Garfield's worst nightmare

Valdemar Erk

RustWeek 2025

2025-05-14

OUTLINE

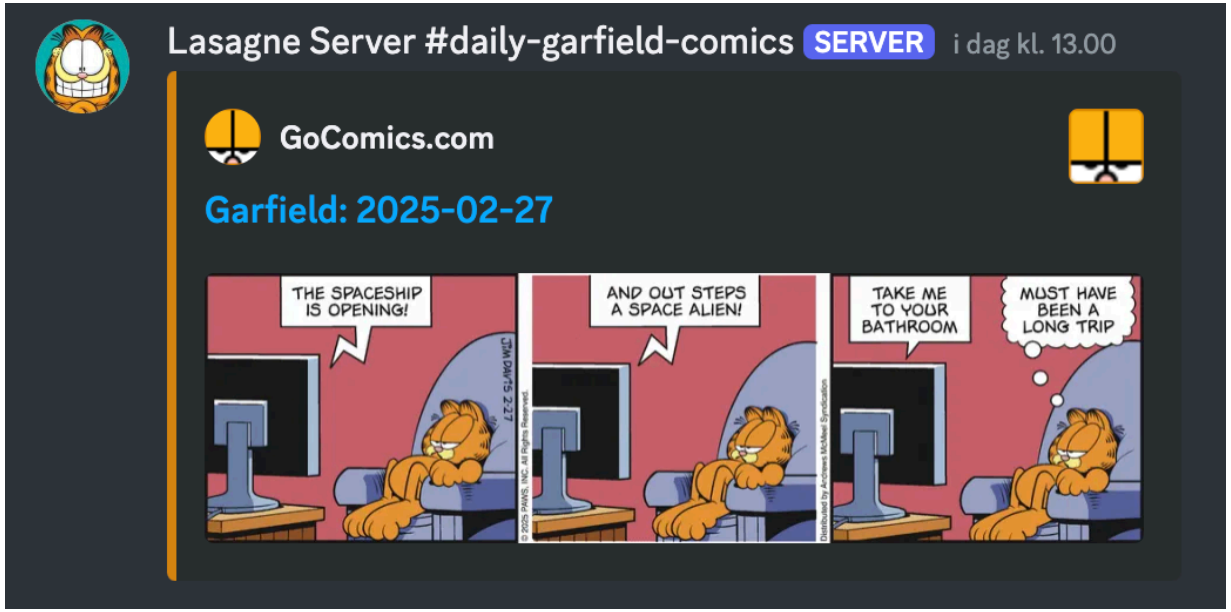
00 Prerequisites
01 The Incident
02 Interlude: What is a date?
03 Garfield

00

Prerequisites

I have a Discord bot, it posts the new comic every single day:

I have a Discord bot, it posts the new comic every single day:



The official web syndicator GoComics does not have a API, so I fetch the website instead.

- Fetch website
- Extract URL for the comic
- Save it in a database.

The official web syndicator GoComics does not have a API, so I fetch the website instead.

- Fetch website
- Extract URL for the comic
- Save it in a database.

The database is keyed by Garfield Epoch which are days since June 19, 1978.

The official web syndicator GoComics does not have a API, so I fetch the website instead.

- Fetch website
- Extract URL for the comic
- Save it in a database.

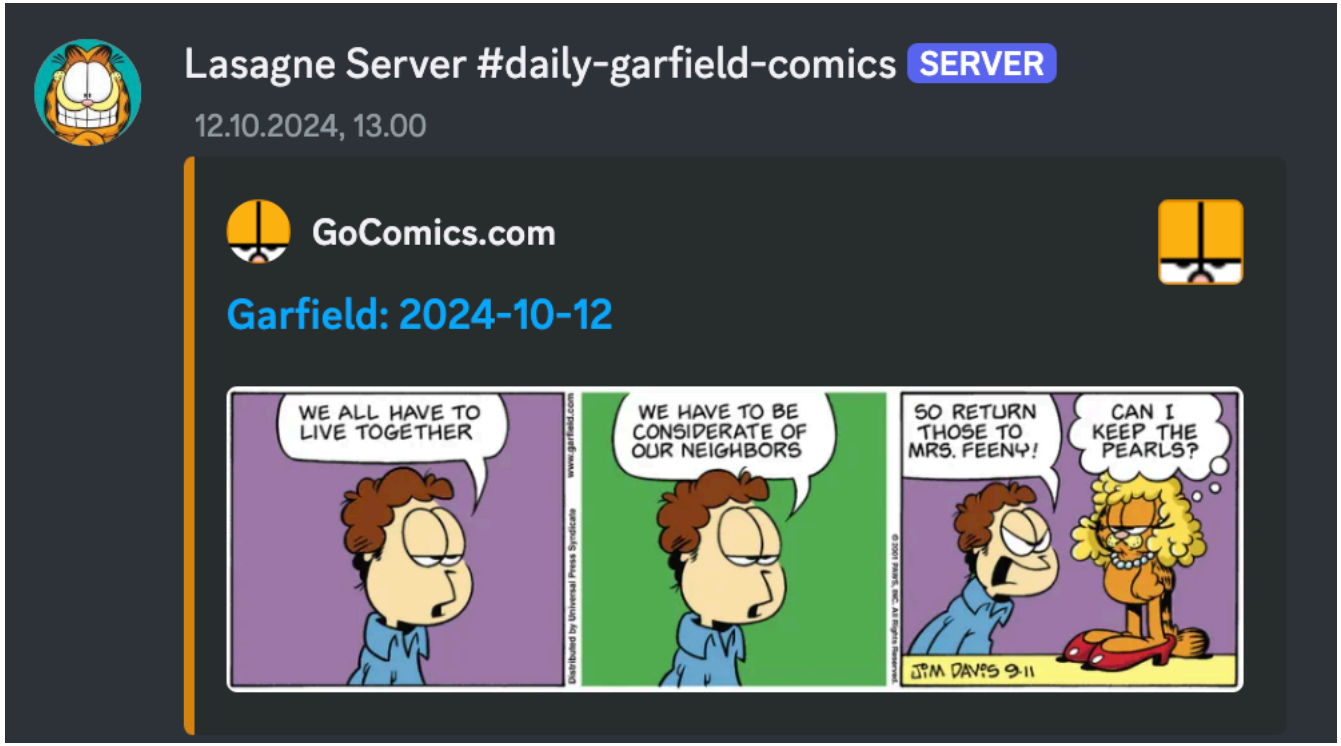
The database is keyed by Garfield Epoch which are days since June 19, 1978.

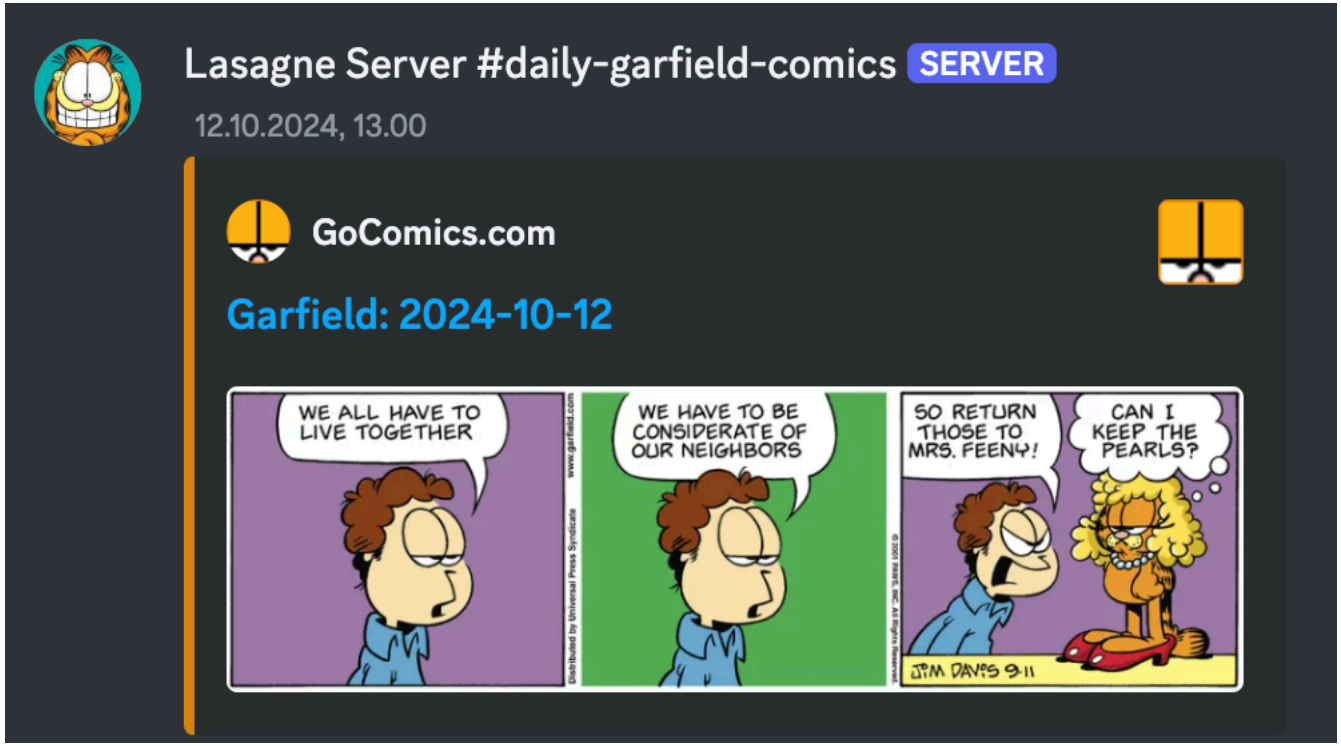
For example today is day 17131 in Garfield Epoch.

01

The Incident

The date is 2024-10-12


Lasagne Server #daily-garfield-comics SERVER
12.10.2024, 13.00
GoComics.com
Garfield: 2024-10-12
WE ALL HAVE TO LIVE TOGETHER
WE HAVE TO BE CONSIDERATE OF OUR NEIGHBORS
SO RETURN THOSE TO MRS. FEENY!
CAN I KEEP THE PEARLS?
JIM DAVIS 9-11



The comic is for 2001-09-11

- Delete the database and restart the bot,

- Delete the database and restart the bot, Issue still exists.

- Delete the database and restart the bot, Issue still exists.
- Rip out the code getting the old code from the database,

- Delete the database and restart the bot, Issue still exists.
- Rip out the code getting the old code from the database, Issue resolved.

- Delete the database and restart the bot, Issue still exists.
- Rip out the code getting the old code from the database, Issue resolved.

Bug probably exists in the database?

- Delete the database and restart the bot, Issue still exists.
- Rip out the code getting the old code from the database, Issue resolved.

Bug probably exists in the database?

But why did it suddenly show up?

My first guess was a database failure.

My first guess was a database failure.

I was running a now unsupported backend for heed, MDBX and had recently moved it to a FreeBSD system.

My first guess was a database failure.

I was running a now unsupported backend for heed, MDBX and had recently moved it to a FreeBSD system.

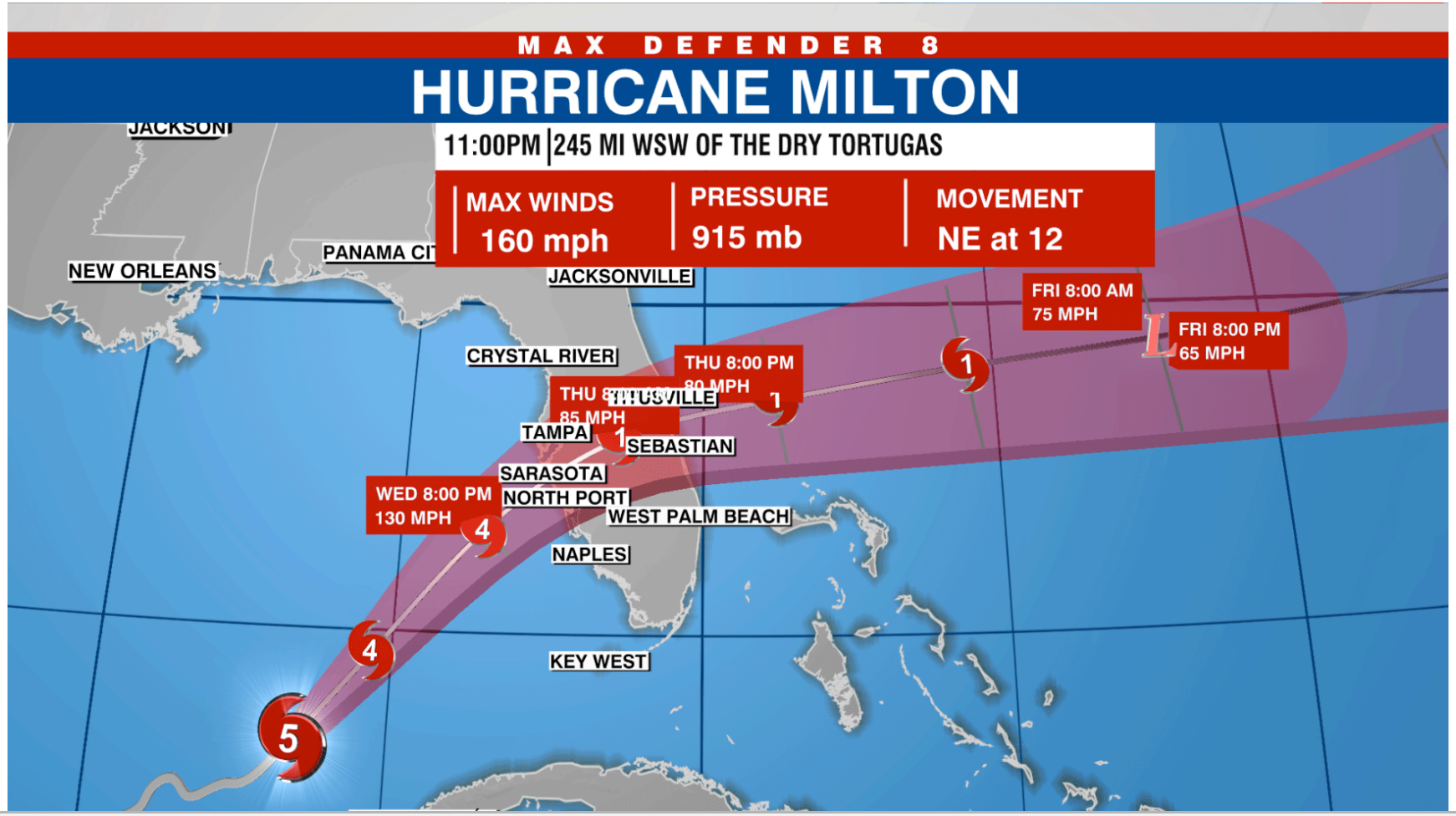
Next thing was to ask myself what had changed recently.

My first guess was a database failure.

I was running a now unsupported backend for heed, MDBX and had recently moved it to a FreeBSD system.

Next thing was to ask myself what had changed recently.

But how recently?


MAX DEFENDER 8
HURRICANE MILTON
11:00PM | 245 MI WSW OF THE DRY TORTUGAS
MAX WINDS
160 mph
PRESSURE
915 mb
MOVEMENT
NE at 12
JACKSONVILLE
NEW ORLEANS
PANAMA CI
CRYSTAL RIVER
THU 8:00 PM
80 MPH
THU 8:00 PM
85 MPH
TAMPA
SEBASTIAN
SARASOTA
NORTH PORT
WED 8:00 PM
130 MPH
WEST PALM BEACH
NAPLES
KEY WEST
FRI 8:00 AM
75 MPH
FRI 8:00 PM
65 MPH

Around 6 months before the incident.

Around 6 months before the incident.

```
commit 689511abad806b3a2b5c8aec861ff9f35fe824db
Author: Valdemar Erk <valdemar@erk.dev>
Date:   Sat Mar 9 16:12:44 2024 +0100

    use eos instead of time
```

02

Interlude: What is a date?

- Time
- Eos

```rust
pub struct Date {
    /// Bitpacked field containing the year, ordinal,
    /// and whether the year is a leap year.
    // |     x      | xxxxxxxxxxxxxxxxxxxx |       x       | xxxxxxxxx |
    // |   1 bit    |       21 bits        |     1 bit     |  9 bits   |
    // | unassigned |         year         | is leap year? |  ordinal  |
    // The year is 15 bits when `large-dates` is not enabled.
    value: NonZeroI32,
}
```

```rust
pub struct Date {
    /// Bitpacked field containing the year, ordinal,
    /// and whether the year is a leap year.
    // |     x      | xxxxxxxxxxxxxxxxxxxxx |       x       | xxxxxxxxx |
    // |   1 bit    |        21 bits        |     1 bit     |  9 bits   |
    // | unassigned |         year          | is leap year? |  ordinal  |
    // The year is 15 bits when `large-dates` is not enabled.
    value: NonZeroI32,
}

pub struct Duration {
    secs: u64,
    nanos: Nanoseconds, // Always 0 <= nanos < NANOS_PER_SEC
}
```

`Nanoseconds` is a `u32` with niche optimizations.

Key generating code:

```
let days = (comic_date - GARFIELD_EPOCH).whole_days();
```

```rust
/// A concrete date in the proleptic Gregorian calendar.
#[derive(Debug, Clone, Copy, PartialEq, Eq, PartialOrd, Ord, Hash)]
pub struct Date {
    pub(crate) year: i16,
    pub(crate) month: u8,
    pub(crate) day: u8,
}
```

```rust
/// A concrete date in the proleptic Gregorian calendar.
#[derive(Debug, Clone, Copy, PartialEq, Eq, PartialOrd, Ord, Hash)]
pub struct Date {
    pub(crate) year: i16,
    pub(crate) month: u8,
    pub(crate) day: u8,
}

pub struct Interval {
    months: i32,
    days: i32,
    microseconds: i64,
}
```

Key generating code:

```
let days = (comic_date - GARFIELD_EPOCH).days() as i64;
```

Key generating code:

```
let days = (comic_date - GARFIELD_EPOCH).days() as i64;
```

```
let days = (date!(2024-10-12) - GARFIELD_EPOCH).days() as i64;
eprintln!(days); // 23
```

03

Garfield

03.00 The dice rolled a critical failure

Days, months and years between 1978-06-19 and 2001-09-11:

days: 23, months: 2, years: 23

Days, months and years between 1978-06-19 and 2001-09-11:

days: 23, months: 2, years: 23

Days, months and years between 1978-06-19 and 2024-10-12:

days: 23, months: 3, years: 46

The bug was fixed by changing the line into:

```
let days = comic_date.days_since(GARFIELD_EPOCH) as i64;
```

The bug was fixed by changing the line into:

```
let days = comic_date.days_since(GARFIELD_EPOCH) as i64;
```

Moral of the story?

The bug was fixed by changing the line into:

```
let days = comic_date.days_since(GARFIELD_EPOCH) as i64;
```

Moral of the story?

Probably read docs.

The bug was fixed by changing the line into:

```
let days = comic_date.days_since(GARFIELD_EPOCH) as i64;
```

Moral of the story?

Probably read docs.

And test stuff.

```rust
#[test]
fn unique_dates() {
    let start = GARFIELD_EPOCH.at(eos::time!(07:00));
    let iter = start
        .every(eos::Interval::from_days(1))
        .at(eos::time!(07:00))
        .until(GARFIELD_EPOCH.with_year(3000).unwrap()
                .at(eos::time!(07:00)),
        )
        .into_iter();

    let mut set = std::collections::HashSet::new();
    for d in iter {
        assert!(set.insert(ComicDb3::key(d.date())));
    }
}
```