

Building **Wasp**, an Open-Source Web Framework

What, why, and how?

Building **Wasp**, an Open-Source Web Framework

What, why, and how?

Building Wasn
Open-Source
Fram
What,

The Power of DSLs and the
Weird Startup World

**Wasp is a web framework
implemented as a DSL**


Open-source



**Wasp is a web framework
implemented as a DSL**

(Built by a startup)

Open-source



**Wasp is a web framework
implemented as a DSL**

(Built by a startup)

Open-source

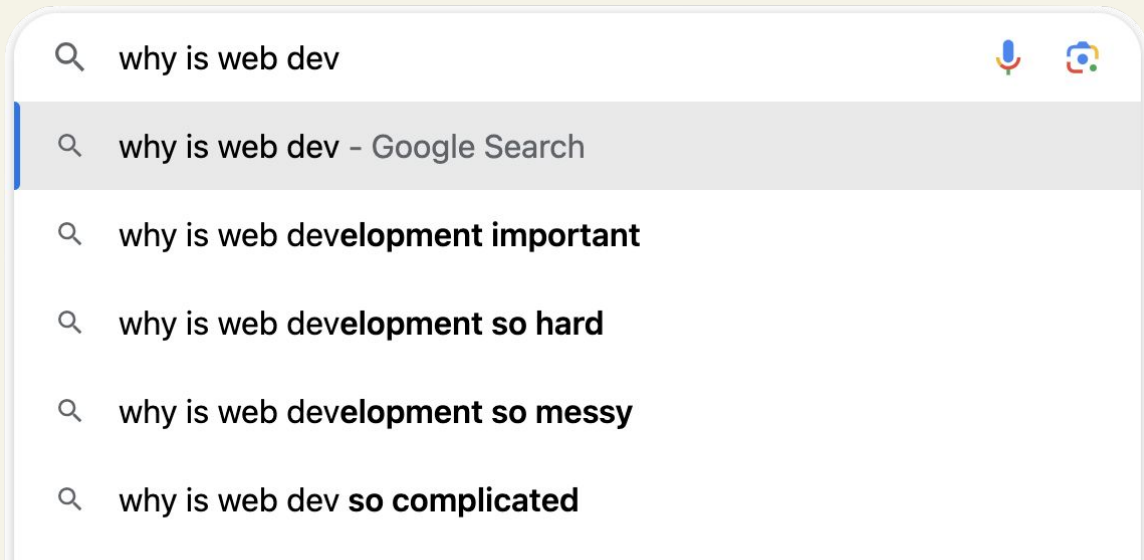


**Wasp is a web framework
implemented as a DSL**

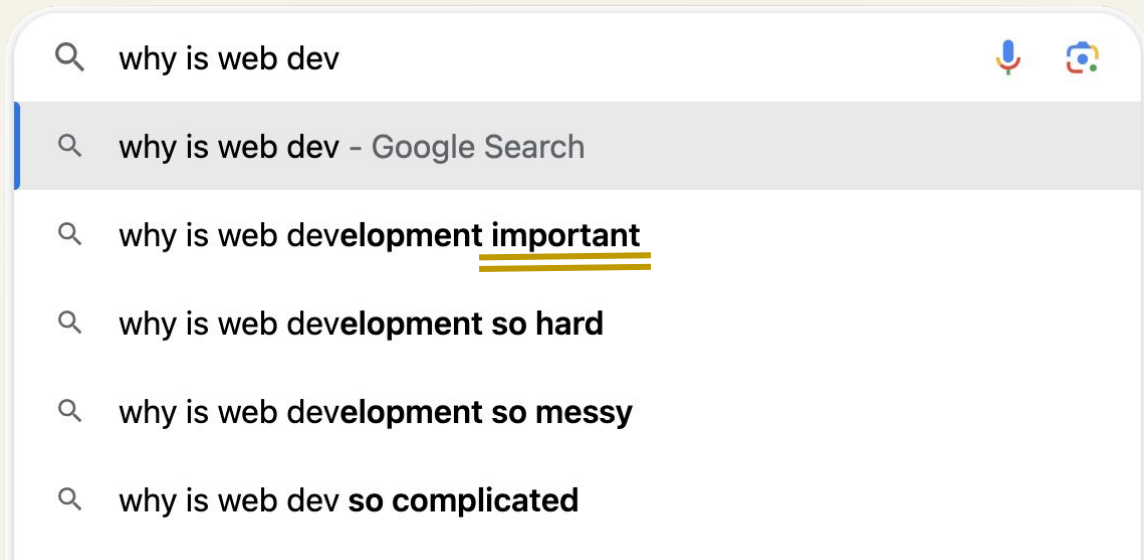
(Built by a startup)

**Another framework,
seriously? Why?**

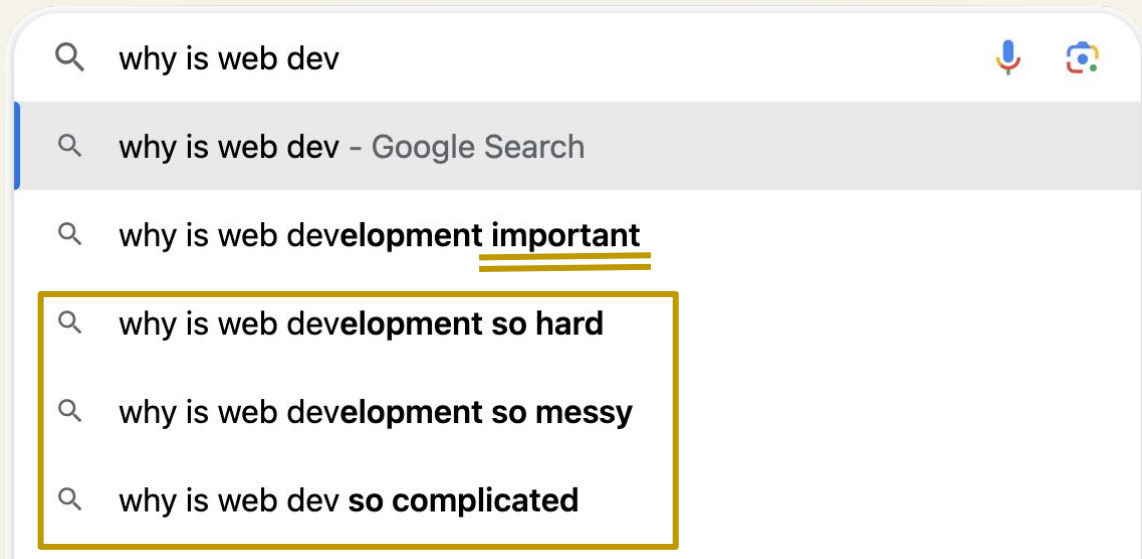
Another **framework**, seriously? Why?



Another **framework**, seriously? Why?



Another **framework**, seriously? Why?



**Wasp is a web framework
implemented as a DSL**

The power of **DSLs**

" [0-9]+ \. [0-9]+ \. [0-9]+ \. [0-9]+ "

```
SELECT *  
FROM Customer  
WHERE Age > 25;
```

```
function HelloMessage({ name }) {  
  return <div>Hello {name}</div>;  
}
```


Language

Specific Language

Domain Specific Language

Domain Specific Language

Domain Specific Language

CSS



NGINX



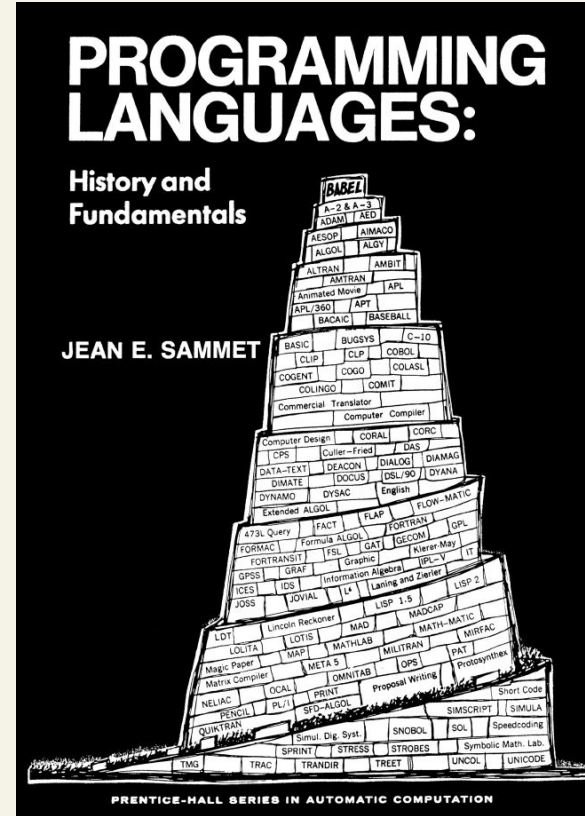
The AWK
Programming
Language

HTML



Ok... but do you really need a new **language**?

(CACM Cover, Jan 1961)



" [0-9]+ \. [0-9]+ \. [0-9]+ \. [0-9]+ "

" [0 - 9] + \ . [0 - 9] + \ . [0 - 9] + \ . [0 - 9] + "

116 . 97 . 107 . 61
123 . 16 . 225 . 110
13 . 50 . 243 . 227
134 . 209 . 213 . 5

" [0 -

```
import sys

def extract_ip_addresses():
    ip_addresses = list()

    for line in sys.stdin:
        parts = line.split()
        for part in parts:
            octets = part.split('.')
            if len(octets) < 4:
                continue;
            if all(o.isdigit() for o in octets[:4]):
                ip_addresses.append('.'.join(octets[:4]))

    return ip_addresses

for ip in extract_ip_addresses():
    print(ip)
```

-9] + "

```
filip@xps-15 → ~/Projects/proxy-shennanigans
> cat dump | py ex2.py
File "/home/filip/Projects/proxy-shennanigans/ex2.py", line 21
  for ip in ips:
IndentationError: expected an indented block after 'for' statement on line 21
```

```
filip@xps-15 → ~/Projects/proxy-shennanigans
> cat dump | py ex2.py
File "/home/filip/Projects/proxy-shennanigans/ex2.py", line 22
  for ip in ips:
IndentationError: expected an indented block after 'for' statement on line 22
```

```
filip@xps-15 → ~/Projects/proxy-shennanigans
> cat dump | py ex2.py
File "/home/filip/Projects/proxy-shennanigans/ex2.py", line 20
  for ip in ips = extract_ip_addresses(file_path):
  ^
```

```
SyntaxError: invalid syntax
filip@xps-15 → ~/Projects/proxy-shennanigans
> □
```

```
filip@xps-15 → ~/Projects/proxy-shennanigans
> nvim ex2.py
File "/home/filip/Projects/proxy-shennanigans/ex2.py", line 20, in <module>
  ips = extract_ip_addresses(file_path)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/home/filip/Projects/proxy-shennanigans/ex2.py", line 13, in extract_ip_addresses
  if all(o.isdigit() for o in octet):
  ^^^^^
SyntaxError: list comprehension iterable element 0 is not a character or string literal. Did you mean: 'octets'?
shennanigans
```

```
filip@xps-15 → ~/Projects/proxy-shennanigans
> cat dump | py ex2.py
Traceback (most recent call last):
  File "/home/filip/Projects/proxy-shennanigans/ex2.py", line 18, in <module>
    for ip in extract_ip_addresses(file_path):
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
NameError: name 'file_path' is not defined
filip@xps-15 → ~/Projects/proxy-shennanigans
> nvim ex2.py
shennanigans/ex2.py", line 18, in <module>
```

```
filip@xps-15 → ~/Projects/proxy-shennanigans
> cat dump | py ex2.py
Traceback (most recent call last):
  File "/home/filip/Projects/proxy-shennanigans/ex2.py", line 4, in extract_ip_addresses
    import sys.stdin
ModuleNotFoundError: No module named 'sys.stdin'; 'sys' is not a valid module name
filip@xps-15 → ~/Projects/proxy-shennanigans
> cat dump | py ex2.py
ips = extract_ip_addresses(file_path)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/home/filip/Projects/proxy-shennanigans/ex2.py", line 4, in extract_ip_addresses
  for line in sys.stdin:
  ^^^^^
NameError: name 'sys' is not defined. Did you forget to import 'sys'?
filip@xps-15 → ~/Projects/proxy-shennanigans
> cat dump | py ex2.py
```

`/[\w._%+-]@[\w.-]+\. [a-zA-Z]{2,4}/`

`filip @ wasp . sh`

`/[\w._%+-]@[\w.-]+\. [a-zA-Z]{2,4}/`

`filip @ wasp . sh`

```
function validateEmail(email) {
  var atSplit = email.split("@");

  // Check if there is only one '@' symbol and the
  // parts are non-empty.
  if (atSplit.length !== 2 || atSplit[0].length === 0
      || atSplit[1].length === 0) {
    return false;
  }

  // Check the domain part.
  var domainParts = atSplit[1].split(".");
  if(domainParts.length < 2) {
    return false;
  }

  const tld = domainParts[domainParts.length - 1];
  if (tld.length < 2 || tld.length > 4) {
    return false;
  }

  // Check if TLD contains only letters
  for(let i = 0; i < tld.length; i++) {
    let charCode = tld.charCodeAt(i);
    if(!(charCode > 64 && charCode < 91) &&
        !(charCode > 96 && charCode < 123)) {
      return false;
    }
  }
  return true;
}
```

```
SELECT *  
  FROM Customer  
  WHERE Age > 25;
```

Total time: 3,903ms

#4

#2

#0

#1

#0

Sort

527ms (13.5%)

4.9 ★

45k

#1

Hash Join

502ms (12.9%)

0.0 ★

45k

#2

Seq Scan - author

941ms (24.1%)

0.0 ★

72k

#3

Hash

181ms (4.6%)

0.0 ★

72k

#4

Seq Scan - article

1,625ms (41.6%)

0.0 ★

#P

Planning

118ms (3.0%)

0.0 ★

SEL
F
WH

```
function HelloMessage({ name }) {  
  return <div>Hello {name}</div>;  
}
```



```
function HelloMessage({ name }) {  
  return React.createElement(  
    "div",  
    null,  
    "Hello ",  
    name  
  );  
}
```

Why DSL #1:

Say **what** you want, not **how**...
... thanks to embedded **knowledge**.

2006

2023



Why DSL #2:

Good DSLs age *really well*.

DSLs

- Allow us to **say what**, not how

DSLs

- Allow us to **say what**, not how
- Built-in **knowledge** and **assumptions**

DSLs

- Allow us to **say what**, not how
- Built-in **knowledge** and **assumptions**
- Tied to the **domain**, not the implementation

DSLs

- Allow us to **say what**, not how
- Built-in **knowledge** and **assumptions**
- Tied to the **domain**, not the implementation
- If they're good, they **age well**

DSLs

- Allow us to **say what**, not how
- Built-in **knowledge** and **assumptions**
- Tied to the **domain**, not the implementation
- If they're good, they **age well**
- For stuff that's **easy to describe**, but **tedious to implement**

Access control

Sessions or tokens

Same-origin policy and
CORS

Cookies or local storage

Choosing Cookie
properties

Connecting to the
database

Third-party Auth

Preventing XSRF

Frontend cache
invalidation

Build pipelines

Best security practices

SEO

Scheduling Jobs

Setting up RPCs

Accessibility

Styling

Database migrations

Deployment

@WaspLang

Let's Plan an App

```
auth {
```

```
}
```

```
auth {  
  methods: []  
  
}
```

```
auth {  
  methods: [  
    usernameAndPassword,  
    google  
  ]  
}
```

```
auth {  
  methods: [  
    usernameAndPassword,  
    google  
  ],  
  userEntity: User  
}
```

```
auth {  
  methods: [  
    usernameAndPassword,  
    google  
  ],  
  userEntity: User,  
  onSuccessRedirectTo: "/"  
}
```



```
app Todo {
  title: "Tech City Pula TodoApp",
  db: PostgreSQL,
  auth: {
    methods: [google, github],
    entity: User
  }
}
```

```
entity User {  
  id:      Int,  
  email:   String  
}
```

```
page Dashboard {  
  route: "/",  
  component: ???  
}
```

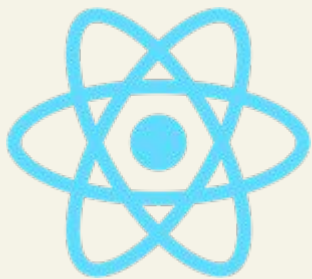
```
page Dashboard {  
  route: "/",  
  component: "pages/Dashboard.jsx"  
}
```

```
app Todo {  
  title: "Tech City Pula ToDo App",  
  db: Postgres,  
  auth: {  
    methods: [usernameAndPassword, google, github],  
    entity: User  
  }  
}
```

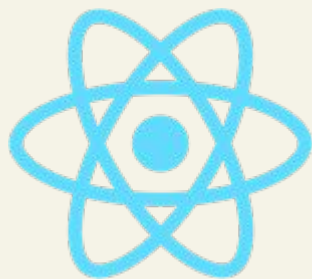
```
page Dashboard {  
  route: "/",  
  component: "pages/Dashboard.jsx"  
}
```

```
entity User {  
  id: Int,  
  email: String  
}
```

This actually works!



Prisma



Prisma

How

SSG

Serverless

SSR

CSR

Edge

What

“This is resource intensive”

“This is a long running operation”

“Optimize for SEO”

```
auth: {  
  methods: {  
    usernameAndPassword: {},  
    google: {},  
    gitHub: {}  
  },  
  onAuthFailedRedirectTo: "/login",  
  userEntity: User,  
}
```

```
import jwt from 'jsonwebtoken'
import SecurePassword from 'secure-password'
import util from 'util'

import prisma from '../dbClient.js'
import { handleRejection } from '../utils.js'
import config from '../config.js'

const jwtSign = util.promisify(jwt.sign)
const jwtVerify = util.promisify(jwt.verify)

const JWT_SECRET = config.auth.jwtSecret

export const sign = (id, options) => jwtSign({ id }, JWT_SECRET, options)
export const verify = (token) => jwtVerify(token, JWT_SECRET)

const auth = handleRejection(async (req, res, next) => {
  const authHeader = req.get('Authorization')
  if (!authHeader) {
    return next()
  }

  if (authHeader.startsWith('Bearer ')) {
    const token = authHeader.substring(7, authHeader.length)

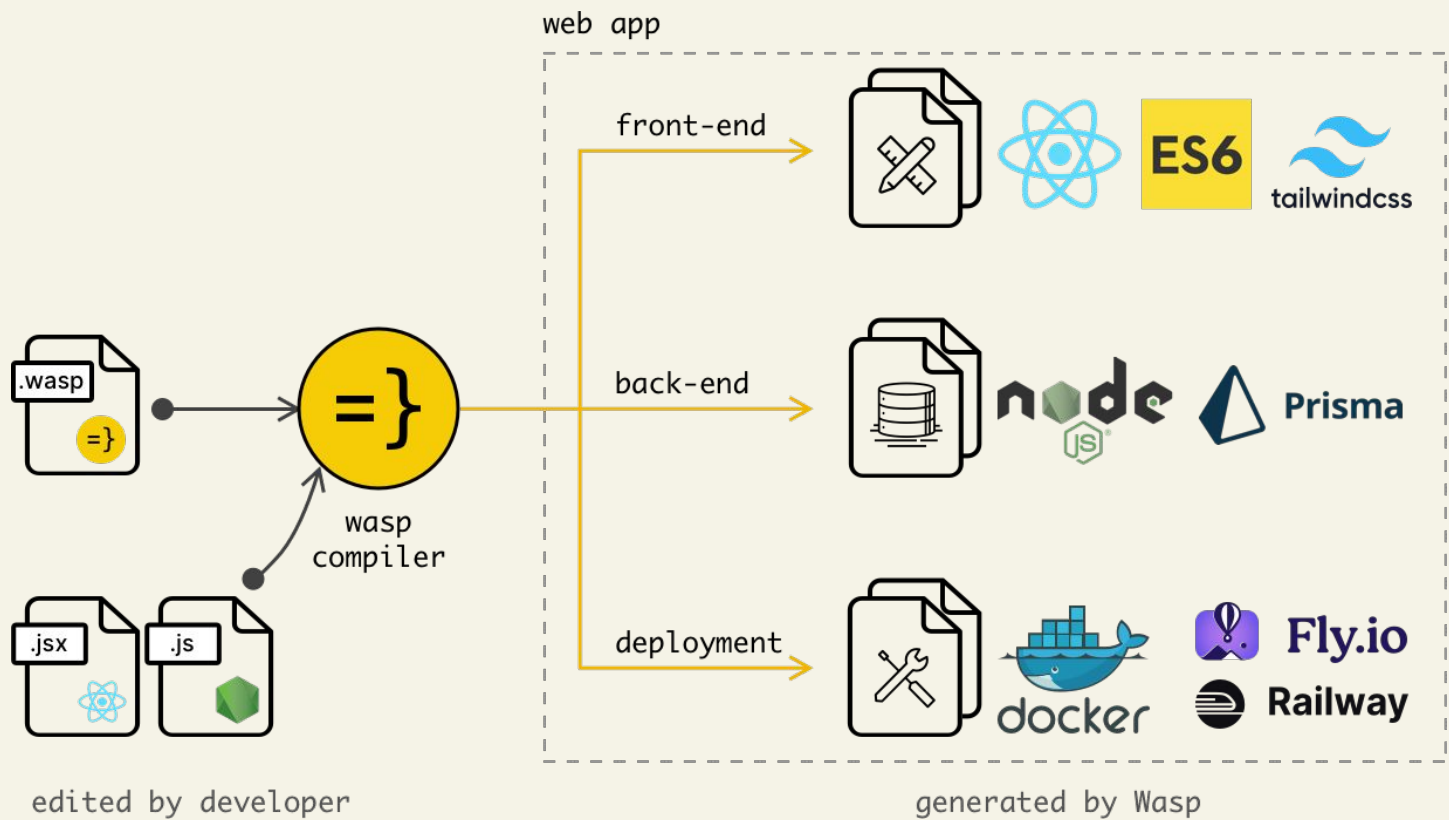
    let userIdFromToken
    try {
      userIdFromToken = (await verify(token)).id
    } catch (error) {
      if (['TokenExpiredError', 'JsonWebTokenError', 'NotBeforeError'].includes(error.name)) {
        return res.status(401).send()
      } else {
        throw error
      }
    }
  }

  const user = await prisma.user.findUnique({ where: { id: userIdFromToken } })
  if (!user) {
    return res.status(401).send()
  }

  const { password, ...userView } = user

  req.user = userView
} else {
  return res.status(401).send()
}

  next()
})
```





The Weird **Startup** World

Wasp

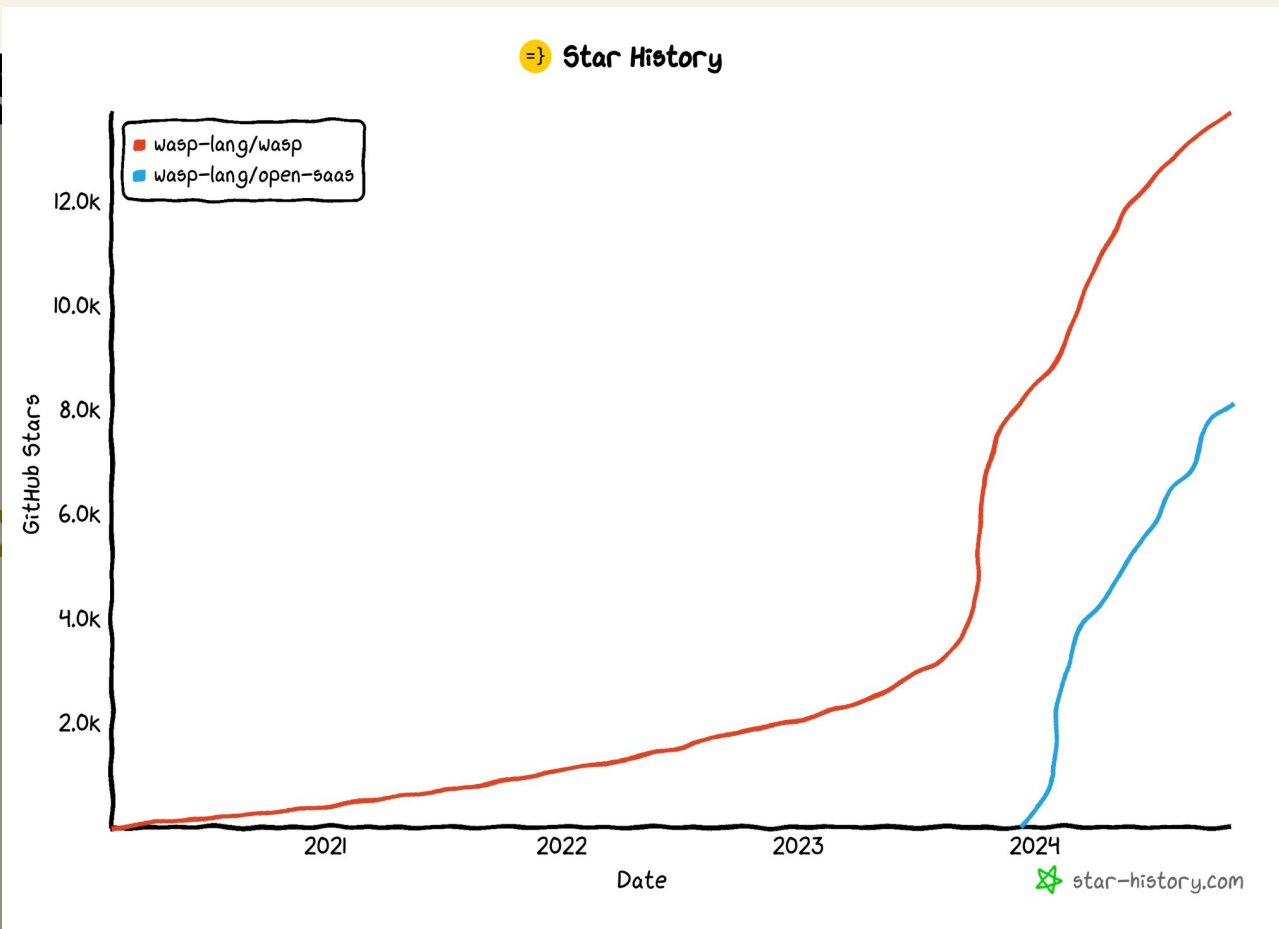
> **21k** GitHub stars
~ **50k** apps created
& **lots** of happy users

Was

> 2



~ 4

& 10



@WaspLang




Samadhi   Today at 02:55

Awesome, thank you guys for all the amazing work you've been doing on Wasp! It's been a game changer for me. I've used pretty much every major web technology over the last 7 years.

IMO Wasp is as game-changing for me as React has been many years back. Wasp's simplicity, and how well the DSL captures most full-stack engineering tasks is pure genius.

I believe Wasp will become the #1 web technology in just a couple years. It has everything that most web devs are looking for.



Samadhi  OP Today at 02:55

Awesome, thank you guys for all the amazing work you've been doing on Wasp! It's been a game changer for me. I've used pretty


IMO Wasp is as game-changing for me as React has been many years back. Wasp's simplicity, and how well the DSL captures most full-stack engineering tasks is pure genius.

full-stack engineering tasks is pure genius.

I believe Wasp will become the #1 web technology in just a couple years. It has everything that most web devs are looking for.

Hacktoberfes t

Update the database docs on seeding the production DB on Fly #1464

 **Open** infomiho opened this issue on Oct 3, 2023 · 15 comments

```
220
221   Check the [API Reference](#cli-commands-for-seeding-the-database) for more details on these commands.
222
223 +
224 + ## Seeding the Production Database on Fly.io
225 +
226 + To seed the production database on Fly.io, follow these steps:
227 +
228 + Certainly! Here's the information you provided in a formatted and organized manner for your GitHub
    documentation:
229 +
230 + ---
231 +
232 + ## Getting the DB Name and Password
233 +
234 + ### Retrieve the DB Name:
235 +
236 + To obtain the DB name, use the following command:
237 +
238 + ```bash
239 + fly postgres connect -a <app_name>
240 + ```
241 +
242 + Replace `<app_name>` with the name of your Wasp app. After connecting to the database, run:
243 +
244 + ```bash
245 + \l
246 + ```
247 +
248 + The DB name will be displayed in the list of databases.
249 +
250 + ### Retrieve the DB Password:
251 +
252 + To get the DB password, run the following command:
253 +
254 + ```bash
255 + fly ssh console -a <app_name>
256 + ```
257 +
258 + Once connected to the database server, execute:
259 +
260 + ```bash
```

```
220
221   Check the [API Reference](#cli-commands-for-seeding-the-database) for more details on these commands.
222
223 +
224 + ## Seeding the Production Database on Fly.io
225 +
226 + To seed the production database on Fly.io, follow these steps:
227 +
228 + Certainly! Here's the information you provided in a formatted and organized manner for your GitHub
229 + documentation:
230 + ---
231 +
232 + ## Getting the DB Name and Password
233 +
234 + ### Retrieve the DB Name:
235 +
236 + To obtain the DB name, use the following command:
237 +
238 + ```bash
239 + fly postgres connect -a <app_name>
240 + ```
241 +
242 + Replace `<app_name>` with the name of your Wasp app. After connecting to the database, run:
243 +
244 + ```bash
245 + \l
246 + ```
247 +
248 + The DB name will be displayed in the list of databases.
249 +
250 + ### Retrieve the DB Password:
251 +
252 + To get the DB password, run the following command:
253 +
254 + ```bash
255 + fly ssh console -a <app_name>
256 + ```
257 +
258 + Once connected to the database server, execute:
259 +
260 + ```bash
```

Add monetization model to the FAQ section of the landing page #856

Edit



maksim36ua opened this issue on Nov 28, 2022 · 2 comments



maksim36ua commented on Nov 28, 2022

Contributor ...

Wasp's monetization model and cash flow are not clear enough. Users might have a suspicion that they would get a limited version of the framework. Or would have to pay at the late stage of the dev process.

Yeah. If you have a company with 6ppl and have backing from Y Combinator... I assume you need to get in 💰 somehow. Understanding your monetization strategy would help for me and many more i would think. But you need to write more than free and open source


It's proposed to add an answer to this question in the FAQ section of the landing page.



maksim36ua added **good first issue** **no-haskell** **webdev** labels

on Nov 28, 2022



 commented 29 days ago



I am ready to work on it; please provide me with some additional clarity.



Discord

- Loyal and useful users
- Weird users
- Users insulting each other



lamlconLiving 🏀 08/03/2024 7:27 PM

**SETUP TRPC,
NEXTJS, CRON, DB, MAILING**

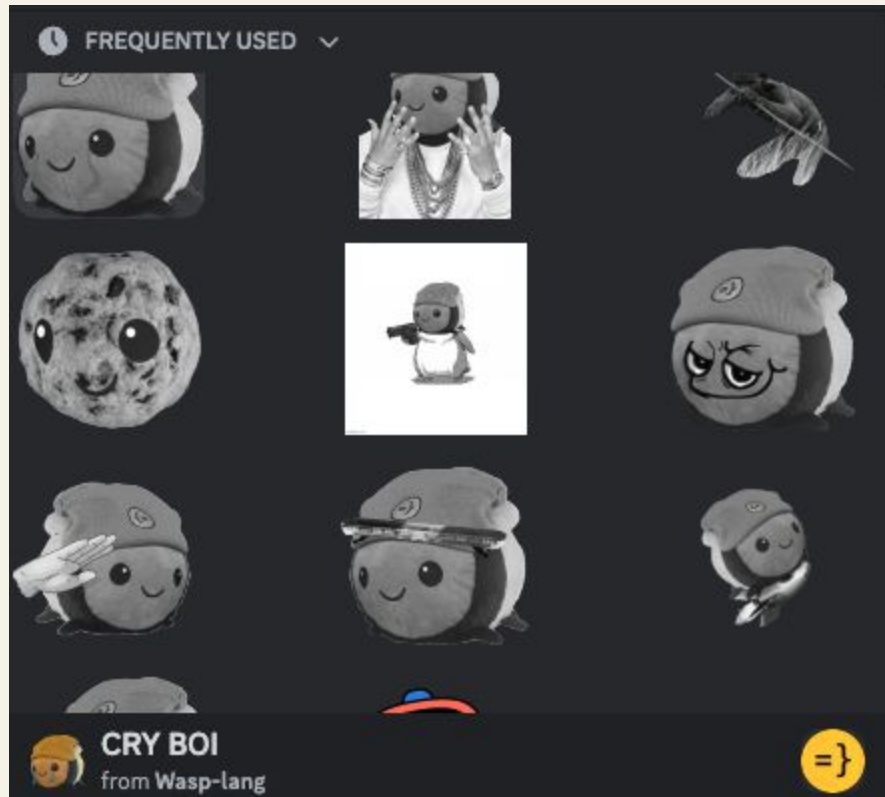


**WASP NEW
WASP DEPLOY**



👤 4 😂 1 🤔 2 😊

@WaspLang



Reddit



r/CroIT · 8 mo. ago

UnfittinglyFit



Shoutout našim momcima koji su kreirali Wasp framework

Original Content

r/webdev · 8 mo. ago

Our full-stack web framework reached 10,000 stars on GitHub!

TL;DR - 3 years ago, my brother and I had an idea to start a new web framework, powered by its own DSL/compiler. A lot of people told us we stand no chance (we weren't sure either tbh), but we kept going. **Today, we reached 10,000 stars on GitHub** - below is a short story about how we started and what it took to get here.



[Redacted] · 3 yr. ago

I think a lot of us just setup our own tools?

I'll check it out, but I think you'll be chasing your tail on this. Shrug.



3



Reply

Give Award

Share

Report

Save

Follow



8mo ago •



Top Commenter

Bravo bojsi! Super je vidjeti projekt kojem je fokus na dobrom proizvodu nastalom iz zabave/značiželje/izazova.



15



Reply



matijash • 8mo ago •

Hvala na lokalnom supportu ljudi! :)



12



Reply





• 8mo ago •

Jos jedan frontend framework je upravo ono sto treba ovom svijetu jer nemamo dovoljno postojecih i problem pracenja razvoja s postojecima /s



• 8mo ago •

Jos jedan frontend framework je upravo ono sto treba ovom svijetu jer nemamo dovoljno postojecih i problem pracenja razvoja s postojecima /s



-14



Reply





8mo ago •

Jos jedan frontend framework je upravo ono sto treba ovom svijetu jer nemamo dovoljno postojećih i problem pracenja razvoja s postojećima /s



-14



Reply



• 8mo ago •

Još jedna osoba koja sere bez da je uopće pročitala o čemu se radi



14



Reply



8 more replies



[redacted] 8mo ago •

Jos jedan frontend framework je upravo ono sto treba ovom svijetu jer nemamo dovoljno postojećih i problem pracenja razvoja s postojećima /s

⊖ ↑ -14 ↓ ○ Reply ...



[redacted] • 8mo ago •

Još jedna osoba koja sere bez da je uopće pročitala o čemu se radi

↑ 14 ↓ ○ Reply ...

⊕ 8 more replies



[redacted] ago •

Jesi pogledao repo prije nego si napisao komentar i zašto baš ne? Ako taj "još jedan" framework popravi problem kojeg ima 10k+ usera, onda nije "još jedan" nego "e ovo smo trebali" framework.

Svaka čast ekipi, bravo!

↑ 17 ↓ ○ Reply ...




[redacted] • 8mo ago •

tipicno hreddit serendanje i zavist jalne stoke

↑ 0 ↓ ○ Reply ...

⊕ 2 more replies

 This is a Wasp powered project. If you like it, [star us on GitHub!](#)

 Star



GPT Web App Generator ✨

Generate your full-stack web app in Wasp, React, Node.js and Prisma

App name *

e.g. TodoApp or FlowerShop

Description *

The *free* SaaS template with superpowers

An open-source, feature-rich, full-stack React + NodeJS template that manages features for you.

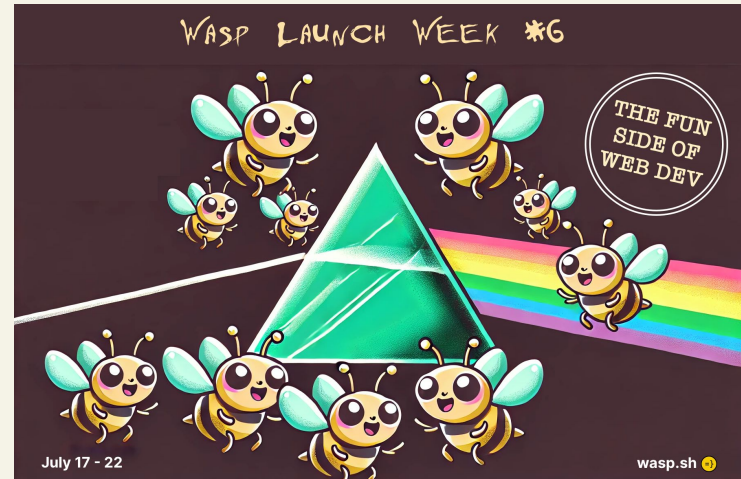
Get Started →

View the Repo

 3960



Riding the hype



Creepy stuff



Open Source

This is the way. Wasp is fully open-source and you're welcome to contribute!

[Learn more →](#)



Full-stack Auth

Add login with social providers or email in a few lines of code with powerful UI helpers. No third party vendor lock-in.

[Learn more →](#)



RPC (Client ↔ Server)

Wasp provides a typesafe RPC layer that instantly brings your data models and server logic to the client.

[Learn more →](#)



Simple Deployment

Deploy your app to any platform. Wasp offers CLI helpers for the most popular options.

[Learn more →](#)



Jobs

Easily define, schedule and run specialized server tasks. Persistent, retryable, delayable.

[Learn more →](#)



Email Sending

All you need to do is connect an email provider and you can send emails!



Full-stack Type Safety

Full support for TypeScript with auto-generated types that span the whole stack.



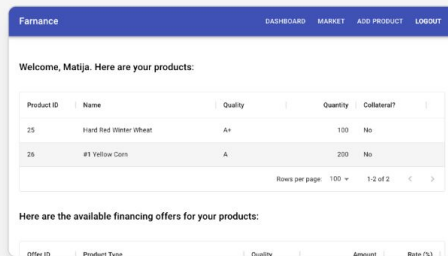
And More!

Custom API routes, database seeding, optimistic updates, automatic cache invalidation on the client, ...

Everyone's making money... except us

🏆 Showcase Gallery 🏆

See what others are building with Wasp.



Farnance: SaaS marketplace for farmers

hackathon

material-ui

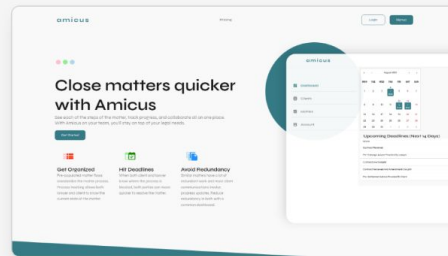
See how Julian won HackLBS 2021 among 250 participants and why Wasp was instrumental for the team's victory.



Grabbit: Easily manage dev environments

internal-tools

See how Michael built and deployed an internal tool for managing dev resources at StudentBeans.



Amicus: Task and workflow management for legal teams

startup

material-ui

See how Eris rolled out fully-fledged SaaS as a team of one in record time and got first paying customers.

Gimmicks

- Giveaways
- Ruffles
- Farming stars
- Old-school arcade



Bok Matija,

Danas ti se vratila pošiljka koju ste slali u Indiju.

Kao što možeš vidjeti na priloženoj fotografiji, prošla je kroz sito i rešeto, proslavila Diwali, bila u borbi s Bengalskim tigrom i vratila se doma :D

Lijep pozdrav,
Jan Popović



JAN POPOVIĆ

Tel: +385 91 7946 384

[Petračićeva 6, 10000 Zagreb](#)

facebook.com/hub385

twitter.com/hub385

One attachment • Scanned by Gmail ⓘ



PRIOTAIRE / PAR AVION
Prioritarno / Zrakoplovom

D.F.T
11/06/22

270

R RG 10 499 483 5 HR

(LPS)
(UG)

शक्ति टाकी (बम)
ता 14 2200 1000
ता 14 2200 1000
13/06/22

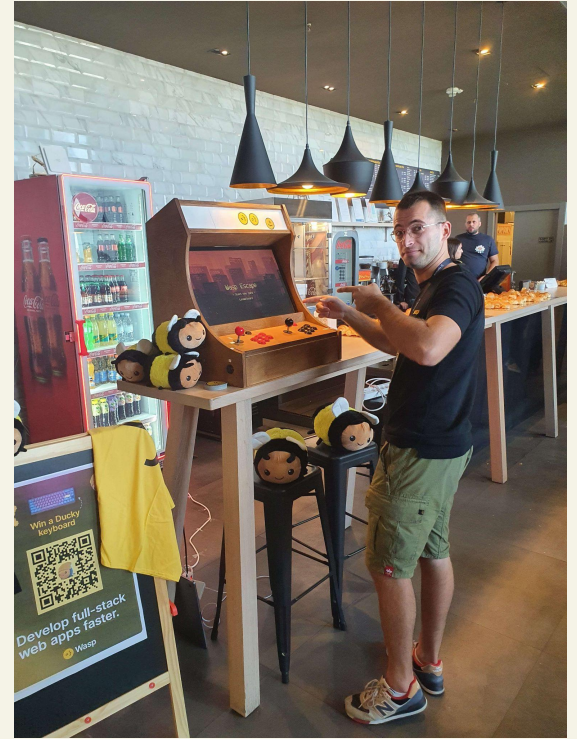
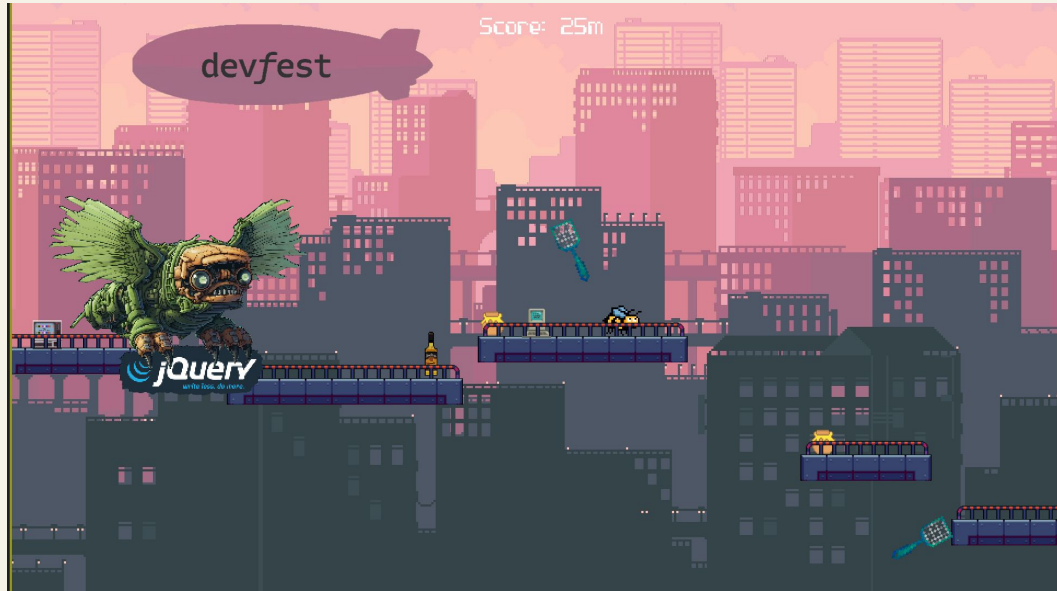
[Handwritten signature]

Registered
11/06/22
16/06/22

[Handwritten signature]

SHORAY SHURAV SINGH
KRISHNA BATTERIES
RELIANCE TRENDS, BAHJOL
CHANDERBI DIST, SAMBAL
UTTAR PRADESH, INDIA

[Handwritten signature]
INDIA



Thank


you!



@WaspLang



CHECK US OUT!

A hand-drawn black arrow with a circular tail and a pointed head, pointing from the text 'CHECK US OUT!' towards the QR code.

Extra: LLMs

@WaspLang



Paul Graham  @paulg · 3h



I expected technology to make programming less laborious, as it does to most things. But I have to admit I expected it to happen by programmers switching to more powerful languages, rather than continuing to write programs full of boilerplate, but having AIs generate most of it.

 81

 105

 916





Paul Graham  @paulg · 3h



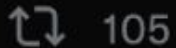
more powerful languages



Als generating tons boilerplate



81



105



916



Always has been

Wait it's all still my code I have
to maintain?

```
49 f_eps[best_idx] < self.best_f:
50 self.best_f = f_eps[best_idx]
51 self.best_x = (self.x1 - self.sigma * d_eps[best_idx, :]).cl
52
53 self.is_mirror:
54 f_eps_mirror = np.apply_along_axis(self.fct, 1, (self.x1 - se
55
56 # check for best solution
57 best_idx = np.
58 if f_eps[best_
59 self.best_
60 self.best_
61
62 f_eps = (f_eps
63
64 eps = (f_eps - n
65 l_eps = np.mean(f
66 scrt (nd_eps.sh
67 @ add docstring
```



GitHub
Copilot



Why DSL #3:

An **ideal** abstraction for **AI**

Why DSL #3:

~~An **ideal** abstraction for **AI**~~

An **ideal** abstraction for **people**

Why DSL #3:

~~An **ideal** abstraction for **AI**~~

~~An **ideal** abstraction for **people**~~

An **ideal** abstraction for **people using AI**



GPT Web App Generator ✨

Generate your full-stack web app in Wasp, React, Node.js and Prisma

App name *

e.g. TodoApp or FlowerShop

Description *

✨ **useMAGE.ai** ✨