

백엔드 스쿨

‘본질’에 집중하는 개발자를 양성하는
백엔드 온라인 취업 교육 과정



ZEROBASE SCHOOL



BACK-END

카카오뱅크
김 하 은

안녕하세요. 카카오뱅크 김하은입니다.

현) 카카오뱅크 서버 개발자

카카오뱅크
김 하 은



Chapter

- 01 챗터내용
- 02 챗터내용
- 03 챗터내용
- 04 챗터내용
- 05 챗터내용
- 06 챗터내용
- 07 챗터내용

스노우볼
김 제 로

01 프로젝트 진행 전

01 주제 설명

Chapter 01

01 주제 설명



우리가 진행할 프로젝트는

주식 배당금 API 서비스



Chapter 01

주제 설명



주식이란

회사의 자본을 구성하는 단위

기업의 자본금 = 1주당 액면가 * 발행 주 수

Chapter 01

주제 설명

카카오뱅크 323410 코스피 2022.03.04 기준(정마감) [실시간](#) [기업개요](#)

[+ MY STOCK 추가](#) [빠른주문](#)

48,700

전일대비 ▲100 +0.21%

전일 48,600 고가 50,300 (상한가 63,100) 거래량 1,923,007
 시가 48,350 저가 48,350 (하한가 34,050) 거래대금 94,415 백만

선차트 1일 1주일 3개월 1년 3년 5년 10년

봉차트 일봉 주봉 월봉



투자정보 호가10단계

시가총액	23조 1,403억원
시가총액순위	코스피 14위
상장주식수	475,159,237
액면가 매매단위	5,000원 1주
외국인한도주식수(A)	475,159,237
외국인보유주식수(B)	77,078,722
외국인소진율(B/A)	16.22%
투자이건 목표주가	3.86배수 57,167
52주최고 최저	94,400 39,550
PER EPS(2020.12)	157.61배 309원
추정PER EPS	72.58배 671원
PBR BPS(2020.12)	4.22배 11,531원
배당수익률	N/A
동일업종 PER	4.61배
동일업종 등락률	-0.88%

Chapter 01 주제 설명

배당금이란

회사의 이익을 주주들에게 배분해주는 금액



Chapter 01

주제 설명

배당금은

당연하지만 회사마다 다르다..!

Chapter 01

주제 설명

주식 배당금 API

코카콜라에서는 얼마의 배당금을 줬을까
나이키의 배당금 추이는 어떻게 될까
애플은 언제 마다 배당금을 줬을까

Chapter 01

주제 설명

백엔드 스쿨

‘본질’에 집중하는 개발자를 양성하는
백엔드 온라인 취업 교육 과정



ZEROBASE SCHOOL



BACK-END

카카오뱅크
김 하 은

Chapter

- 01 챗터내용
- 02 챗터내용
- 03 챗터내용
- 04 챗터내용
- 05 챗터내용
- 06 챗터내용
- 07 챗터내용

스노우볼
김 제 로

01 프로젝트 진행 전

02 요구사항 정리

Chapter 01

02 요구사항 정리



요구사항 정리

- 내가 원하는 회사의 배당금 정보를 보고 싶어요
- [회사]를 저장/삭제 관리할 수 있는 기능
- 회사에 해당하는 [배당금] 정보를 제공하는 기능

Chapter 01

02 요구사항 정리

요구사항 정리

- [회사]를 저장/삭제 관리할 수 있는 기능
 - ✓ 회사 정보를 저장하는 API
 - ✓ 저장된 회사 정보를 삭제하는 API
 - ✓ 회사 정보를 관리하기 위한 DB 테이블 설계

Chapter 01
02 요구사항 정리

요구사항 정리

- 회사에 해당하는 [배당금] 정보를 제공하는 기능
 - ✓ 회사의 배당금 정보를 조회하는 API
 - ✓ 회사의 배당금 정보를 스크래핑
 - ✓ 회사 배당금 정보를 관리하기 위한 DB 테이블 설계

Chapter 01

02 요구사항 정리

왜 스크래핑을 하나요?

스크래핑의 활용 분야는 정말 다양하다!

쇼핑몰, 금융, 업무 자동화 등등 ...

Chapter 01

02 요구사항 정리

정리 해보면

- ✓ 스크래핑으로 배당금 데이터 가져오기
- ✓ 서비스 제공을 위한 API 개발하기
- ✓ 서버 부하를 줄이기 위한 캐시 서버 구성하기

Chapter 01

02 요구사항 정리

02 서비스 설계

01 API 인터페이스 설계

Chapter 02

01 API 인터페이스 설계

이제 우리가 할 것들

- API 인터페이스 설계
- DB 설계

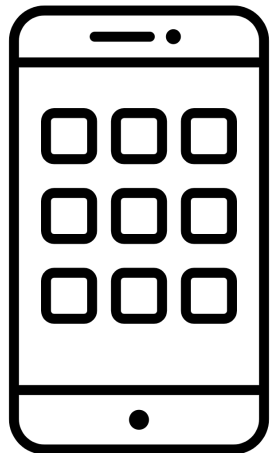
Chapter 02

01 API 인터페이스 설계

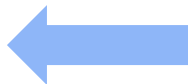
API 인터페이스 설계

Chapter 02

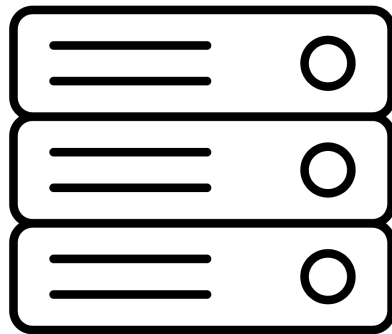
01 API 인터페이스 설계



oo 회사의 배당금 데이터를 줘!



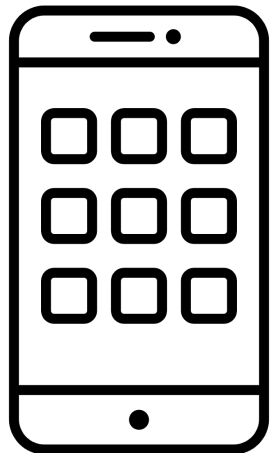
알겠어!



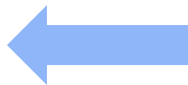
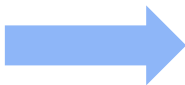
API 인터페이스 설계

Chapter 02

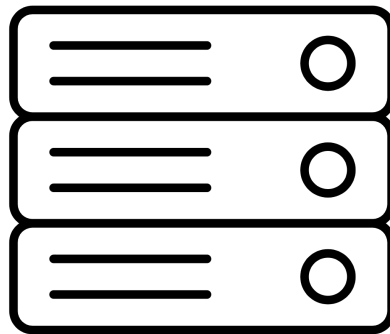
01 API 인터페이스 설계



GET /dividend/{company}

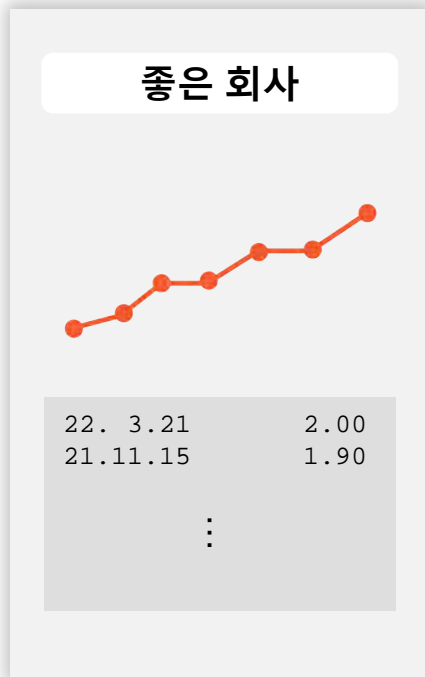


```
{  
  companyName: ...,  
  dividends: [ ... ]  
}
```



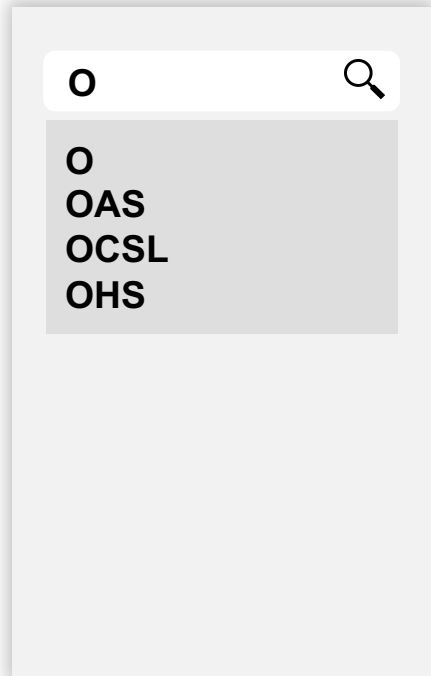
특정 회사의 배당금 조회

- GET
- /finance/dividend/{companyName}
- {
 companyName: “좋은 회사”,
 dividend: [
 {
 date: “2022. 3. 21”
 price: “2.00”
 }, ...
]
}



배당금 검색 - 자동완성

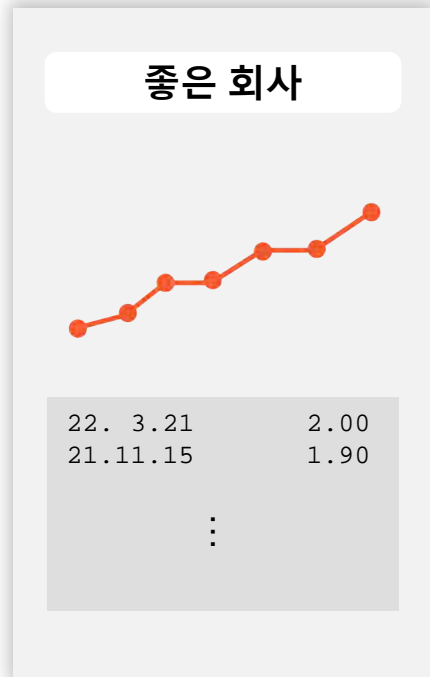
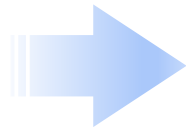
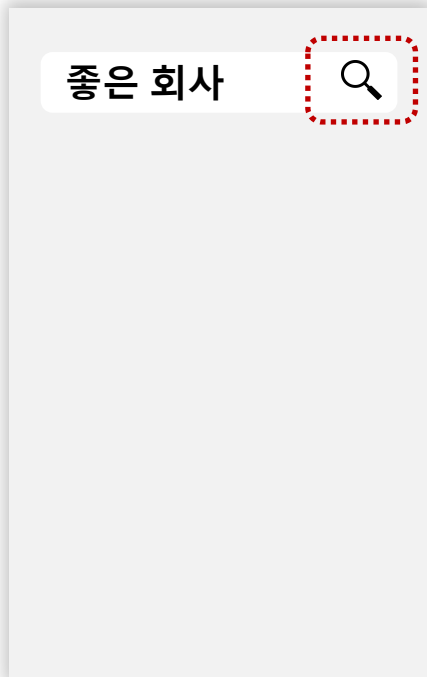
- GET
- /company/autocomplete?keyword=O
- {
 result: ["O", "OAS", ...]
}



Chapter 02

01 API 인터페이스 설계

배당금 검색



Chapter 02

01 API 인터페이스 설계

02 서비스 설계

02 API 인터페이스 설계 02

Chapter 02

02 API 인터페이스 설계

회사 리스트 조회

- GET
• /company
- {
 result: [
 {
 companyName: “좋은 회사”,
 ticker: “GOOD”
 }, ...
]
}

좋은 회사

우리 회사

더 좋은 회사

OO 회사

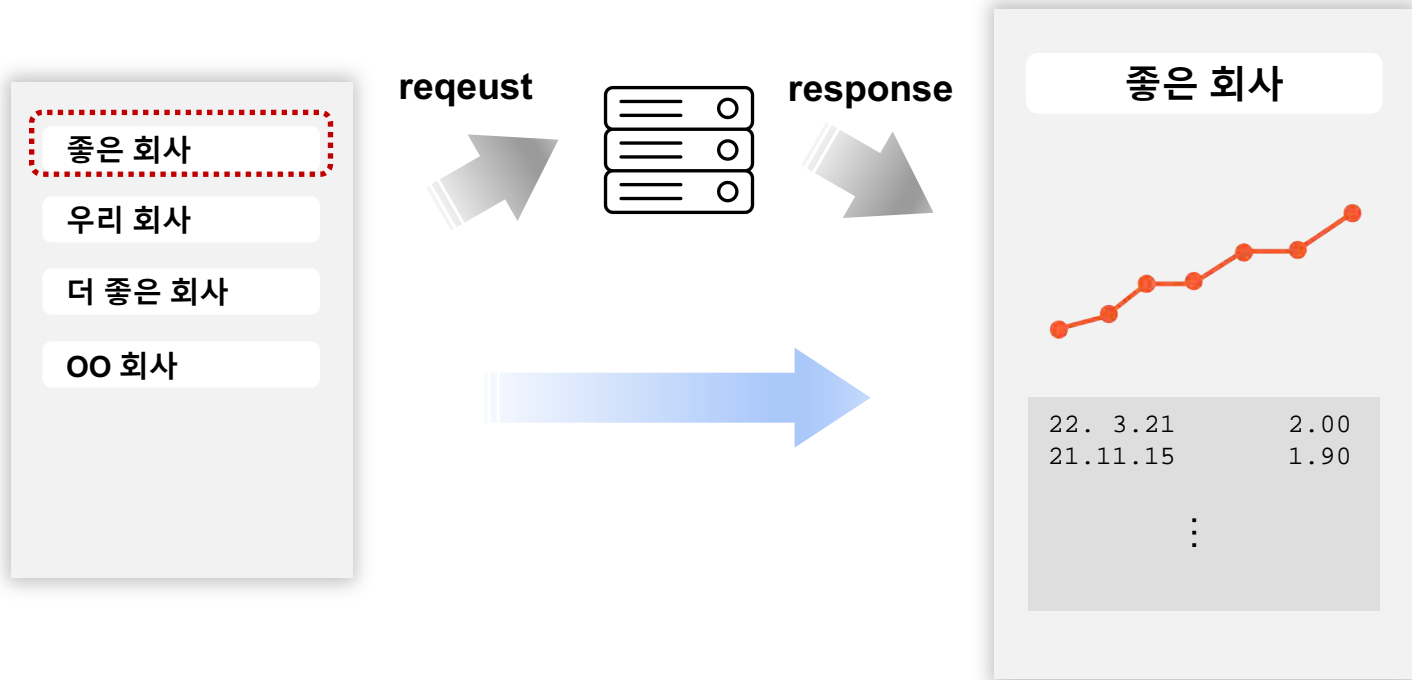
Chapter 02

02 API 인터페이스 설계

회사 리스트 조회

Chapter 02

02 API 인터페이스 설계



관리자 기능 - 배당금 저장

- POST
- /company
- { ticker: "GOOD" }
- {
 ticker: "GOOD",
 companyName: "좋은회사", ...
}

Chapter 02

02 API 인터페이스 설계

관리자 기능 - 배당금 삭제

- DELETE
- /company?ticker=GOOD

Chapter 02

02 API 인터페이스 설계

회원 API

- 회원가입
- 로그인
- 로그아웃

Chapter 02

02 API 인터페이스 설계

서비스 API

- 배당금 조회
- 배당금 검색 - 자동완성
- 회사 리스트 조회
- 배당금 데이터 저장
- 배당금 데이터 삭제
- TODO 회원 인증

Chapter 02

02 API 인터페이스 설계

02 서비스 설계

04 DB 설계

Chapter 02

04 DB 설계

이제 우리가 할 것들

- ~~API 인터페이스 설계~~
- DB 설계

Chapter 02
04 DB 설계

DB 설계에 고려할 수 있는 것들

Chapter 02

04 DB 설계

- 어떤 타입의 데이터가 저장되는지
- 데이터의 규모는 어떻게 되는지
- 데이터의 저장 주기는 어떻게 되는지
- 데이터의 읽기와 쓰기의 비율
- 속도 vs 정확도
- READ 연산시 어떤 컬럼을 기준으로 읽어오는 지(인덱스)
- 키는 어떻게 생성 해줄건지
- 예상 트래픽은 어느 정도인지
- 파티션은 어떻게 구성할 건지
- 등등등 ...



회사

Chapter 02

04 DB 설계

column	type	unique	example
id	Long	✓	1
name	String		Coca-Cola
ticker	String	✓	COKE

배당금

Chapter 02

04 DB 설계

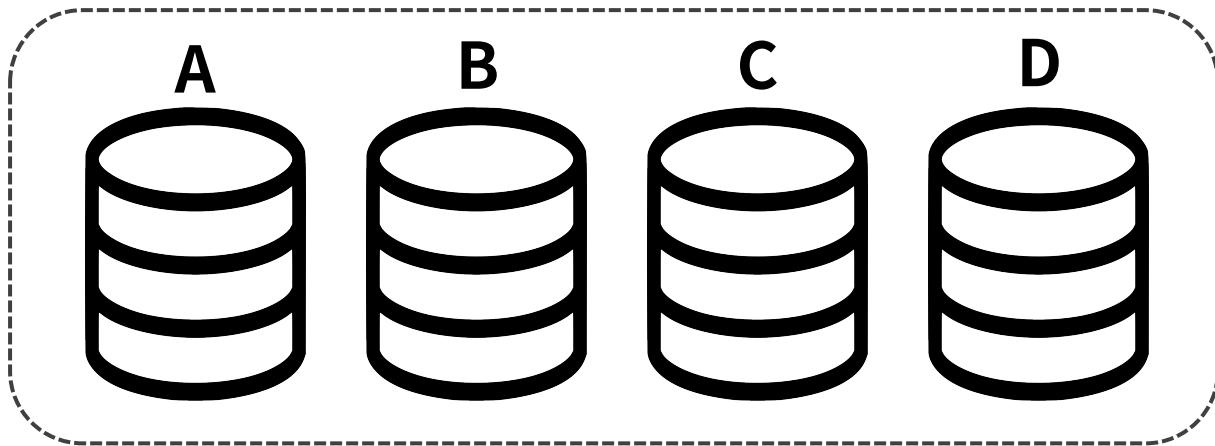
column	type	unique	example
id	Long	✓	3
company_id	Long		1
date	LocalDateTime		2022-05-05
dividend	String		2.00

분산 데이터베이스

Chapter 02

04 DB 설계

request ↓ ↑ response



03 서비스 구현

01 배당금 저장 기능 구현

Chapter 03

01 배당금 저장 기능 구현

구현 동작

1. 인풋으로 저장할 회사의 ticker 를 받는다
2. 이미 저장 되어있는 회사의 ticker 일 경우 오류 처리
3. 받은 ticker 의 데이터를 야후 파이낸스에서 스크래핑 한다
4. 스크래핑 데이터가 조회되지 않는 경우 오류 처리
5. 스크래핑한 회사의 메타 정보와 배당금 정보를 각각 DB 에 저장한다
6. 저장한 회사의 메타 정보를 응답으로 내려준다

Chapter 03

01 배당금 저장 기능 구현

강의 객체

과목번호 1452

과목명 자바

강사명 김자바

=?

과목번호 1452

과목명 자바

강사명 박파이

Chapter 03

01 배당금 저장 기능
구현

Builder Pattern – Coffee

```
class Coffee {  
    int shots;  
    int water;  
    int milk;  
    int syrup;  
    boolean whippedCream;  
}
```

```
Coffee americano = new Coffee(1, 150, null, 1, false);
```

Chapter 03

01 배당금 저장 기능
구현

Builder Pattern – Coffee

Chapter 03

01 배당금 저장 기능 구현

```
Coffee americano = new Coffee(1, 150, null, false, 1);
```

```
Coffee americano = Coffee.builder()  
    .shots(1)  
    .water(250)  
    .syrup(1).build();
```

```
Coffee latte = Coffee.builder()  
    .milk(200)  
    .shots(1)  
    .water(50)  
    .whippedCream(true)  
    .syrup(2).build();
```

03 서비스 구현

02 배당금 저장 기능 구현

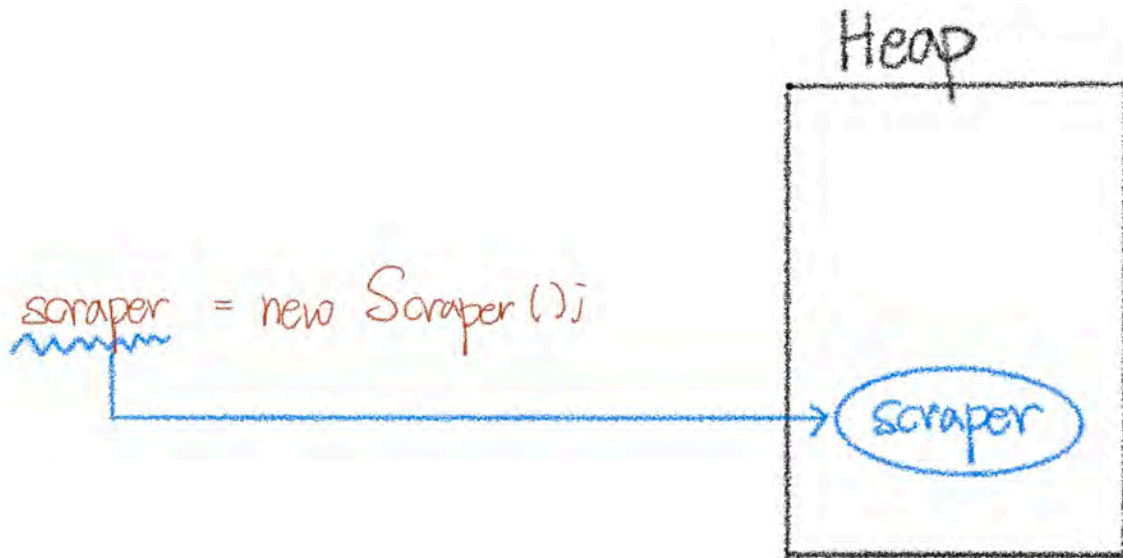
Chapter 03

02 배당금 저장 기능 구현

자바 메모리

Chapter 03

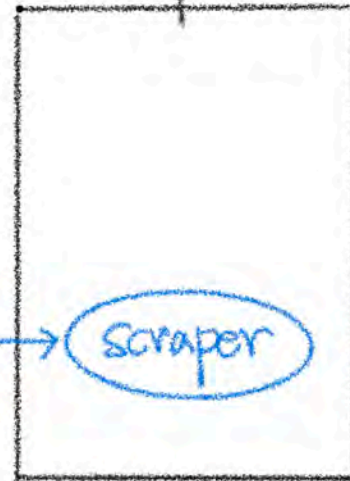
02 배당금 저장 기능
구현



scraper = new Scraper();



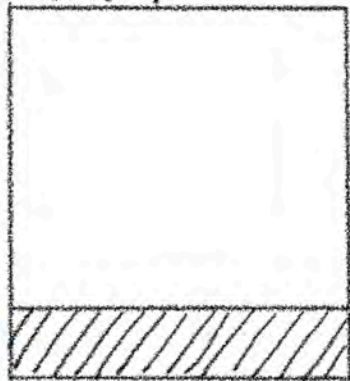
Heap



Scraper.scrap();



Stack



Chapter 03

02 배당금 저장 기능 구현

Heap



scraper = new Scraper();



Stack



Scraper.scrap();

Scraper.scrap();



Chapter 03

02 배당금 저장 기능 구현

Heap



scraper = new Scraper();



Stack



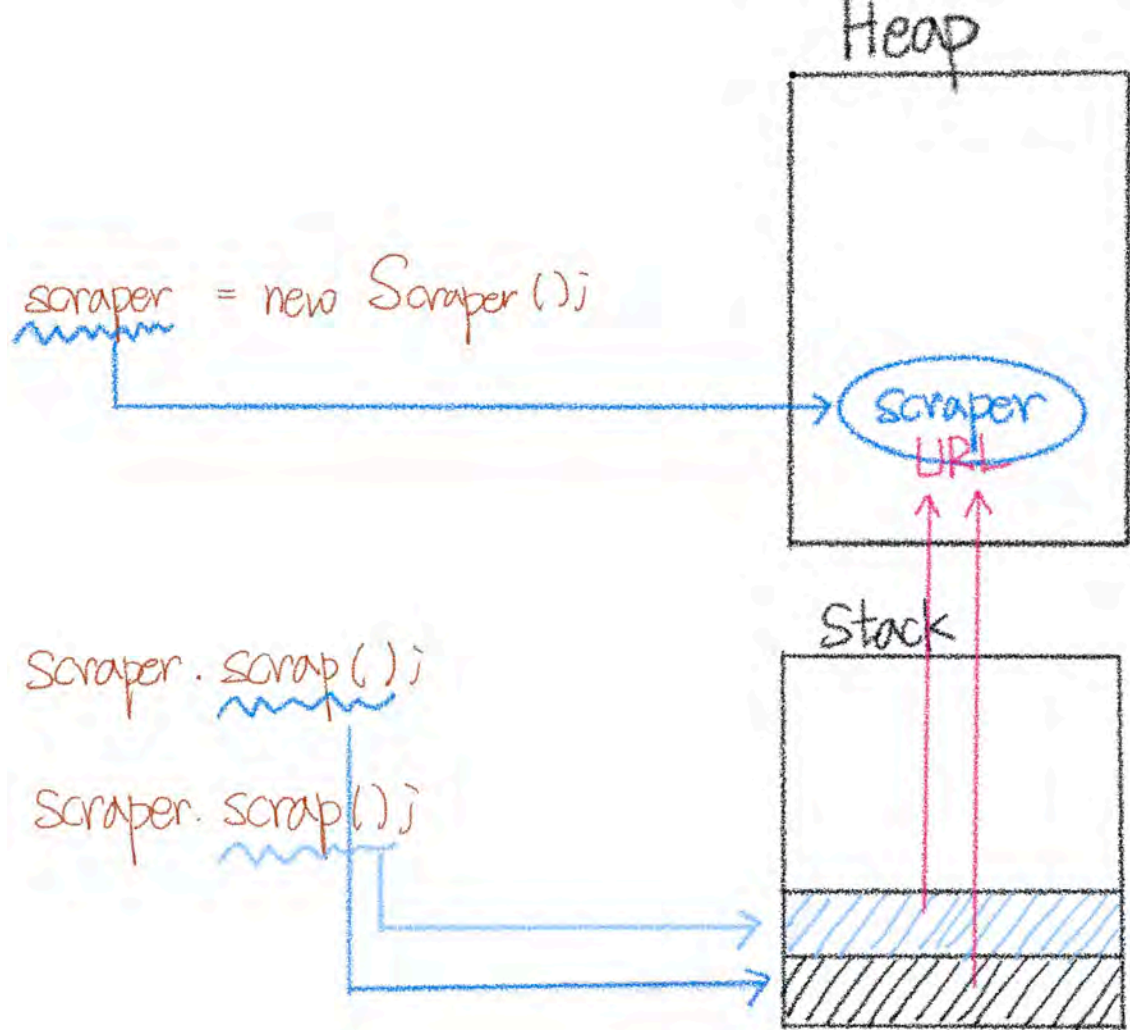
Scraper.scrap();

Scraper.scrap();



Chapter 03

02 배당금 저장 기능 구현



Chapter 03

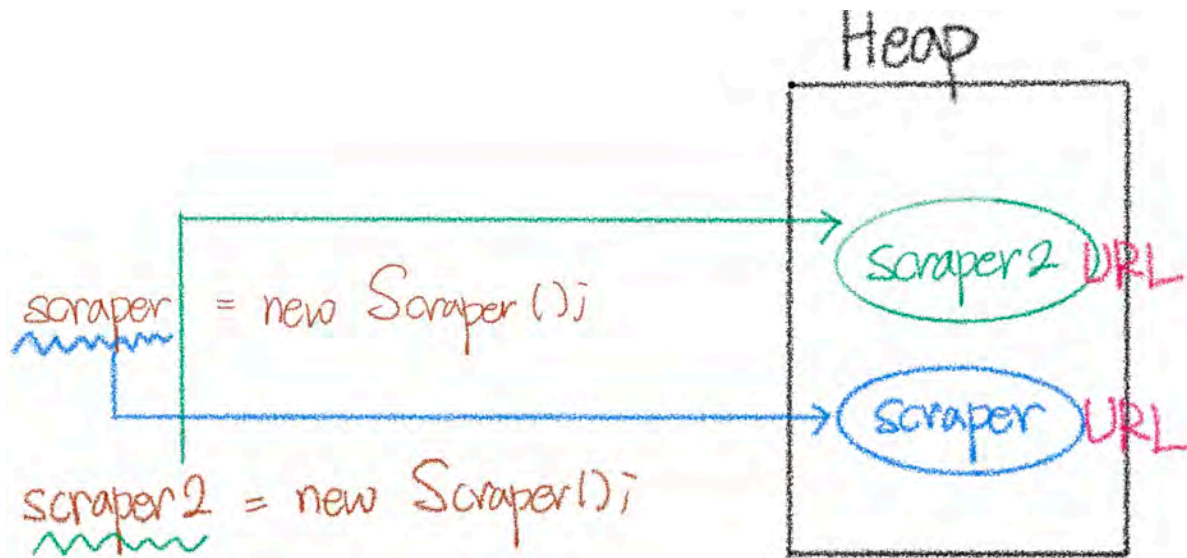
02 배당금 저장 기능 구현

Garbage Collection

- 메모리 할당 후 해제가 안되면 메모리 누수 발생
- 수동으로 개발자가 직접 해제할 경우(C언어), 올바르게 해제되지 않아 버그 발생 가능성 높음
- 가비지 컬렉터는 동적으로 할당된 메모리 영역 중 더 이상 쓰이지 않는 영역을 찾아내서 해제

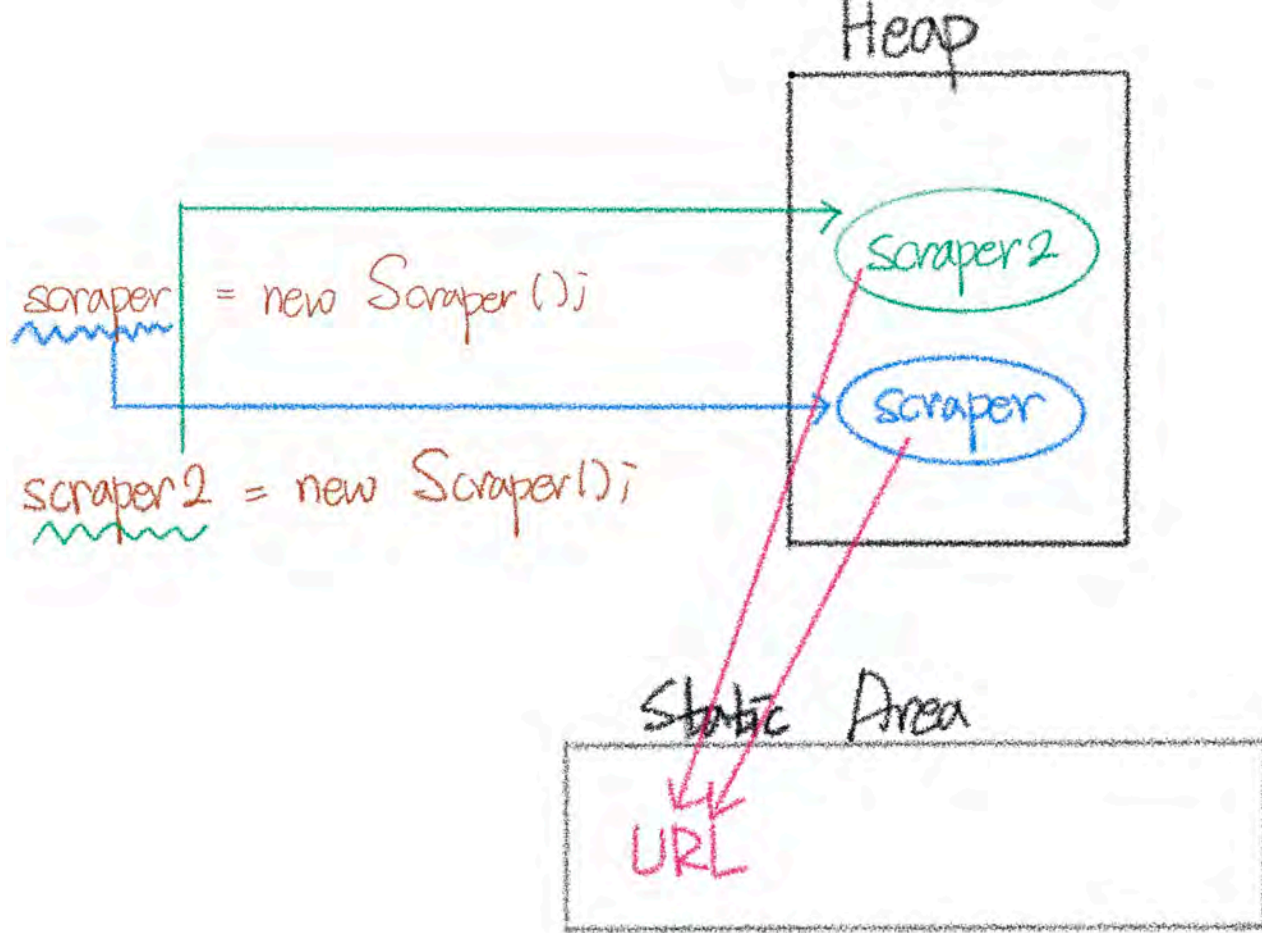
Chapter 03

02 배당금 저장 기능
구현



Chapter 03

02 배당금 저장 기능 구현



Chapter 03

02 배당금 저장 기능 구현



03 서비스 구현

03 배당금 저장 기능 구현

Chapter 03

03 배당금 저장 기능
구현

int to LocalDateTime

```
LocalDateTime date = LocalDateTime.of(2020, 5, 13, 6, 30);
```

Chapter 03

0 배당금 저장 기능 구현

Month

- Jan -> 1
- Feb -> 2
- Mar -> 3
- ...

Chapter 03

03 배당금 저장 기능 구현

03 서비스 구현

04 자동 완성

Chapter 03

04 자동 완성

Trie

- 트리형 자료구조
- 문자열 탐색을 효율적으로 할 수 있음
- 중복 저장 X

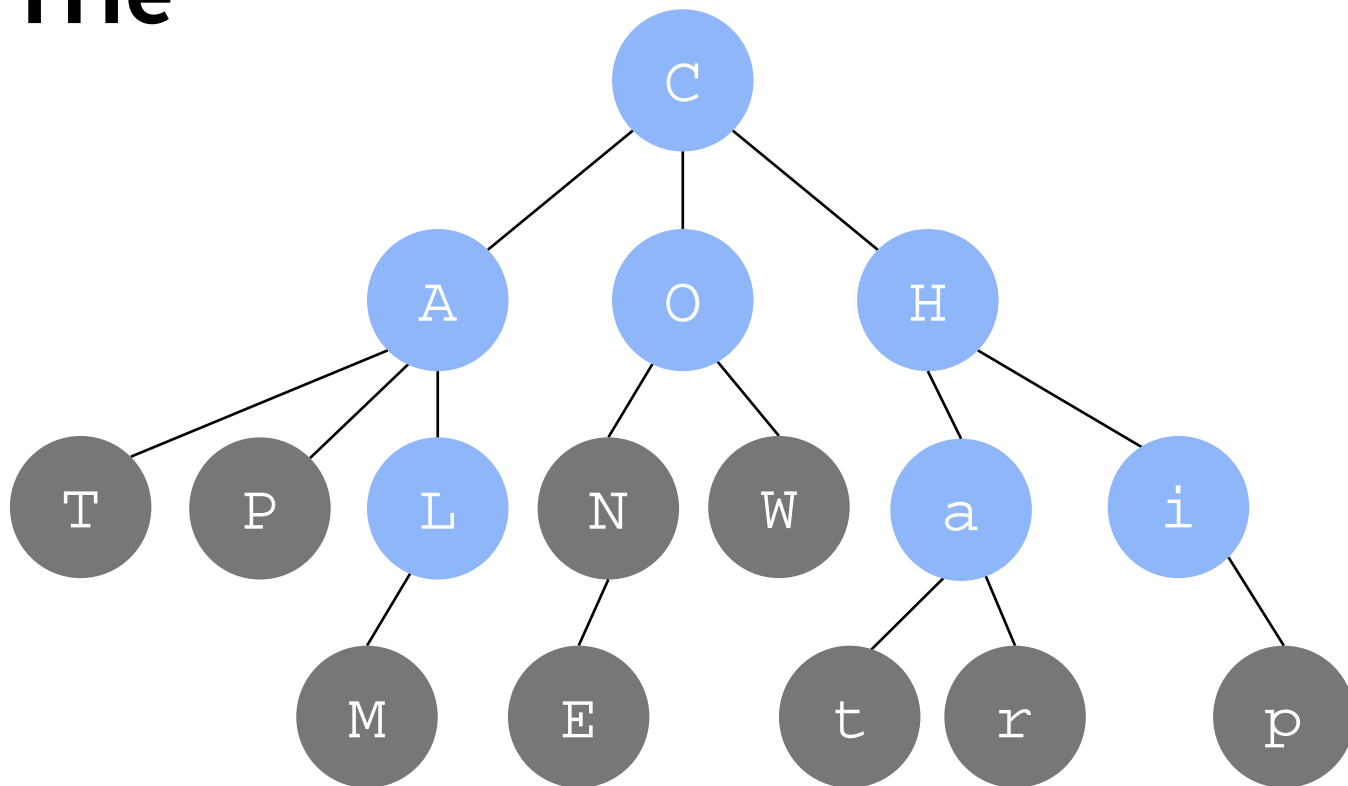
Chapter 03

04 자동완성

Trie

Chapter 03

04 자동완성



Trie 에 데이터 저장하기

- 삽입하고자 하는 문자열을 앞에서부터 한 글자씩 가져온다
- 트리의 루트부터 적합한 노드 위치를 찾아가면서 저장
- 마지막 글자까지 삽입이 되면 isEnd 플래그로 단어의 끝을 표시

Chapter 03

04 자동완성



Trie 에서 데이터 검색하기

- 인풋으로 받은 문자열을 한글자씩 파싱
- 파싱된 문자를 앞에서부터 비교
- 해당 문자 노드가 존재하지 않거나, 리프노드에 도달할 때 까지 탐색

Chapter 03

04 자동완성

시간 복잡도

- $O(L)$

L: 문자열의 길이

Chapter 03

04 자동완성

Trie 단점

- 메모리

Chapter 03

04 자동완성

03 서비스 구현

04 자동 완성

Chapter 03

04 자동 완성

Like 연산자

- 부분적으로 일치하는 조건으로 데이터를 찾기 위해 사용
- 특정 문자열이 포함된 경우
- 특정 문자열로 시작하거나
- 특정 문자열로 종료될 때

Chapter 03

04 자동완성

Like 연산자

- `SELECT *`
 `FROM company`
 `WHERE name LIKE "LA%";`

Like 연산자

- % : 모든 문자
- _ : 한 글자

Chapter 03

04 자동완성

Like 연산자

- SELECT *
FROM company
WHERE name LIKE "%A%";
- SELECT *
FROM company
WHERE name LIKE "%A";

Like 연산자

- NOT LIKE
- LIKE IN

Chapter 03

04 자동완성

03 서비스 구현

05 스케줄러

Chapter 03

05 스케줄러

Scheduler

- 일정 주기마다 특정 작업을 수행

Chapter 03

05 스케줄러

Scheduler

- fixedDelay
- fixedRate
- cron

Chapter 03

05 스케줄러

Cron 표현식

- 스케줄러 정규 표현식
- 초 / 분 / 시 / 일 / 월 / 요일 / 년도(생략가능)
- 0 5 * * * *
- 0 0/10 * * * *

Cron

Chapter 03

05 스케줄러

초	0 - 59
분	0 - 59
시	0 - 23
일	1 - 31
월	1 - 12 JAN - DEC
요일	0 - 6 SUN - SAT
연도	1970 - 2099

Cron

Chapter 03

05 스케줄러

*	모든 수
?	조건 없음 (날짜와 요일에만 사용)
-	범위(기간) 지정
,	특정 여러 시간 지정
/	시작 시간과 반복 간격
L	지정할 수 있는 범위의 마지막 값 (날짜와 요일에만 사용)

Cron

- 0 0 14 * * * : 매일 오후 2시
- 0 0 0 1 * * : 매달 1일 0시
- 0 5 1 ? 7 MON-WEB : 매년 7월 월-수 1시 5분

Spring Batch

Chapter 03

05 스케줄러

- **Job**: 배치 처리 과정의 단위
- **step**: 실제 batch 작업을 수행하고자 하는 내용
- 하나 이상의 step 이 Job 을 구성

03 서비스 구현

05 스케줄러

Chapter 03

05 스케줄러

Unique Key

- 중복 데이터 저장을 방지하는 제약조건
- 단일 컬럼 뿐 아니라 복합 컬럼을 지정할 수도 있음

Chapter 03

05 스케줄러

INSERT IGNORE

- `INSERT IGNORE INTO [TABLE] (COLUMN1, COLUMN2)
VALUES (VALUE1, VALUE2)`

ON DUPLICATE KEY UPDATE

- `INSERT INTO [TABLE] (COLUMN1, COLUMN2)`
 `VALUES (VALUE1, VALUE2)`
 `ON DUPLICATE KEY UPDATE (...)`

DB Index

- Cardinality
- 성별
- 주민등록번호

Chapter 03

05 스케줄러

03 서비스 구현

05 스케줄러

Chapter 03

05 스케줄러

Thread.sleep(n)

- 실행중인 스레드를 잠시 멈추게 할 때 사용
- `thread.sleep(1000)`: 1초간 스레드 정지

InterruptedException

Chapter 03

05 스케줄러

- 인터럽트를 받는 스레드가 blocking 될 수 있는 메소드를 실행할 때 발생

sleep() vs wait()

- wait() 는 스레드를 대기 상태에 빠뜨림
- notify() 나 notifyAll() 메소드를 호출할 때까지 자동으로 깨지 않음

Thread 5 status

- New
- Ready
- Running
- Blocked/Waiting
- Exit

Chapter 03

05 스케줄러

03 서비스 구현

05 스케줄러

Chapter 03

05 스케줄러

Thread Pool

- 여러 개의 스레드를 유지/관리

Chapter 03

05 스케줄러

Thread Pool 의 적정 사이즈는?

- CPU 처리가 많은 경우
- I/O 작업이 많은 경우

Chapter 03

05 스케줄러

03 서비스 구현

06 캐시

Chapter 03

06 캐시

Redis 명령어

- `$> set myKey myValue`
- `$> OK`

- `$> get myKey`
- `$> myValue`

- `$> del myKey`
- `$> myValue`

- `$> get myKey`
- `$> (nil)`

Redis 명령어

- `$> keys *`
- http://redisgate.kr/redis/introduction/redis_intro.php

Chapter 03

06 캐시

03 서비스 구현

06 캐시

Chapter 03

06 캐시

Serialization

- 데이터 or 오브젝트와 같은 값들을 바이트 형태로 변환



03 서비스 구현

01 회원관리

Chapter 03

01 회원관리

JWT(Json Web Token)

- 사용자 인증 및 식별에 사용되는 토큰
- 토큰은 사용자 정보를 포함

Chapter 03

01 회원관리

JWT(Json Web Token) 구조

- Header
- Payload
- Signature

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikp  
vaG4gRG91IiwiaWF0IjoxNTE2MjM5MDIyfQ.  
Sf1KxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQs  
sw5c
```

암호화 알고리즘

- 암호화 vs 복호화
- 대칭키 vs 비대칭키

Chapter 03

01 회원관리

쿠키와 세션

- 쿠키와 세션의 차이
- 쿠키와 세션을 사용하는 이유

Chapter 03

01 회원관리

05 완성도 높이기

02 logging

Chapter 05
02 logging

Log Level

- DEBUG
- INFO
- WARN
- ERROR

Log

- Console
- File
- 중앙화

Chapter 05
02 logging

00 프로젝트 이후

00 기술 면접 준비

Chapter 00

00 기술 면접 준비



Java

- 객체지향
- GC
 - Garbage Collector 란?
 - STW(stop-the-world)

Chapter 00

00 기술면접준비

Spring

- 스프링 프레임워크의 장점과 단점
- DI
- AOP

Chapter 00

00 기술면접준비

Thread

- Thread vs Process
- 스레드 메모리 구조
 - 스레드 간 공유되는 공간과 공유되지 않는 공간
 - 메소드 내에서 선언된 변수가 위치하는 공간
- Context Switching

Chapter 00

00 기술면접준비

Data Structure

- Hash
 - hash function
 - hash collision
- 시간복잡도
- ArrayList vs LinkedList

Chapter 00

00 기술면접준비

Database

- Index

Chapter 00

00 기술면접준비

HTTP

- Rest API
- Status Code
- POST vs GET
- HTTP Protocol
- Cookie vs Session

Chapter 00

00 기술면접준비

Redis

- 다른 스토리지와 비교했을 때 어떤 특징이 있는지

Chapter 00

00 기술면접준비

Encryption

- 대칭키 vs 비대칭키

Chapter 00

00 기술면접준비

그 외에 ..

- 프로젝트를 하면서 어떤 기술적인 이슈가 있었는지
-> 어떻게 해결했는지
- 어떤 부분을 개선시킬 수 있는지
- 프로젝트 중 팀원과의 갈등 상황은 어떻게 해결하는지

Chapter 00

00 기술면접준비