



InNeed Shopping List



Luzia Kündig, David Kalchofner, Jamie Maier



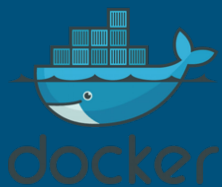
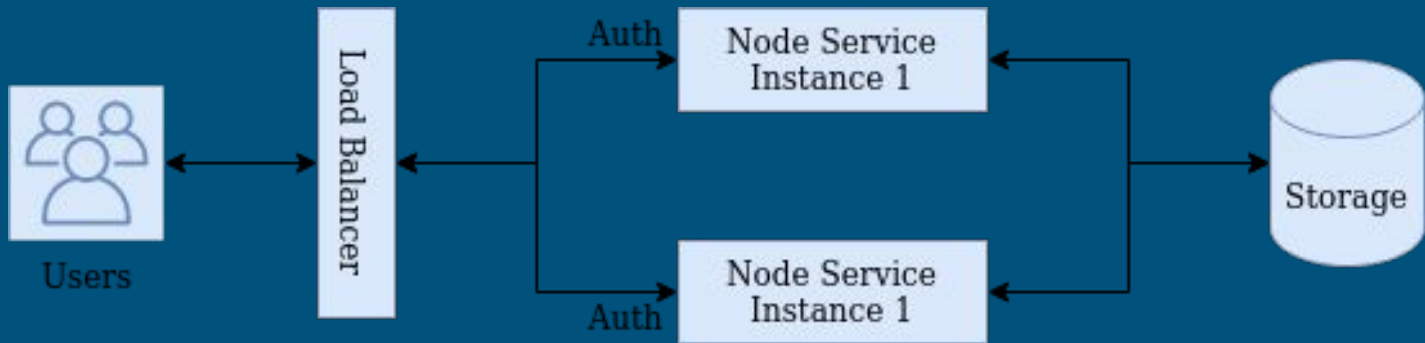
Inhalt

- Idee und Plan
- Technologien
- Architektur
- Implementation der Requirements
 - Simple Frontend
 - Load Balancer and 2 Service Instances
 - Storage Backend
 - Authentication with JWT
- Fazit

Idee und Plan

- Erste Idee: Statistiken Darstellen
 - Schwierig zu implementieren, zu wenig geplant / angeschaut
- Zweite Idee: Einkaufsliste
 - Besserer Use-Case, konnten dies besser modellieren und implementieren
- Aufteilung
 - David: Docker Setup, Node Setup & DB
 - Luzia: Frontend
 - Jamie: Authentication

Architektur & Technologies



Implementation - Frontend



Homepage / Login:

index.html

Shopping lists:

overview.html

Single List:

detailview.html

InNeed Home Lists Logout

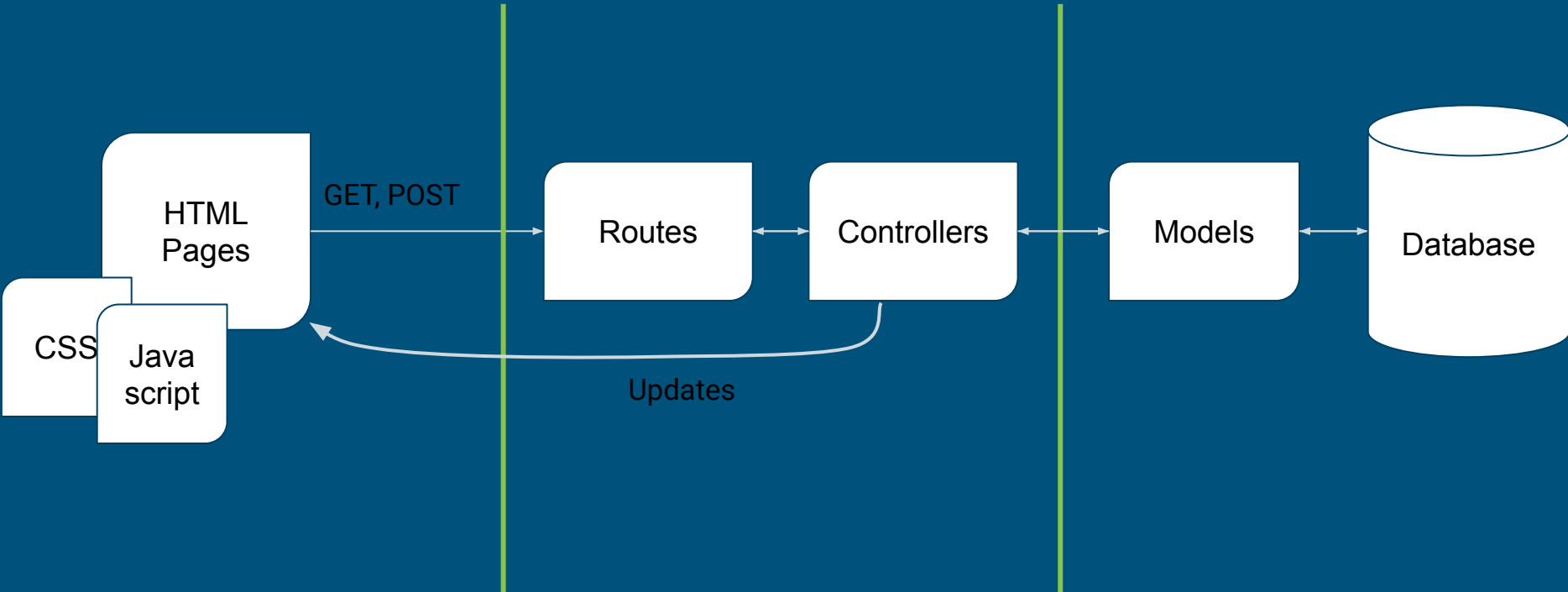
Example Shopping List

Add a new Product

Product Name

Product	Amount	Got It?	Options
Apples	2	☐	Remove
Chocolate	1	☐	Remove
Flour	1	☐	Remove
Coffee	1	☐	Remove

App-Struktur



Implementation - Load Balancer



- Traefik
 - Default Configs -> Round Robin
 - Statische Config, ohne Dashboard
- User gelangt bei Seitenwechsel auf andere Service Instanz
 - Mussten Checklist speichern beim klicken, sonst könnte Wert verloren gehen

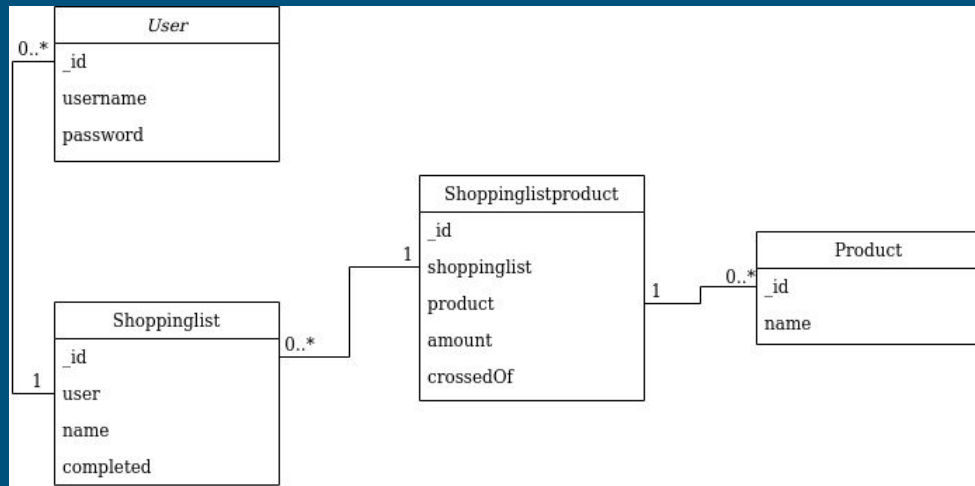
```
1  [entryPoints.web]
2  address = ":80"
3
4  [providers.file]
5  filename = "traefik.toml"
6
7  [http.routers.in-need]
8  rule = "PathPrefix(`/`)"
9  entrypoints = ["web"]
10 service = "in-need"
11
12 [[http.services.in-need.loadBalancer.servers]]
13 url = "http://service:8080"
14
15 [accessLog]
16
17 [log]
18 level = "INFO"
```

Implementation - Storage Backend



mongoose

- MongoDB - mongoose
- Modellierung der Daten
 - Simple aber effektiv Struktur
- Seeden für Dummy Daten
 - Custom Implementation
 - Nur falls DB leer ist (first start)



Implementation - Authentication

```
var token = jwt.sign({ id: user.id }, config.secret, {  
  expiresIn: 86400 // 24 hours  
});
```

- JsonWebToken sign() & verify()

Token wird bei jedem sign in neu generiert. Expiration time: 24h

Überprüft bei jedem /lists und /lists/detailview Aufruf

- Bcryptjs um die Passwörter
in der DB abzulegen

```
jwt.verify(token, config.secret, (err, decoded) => {  
  if (err) {  
    return res.status(401).render('index', { message: "Unauthorized!" });  
  }  
  console.log('authorized');  
  req.userId = decoded.id;  
});
```

Fazit

David

Jamie

Luzia