DEVELOPING A SAFER WEB WITH RUST

Stanko Krtalic Rusendic

github.com/stankec
@monorkin
hi@stanko.io

Stanko Krtalić Rusendić

Scandinavian & Rustacean

"Lets dockerize this Rust compiler so we can cross compile for Arduino".
Average happiness Level: 5/10.

01

THE WEB OF BUGS

(Something only spiders love)

Heart-bleed

FROM THE ARCHIVES

Gmail bug caused service to incorrectly categorize some...

January 29, 2014

Twitter crashes caused by software bug, disturb NBA-related...

June 21, 2012

Nokia issues fix and $100 credit for bug in Lumia 900...

April 12, 2012

MORE STORIES ABOUT

Consumer Protection

Equipment Failures

Apologies

Telephone Calls

AT&T Reaches Out to Public and Apologizes for Breakdown : Telecommunications: The firm suspects that a 'bug' in computer software caused its system's collapse. It may offer restitution to some and a day of discounted rates to all.

January 17, 1990 | KAREN TUMULTY | TIMES STAFF WRITER

NEW YORK — American Telephone & Telegraph, embarrassed and apologetic after its biggest long-distance breakdown ever, suggested Tuesday that it might make restitution to some victims and may offer all customers one day of discounted rates.

Company Chairman Robert E. Allen further admitted that AT&T operators compounded the problem during the early hours of the system collapse Monday by refusing to give customers the phone numbers they needed to gain access to competitors' long-distance systems.

Ultimately, he said, operators were instructed to "offer those numbers to our customers in the event they asked for them."

The problem is particularly embarrassing to the company because it has made reliability the cornerstone of its effort to win back market share from its smaller competitors. Although it is not yet clear how much,

CHRISTOPHER NULL 11.05.15 5:26 AM

HELLO, I'M MR. NULL. MY NAME MAKES ME INVISIBLE TO COMPUTERS

MOST POPULAR

PODCASTS
Aliens Would Probably Love Science Fiction
GEEK'S GUIDE TO THE GALAXY

GADGET LAB PODCAST
So Much Alexa
WIRED STAFF

PRODUCT REVIEW
Review: Polaroid Originals OneStep 2
BRENDAN NYSTEDT

MORE STORIES

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

In other projects
Wikimedia Commons

Languages
Español
Français
Italiano
日本語
Русский

Null Island

From Wikipedia, the free encyclopedia

Coordinates: 0°N 0°E

Null Island is a fictional island in the Gulf of Guinea added to the Natural Earth public domain map dataset,[1] located where the equator crosses the prime meridian, at coordinates 0°N 0°E.[2][3][4] Natural Earth describes the entity as a "1 meter square island" with "scale rank 100, indicating it should never be shown in mapping."[1] Although intended humorously, the fiction has a serious purpose and is used by mapping systems to trap errors.[3] Such errors arise, for example, where an image artifact is erroneously associated to the location by software which cannot attribute a geo-position, and instead associates a latitude and longitude of "Null,Null" or "0,0".[5] Null Island was developed as an idea in 2011 or slightly earlier. Since then, numerous web pages have documented this fictional landmass's flag, geography, and history.[6]

In reality, a weather observation buoy, part of the PIRATA system, is moored at the supposed location of the island.[7] The depth at this place is around 4,940 metres (16,210 ft).[8]

See also [edit]

- *Colonel Bleep*, a 1957 cartoon that took place on the fictitious Zero Zero Island, at the same location as Null Island, where earth's equator meets the Greenwich Meridian.

References [edit]

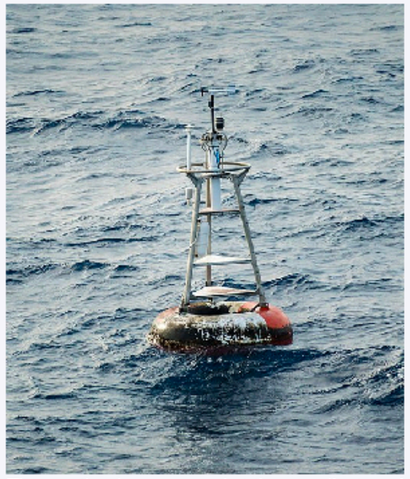
1. ^ a b Kurgan, Laura (2013). *Close Up at a Distance: Mapping, Technology and Politics*. New York: Zone Books. p. 157. ISBN 9781935408284.
2. ^ "Null Island". Retrieved 1 October 2013.
3. ^ a b "Natural Earth version 1.3 release notes". 2011-01-31. Retrieved 1 October 2013.
4. ^ Hotz, Robert Lee (14 July 2016). "If You Can't Follow Directions, You'll End Up on Null Island". *Wall Street Journal*. Retrieved 14 July 2016.
5. ^ "Null Island is One of the Most Visited Places on Earth. Too Bad It Doesn't Exist". Retrieved 25 August 2017.
6. ^ St. Onge, Tim. "The Geographical Oddity of Null Island". Library of Congress. Retrieved 13 May 2016.
7. ^ "Station 13010 - Soul". *National Data Buoy Center*. Retrieved 19 August 2014.
8. ^ "Bathymetric Data Viewer". *National Geophysical Data Center (NGDC)*. National Oceanic and Atmospheric Administration. Retrieved 2 July 2017.

External links [edit]

- Satirical website purporting to be that of The Republic of Null Island

This fictional location article is a stub. You can help Wikipedia by expanding it.

Null Island



The weather buoy moored at the coordinates of Null Island

Geography	
Coordinates	0°N 0°E
Additional information	
This fictional island is a cartographical convenience only	

Null Island

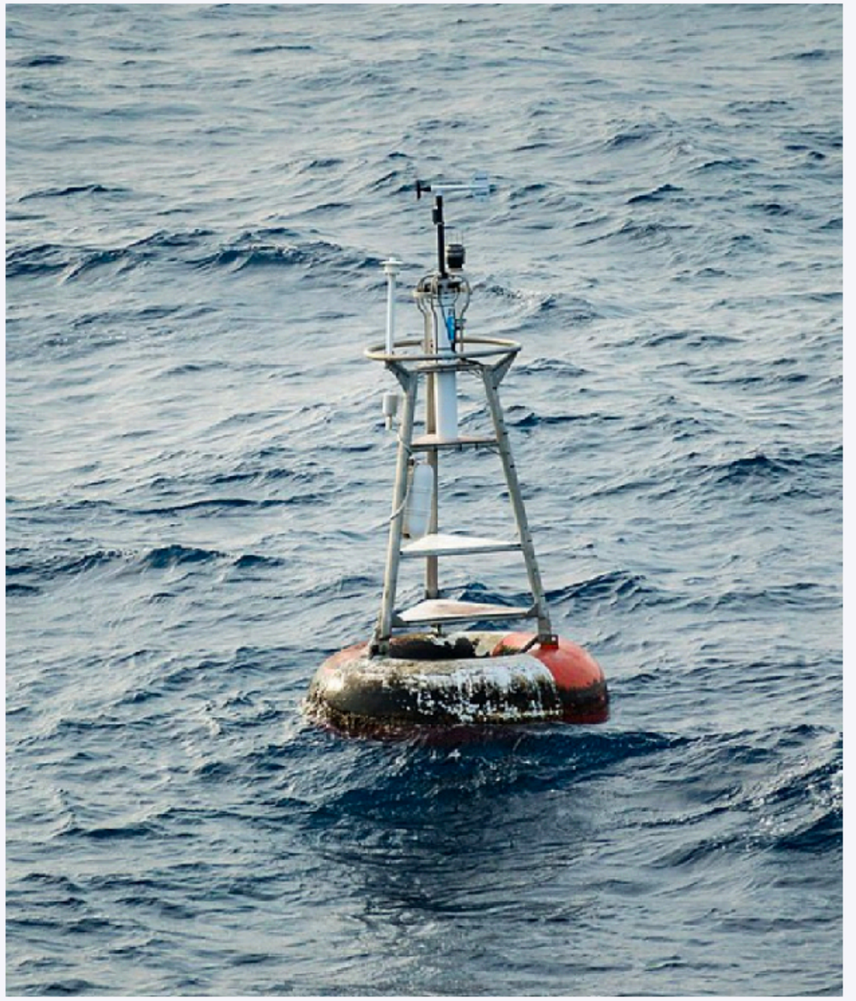
From Wikipedia, the free encyclopedia

Coordinates: 0°N 0°E

Null Island is a fictional island in the Gulf of Guinea added to the Natural Earth public domain map dataset,[1] located where the equator crosses the prime meridian, at coordinates 0°N 0°E.[2][3][4] Natural Earth describes the entity as a "1 meter square island" with "scale rank 100, indicating it should never be shown in mapping."[1] Although intended humorously, the fiction has a serious purpose and is used by mapping systems to trap errors.[3] Such errors arise, for example, where an image artifact is erroneously associated to the location by software which cannot attribute a geo-position, and instead associates a latitude and longitude of "Null,Null" or "0,0".[5] Null Island was developed as an idea in 2011 or slightly earlier. Since then, numerous web pages have documented this fictional landmass's flag, geography, and history.[6]

In reality, a weather observation buoy, part of the PIRATA system, is moored at the supposed location of the island.[7] The depth at this place is around 4,940 metres (16,210 ft).[8]

Null Island



The weather buoy moored at the coordinates of

See also [edit]

UK hacker exploits online bank loophole to steal £100,000

21 June 2017 | Technology

A hacker stole almost £100,000 by exploiting a loophole in Clydesdale Bank's online banking system

A UK hacker has been jailed for stealing almost £100,000 from a bank by exploiting a bug in the bank's online banking system.

James Ejankowski, who's 24, defrauded Clydesdale and Yorkshire Banking Group of more than £99,000 in December 2016.

He spent the money on a BMW, a Range Rover and tattoos for his face.

Top Stories

Police hunt for Las Vegas killer's motive

8 hours ago

US musician Tom Petty dies aged 66

3 hours ago

Anti-police strike grips Catalonia

42 minutes ago

Features

Las Vegas shooting - what we know

02

RUST

Memory safety

(Data) Race condition safe

NULL doesn’t exist

Performance

Low level

No garbage collector

Compiled

Results / Options

Ownership

Borrowing

Lifetimes

```
1
2  fn sign_in(username: String, password: String) → Result<User, String> {
3      if username == "user".to_string() && password == "password".to_string() {
4          return Ok(
5              User::new(username, password)
6          );
7      }
8
9      return Err("No such user found");
10 }
```

```
1
2  fn sign_in(username: String, password: String) → Result<User, String> {
3      if username == "user".to_string() && password == "password".to_string() {
4          return Ok(
5              User::new(username, password)
6          );
7      }
8
9      return Err("No such user found");
10 }
```

```
1
2  fn sign_in(username: String, password: String) → Result<User, String> {
3      if username == "user".to_string() && password == "password".to_string() {
4          return Ok(
5              User::new(username, password)
6          );
7      }
8
9      return Err("No such user found");
10 }
```

```
1
2  fn sign_in(username: String, password: String) → Result<User, String> {
3      if username == "user".to_string() && password == "password".to_string() {
4          return Ok(
5              User::new(username, password)
6          );
7      }
8
9      return Err("No such user found");
10 }
```

```
1
2  fn sign_in(username: String, password: String) → Result<User, String> {
3      if username == "user".to_string() && password == "password".to_string() {
4          return Ok(
5              User::new(username, password)
6          );
7      }
8
9      return Err("No such user found");
10 }
```

```
1
2  fn sign_in(username: String, password: String) → Result<User, String> {
3      if username == "user".to_string() && password == "password".to_string() {
4          return Ok(
5              User::new(username, password)
6          );
7      }
8
9      return Err("No such user found");
10 }
```

Username and password are deallocated

```
1
2 fn process_login(username: String, password: String) {
3     match sign_in(username, password) {
4         Ok(user) ⇒ println!("User {} logged in", user.username),
5         Err(message) ⇒ println!("ERROR: {}", message)
6     }
7 }
```

```
fn main {
    let name = "petar".to_string();
    let amne = shuffle(name); // ⟵ The function shuffle became the owner of
                              //     the name variable

    println!("Original: {}", name); // ⟵ Errors during compilation
                                    //  We tried to use a variable we don't own
    println!("Shuffled: {}", amne);
}
```

03

WEB

Rocket

Introducing Rocket.

Rocket is a web framework for Rust that makes it simple to write *fast* web applications without sacrificing flexibility or **type safety**. All with minimal code.

Get Started | Learn More

Latest Release: 0.3.3 (Sep 25, 2017)

Type Safe

From request to response Rocket ensures that your types mean something.

Learn More

Boilerplate Free

Spend your time writing code that really matters, and let Rocket generate the rest.

See Examples

Easy To Use

Rocket makes extensive use of Rust's code generation tools to provide a clean API.

Get Started

Extensible

Easily create your own primitives that any Rocket application can use.

See How

```
1  #![feature(plugin)]
2  #![plugin(rocket_codegen)]
3
```

Hello, Rocket!

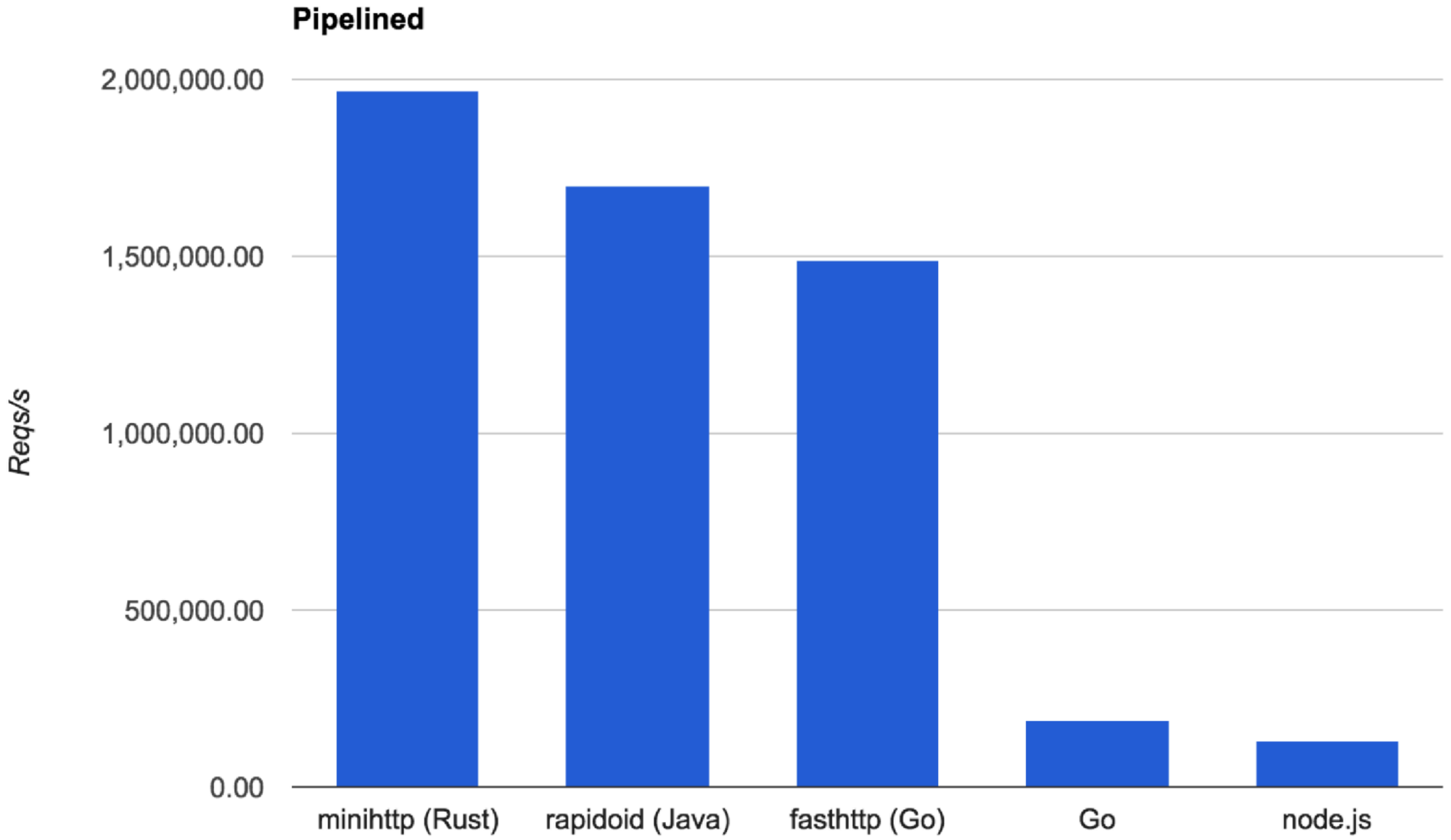
```
#[get("/user/<id>")]
fn user(id: usize) → T {  ...  }

#[get("/user/<id>", rank = 2)]
fn user_int(id: isize) → T {  ...  }

#[get("/user/<id>", rank = 3)]
fn user_str(id: &RawStr) → T {  ...  }

```

```rust
#[get("/admin")]
fn admin_panel(admin: AdminUser) -> &'static str {
    "Hello, administrator. This is the admin panel!"
}

#[get("/admin", rank = 2)]
fn admin_panel_user(user: User) -> &'static str {
    "Sorry, you must be an administrator to access this page."
}

#[get("/admin", rank = 3)]
fn admin_panel_redirect() -> Redirect {
    Redirect::to("/login")
}
```

```rust
#[get("/")]
fn index(cookies: Cookies) -> Option<String> {
    cookies.get("message")
        .map(|value| format!("Message: {}", value))
}

#[derive(FromForm)]
struct Task {
    complete: bool,
    description: String,
}

#[post("/todo", data = "<task>")]
fn new(task: Form<Task>) -> String { ... }

#[derive(Deserialize)]
struct Task {
    description: String,
    complete: bool
}

#[post("/todo", data = "<task>")]
fn new(task: Json<Task>) -> String { ... }
```

```
Running 10s test @ http://localhost:80
  1 threads and 18 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency   153.01us   42.25us 449.00us   75.54%
    Req/Sec    75.58k    11.75k   90.22k    54.46%
  758044 requests in 10.10s, 105.55MB read
Requests/sec:  75051.28
Transfer/sec:     10.45MB
```

Here are the results, in number of “Hello world!"s served per second on an 8 core Linux machine:



Diesel is a Safe, Extensible ORM and Query Builder for Rust

Diesel is the most productive way to interact with databases in Rust because of its safe and composable abstractions over queries.

Get Started View on Github

Why did we make Diesel?

Preventing Runtime Errors

We don't want to waste time tracking down runtime errors. We achieve this by having Diesel eliminate the possibility of incorrect database interactions at compile time.

Built for Performance

Diesel offers a high level query builder and lets you think about your problems in Rust, not SQL. Our focus on zero-cost abstractions allows Diesel to run your query and load your data even

Productive and Extensible

Unlike Active Record and other ORMs, Diesel is designed to be abstracted over. Diesel enables you to write reusable code and think in terms of your problem domain and not SQL.

```rust
fn main() {
    use diesel_demo::schema::posts::dsl::*;

    let connection = establish_connection();
    let result = posts.filter(published.eq(true))
        .limit(5)
        .load::<Post>(&connection);
    let posts = match result {
        Ok(posts) => posts,
        Err(message) => {
            println!("ERROR: {}", message);
            return;
        }
    };

    println!("Displaying {} posts", posts.len());
}
```

```rust
infer_schema!("dotenv:DATABASE_URL");

#[derive(Queryable)]
pub struct Post {
    pub id: i32,
    pub title: String,
    pub body: String,
    pub published: bool,
}
```

```
infer_schema!("dotenv:DATABASE_URL");

#[derive(Queryable)]
pub struct Post {
    pub id: i32,
    pub title: String,
    pub body: String,
    pub published: bool,
}
```

```
⇒ diesel migration generate create_posts
```

Iron

Tokio

Mio

Rayon

Futures

Grpc-rust

Juniper

Rust-ampq


CANONICAL
wire
Dropbox
npm
SENTRY
redhat
mozilla
coursera
Braintree

04

WHEN TO USE IT

Rust is not a silver bullet

Steep learning curve

Slower development cycle

But…

Processing sensitive data

– e.g. Fintech

High throughput public-facing services

- e.g. Auth servers

Systems that are too important to fail

If you want to use Go consider using Rust

QUESTIONS

 github.com/stankec
 @monorkin
 hi@stanko.io

Leave feedback on:
https://goo.gl/rwXEXd