

# OINK



## Factsheet

### ▶ Team

- ▶ Tim Burkei
- ▶ Eric Bernet
- ▶ Simon Arndt
- ▶ Moritz Renner

### ▶ 4. Semester MI bzw. MM

### ▶ Umsetzung/Lösung

- ▶ Django-Framework
- ▶ Python
- ▶ VUE.JS
- ▶ Asynchrone Web Scrapper

### ▶ Motivation/Problemstellung

- ▶ Börsensimulation
- ▶ Mit echten Daten seine Trading-Strategien testen
- ▶ Portfolio aufbauen
- ▶ Multiplayer-Funktion
  - ▶ Mit freunden Erfolg vergleichen



**django**





# OINK

## ▶ Besondere Probleme

- ▶ Erstes mal Django und Python (eingewöhnen)
- ▶ Neue Technologie kennenlernen
- ▶ Erste mal mit API-Abfragen zu arbeiten
- ▶ Webscrapping sehr fragil
- ▶ IP-Blocking bei zu vielen Aufrufen
- ▶ Ergebnisvalidierung von Webresultaten
- ▶ Performance Probleme

```
async def download_image(url: str, symbol: str):
    http_code, image = await fetch_url(url=url)
    if http_code == 200:
        image_path = await save_img(image_content=image, symbol=symbol)
    return image_path

± sa094 +1
async def crawl_wikipedia(isin: str = None, name: str = None):
    logo_href = None
    wiki_href = None

    if name is not None:
        url = await wikipedia_name_search_url(name=name)
        http_code, html = await fetch_url(url)
        if http_code == 200:
            wiki_href, logo_href = await wikipedia_parser(html=html, search_query=name, url=url)
```



# OINK

## ▶ Funktionen

- ▶ Login
- ▶ Aktien Suchfunktion
- ▶ Verschiedene Aktien anschauen und kaufen
- ▶ Portfolio aufbauen
- ▶ Gewinn/Verlust mit echten, tagesaktuellen Daten simulieren
- ▶ Freunde hinzufügen
- ▶ Mit freunden messen

