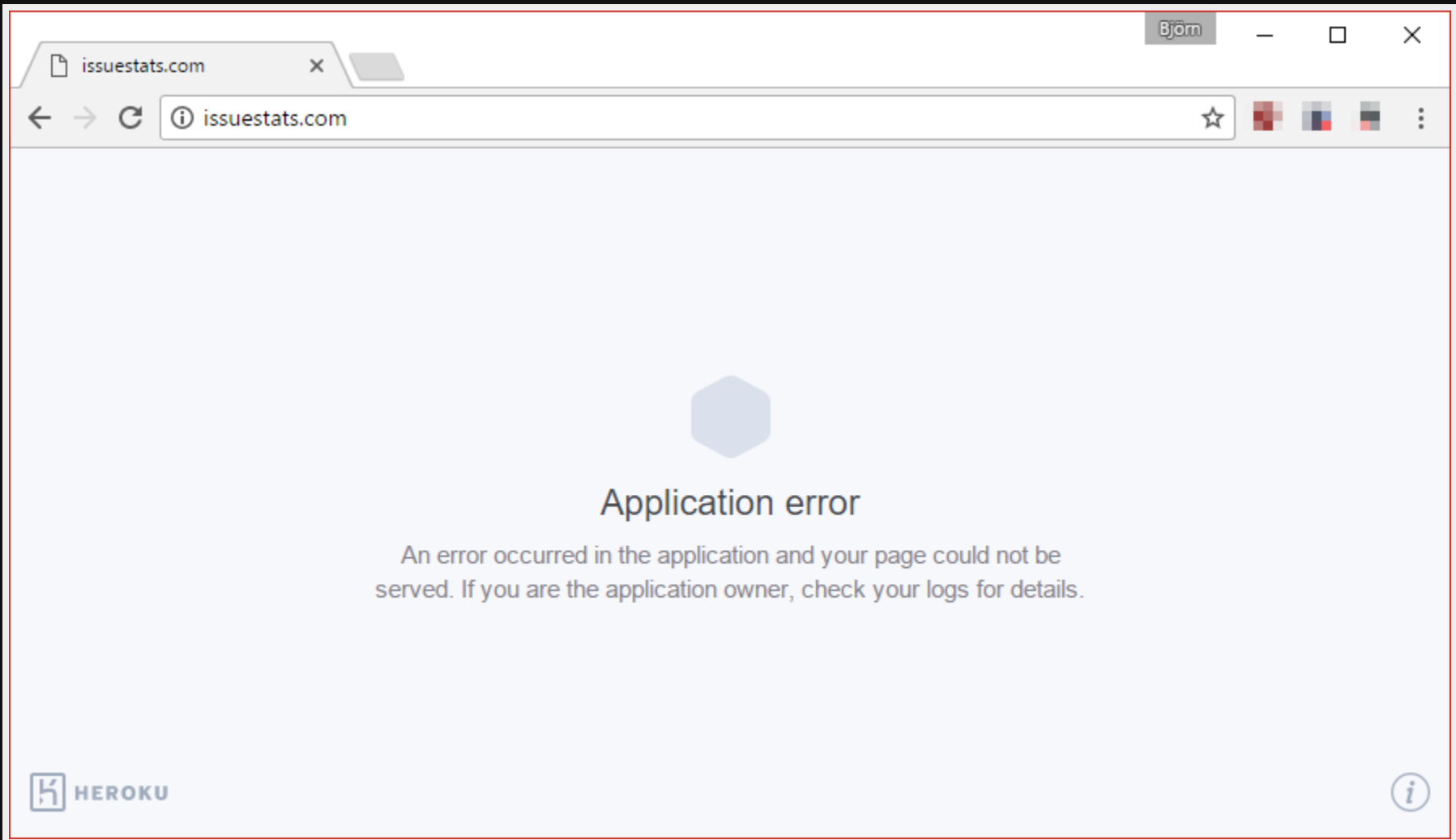
✓ ✓ ✗ ✓ ✗ ✓. With no code changes in between.

I wonder how our Live Release is doing...



Free beer

Some services might be free of charge for OSS projects but come with some other hidden costs or annoyance.





Enhance the value of your CI workflow

Sauce fits into your existing deployment pipeline to help scale your testing and pinpoint issues easily across hundreds of platforms.

[Get Started Free](#)

[Already a member? Login](#)



1514
started Oct 27th at 3:47AM by @juice-shop

Passed
7 tests ran in 25m 39s



1513
started Oct 27th at 3:39AM by @juice-shop

Passed
7 tests ran in 11m 52s



1512
started Oct 27th at 3:24AM by @juice-shop

4 Failed ▾
7 tests ran in 9m 28s



1507
started Oct 27th at 1:49AM by @juice-shop

6 Failed ▾
7 tests ran in 10m 3s

Wednesday, Oct 26th



1506
started Oct 26th at 9:16PM by @juice-shop

6 Failed ▾
7 tests ran in 11m 56s



1494
started Oct 26th at 5:44PM by @juice-shop

6 Failed ▾
7 tests ran in 11m 8s

Tuesday, Oct 25th



1479
started Oct 25th at 8:57PM by @juice-shop

5 Failed ▾
7 tests ran in 9m 38s



1478
started Oct 25th at 12:34PM by @juice-shop

Passed
7 tests ran in 9m 49s



1475
started Oct 25th at 2:02AM by @juice-shop

Passed
7 tests ran in 11m 36s



1474
started Oct 25th at 1:36AM by @juice-shop

1 Failed ▾
7 tests ran in 10m 5s

Contributor Laurels

Not giving enough credit to contributors.

Easy Counter: Make them visible

Credits

- Inspired by the "classic" [Bodgelt Store](#) by [@psiinon](#)
- Revised OWASP Juice Shop and Juice Shop CTF logo artworks by Emily Gundry (courtesy of [@SecureState](#))

Contributors

Ordered by date of first contribution. [Auto-generated](#) on Wed, 19 Apr 2017 08:32:58 GMT.

- [Björn Kimminich](#) aka [bkimminich](#)
- [Bitdeli Chef](#) aka [bitdeli-chef](#)
- [The Gitter Badger](#) aka [gitter-badger](#)
- [Aaron Edwards](#) aka [aaron-m-edwards](#)
- [Alec Brooks](#) aka [alecbrooks](#)
- [Dinis Cruz](#) aka [DinisCruz](#)
- [Timo Pagel](#) aka [wurstbrot](#)
- [Gorka Vicente](#) aka [gorkavicente](#)
- [Alvaro Viebrantz](#) aka [alvarowolfx](#)
- [Johanna A](#) aka [yuhama](#)
- [Stephen OBrien](#) aka [stephenobrien](#)
- [Joe Butler](#) aka [joelicious](#)
- [Abhishek bundela](#) aka [abhishekbundela](#)
- [ninoseki](#)
- [Jannik Hollenbach](#) aka [j12934](#)

Some contribs deserve bonus visibility

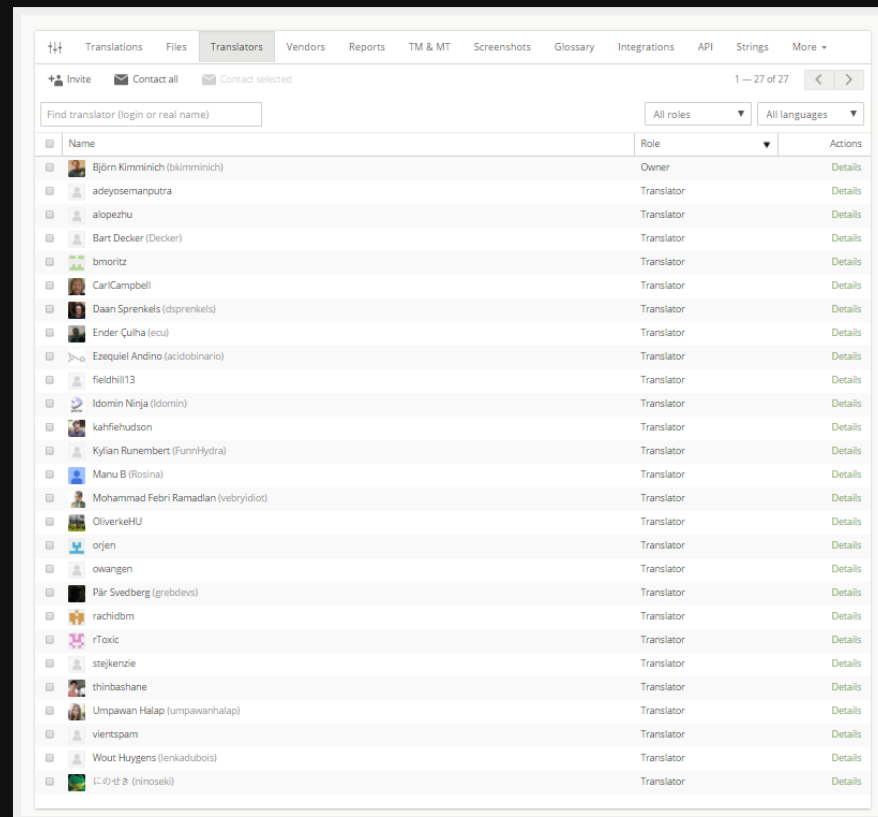
Josh Grossmann (CTFd SQLs )

Timo Pagel (Custom theming )

Jannik Hollenbach (CTF /Docker )

git **commits are not everything!**

Juice Shop's Crowdin Translators



The screenshot displays the 'Translators' tab in the Crowdin interface. At the top, there are navigation tabs: Translations, Files, Translators (selected), Vendors, Reports, TM & MT, Screenshots, Glossary, Integrations, API, Strings, and More. Below the navigation, there are options to 'Invite', 'Contact all', and 'Contact selected'. A search bar is labeled 'Find translator (login or real name)'. There are also dropdown menus for 'All roles' and 'All languages'. The main content is a table with 27 rows, each representing a translator. The table has three columns: Name, Role, and Actions. The roles listed are 'Owner' for the first translator and 'Translator' for the others. Each row has a 'Details' link in the Actions column.

Name	Role	Actions
Björn Kimminich (bkimminich)	Owner	Details
adeyosemanputra	Translator	Details
alopezhu	Translator	Details
Bart Decker (Decker)	Translator	Details
bmoritz	Translator	Details
CarlCampbell	Translator	Details
Daan Sprekels (dsprekels)	Translator	Details
Ender Çulha (ecu)	Translator	Details
Ezequiel Andino (acidobinario)	Translator	Details
fieldhill13	Translator	Details
Idomin Ninja (Idomin)	Translator	Details
kahfiehudson	Translator	Details
Kylian Runembert (FunnHydra)	Translator	Details
Manu B (Rosina)	Translator	Details
Mohammad Febri Ramadhan (vebrydiot)	Translator	Details
OliverkeHU	Translator	Details
orjen	Translator	Details
owangen	Translator	Details
Pär Svedberg (grebdevs)	Translator	Details
rachidbm	Translator	Details
rToxic	Translator	Details
stejkenzie	Translator	Details
thinbashane	Translator	Details
Umpawan Halap (umpawanhalap)	Translator	Details
vientspam	Translator	Details
Wout Huygens (lenkadubois)	Translator	Details
いのせき (ninoseki)	Translator	Details

<https://crowdin.com/project/owasp-juice-shop>

Bloggers and Podcasters

Web Links

- Interview on [OWASP 24/7 Podcast: Less than 10 Minutes Series: The Juice Shop Project](#)
- Vulnerable website collection on [Bonkers About Tech: 40+ Intentionally Vulnerable Websites To \(Legally\) Practice Your Hacking Skills](#)
- Hacking-session writeup on [Testhexen: Learning Application Security – Fun with the Juice Shop](#)
- Blog post (🇪🇺) on [LOL Security: Juice Shop Walkthrough](#) 🗣️
- Blog post on [IncognitJoe: Hacking\(and automating!\) the OWASP Juice Shop](#) 🗣️
 - [Automated solving script for the OWASP Juice Shop](#) written in Python as mentioned in above blog post 🗣️
- [7 Minute Security](#) Podcast:
 - Episode #229: [7MS #229: Intro to Docker for Pentesters](#) (Youtube)
 - Episode #230: [7MS #230: Pentesting OWASP Juice Shop - Part 1](#) (Youtube) 🗣️
 - Episode #231: [7MS #231: Pentesting OWASP Juice Shop - Part 2](#) (Youtube) 🗣️
 - Episode #232: [7MS #232: Pentesting OWASP Juice Shop - Part 3](#) (Youtube) 🗣️
 - Episode #233: [7MS #233: Pentesting OWASP Juice Shop - Part 4](#) (Youtube) 🗣️
 - Episode #234: [7MS #234: Pentesting OWASP Juice Shop - Part 5](#) (Youtube) 🗣️
- Guest post (🇩🇪) on [Informatik Aktuell: Juice Shop - Der kleine Saftladen für Sicherheitstrainings](#)
- Guest post on [The official Sauce Labs Blog: Proving that an application is as broken as intended](#)
- Teaser post on [Björn Kimminich's Blog: Juice Shop](#)

<https://github.com/bkimminich/juice-shop#references>

Conferences & Meetups

Conference and Meetup Appearances

2017

- 2 Hour Hacking: Juice Shop, 10.10.2017
- OWASP Juice Shop: Achieving sustainability for open source projects, AppSec Europe 2017, 11.05.2017
- OWASP Juice Shop: Stammtisch-Lightning-Update, 27. OWASP Stammtisch Hamburg, 25.04.2017
- Juice Shop Hacking Session, Software-Test User Group Hamburg, 21.03.2017
- Hands on = Juice Shop Hacking Session, Software Tester Group Hamburg (English-speaking), 16.03.2017
- Kurzvortrag: Hack the Juice Shop, PHP-Usergroup Hamburg, 14.02.2017

2016

- Lightning Talk: What's new in OWASP Juice Shop, German OWASP Day 2016, 29.11.2016
- Gothenburg pwns the OWASP Juice Shop, OWASP Gothenburg Day 2016, 24.11.2016
- Hacking the OWASP Juice Shop, OWASP NL Chapter Meeting, 22.09.2016 (Youtube, 🗣️ in last 10min)
- Hacking-Session für Developer (und Pentester), Kieler Open Source und Linux Tage, 16.09.2016
- Security-Auditing aus der Cloud – Softwareentwicklung kontinuierlich auf dem Prüfstand, SeaCon 2016, 12.05.2016
- Hacking the Juice Shop ("So ein Saftladen!"), JavaLand 2016, 08.03.2016
- Hacking the JuiceShop! ("Hackt den Saftladen!"), node.HH Meetup: Security!, 03.02.2016

2015

- Lightning Talk: Hacking the Juice Shop ("So ein Saftladen!"), German OWASP Day 2015, 01.12.2015
- Juice Shop - Hacking an intentionally insecure Javascript Web Application, JS Unconf 2015, 25.04.2015
- So ein Saftladen! - Hacking Session für Developer (und Pentester), 17. OWASP Stammtisch Hamburg, 27.01.2015

<https://github.com/bkimminich/juice-shop#references>

Pro Counter: Stickers!



Pro+ Counter: Shirts & Stuff!



Chapter NaN

Release Outcome

✓ = 😎 or ✗ = 😭?

Epic 😎/😓 Dashboard

Juice Shop

build passing

📍 build passing

release v3.0.1

CTF Extension

build passing

release v1.0.1



OWA
AppSec
Belt

8th to 12th
of May
2017

Waterfront
Conference
Center

Copyright (c) 2017 Björn Kimminich

Created with [reveal.js](#) - The HTML Presentation Framework