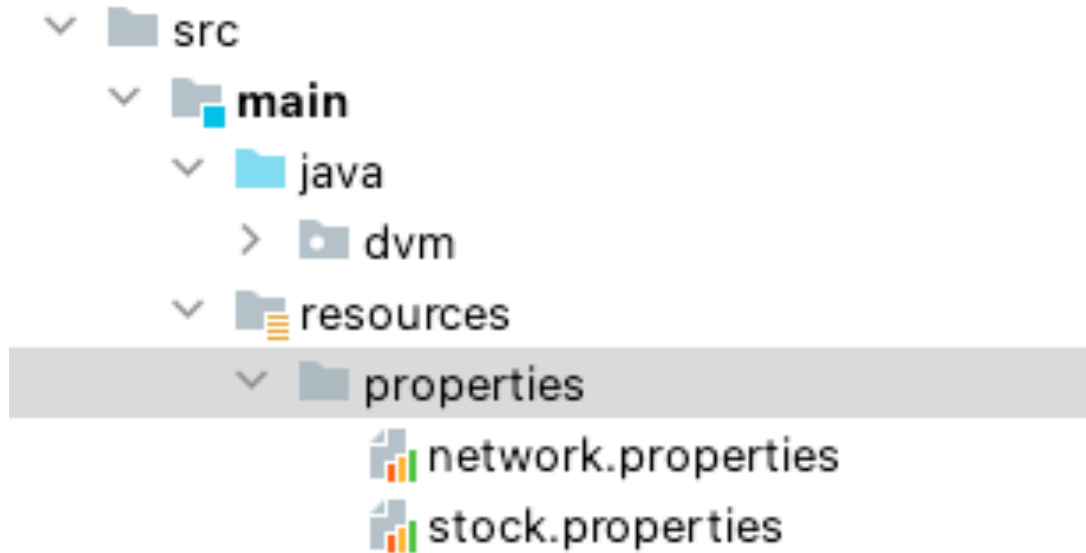


DVM 프로젝트 설정 방법

Team6

개요



- IP주소, 자판기 ID 설정, 초기 재고 설정등을 간편하게 하기 위해 properties 파일 사용
- 네트워크 설정을 담당하는 network.properties
- 재고 설정을 담당하는 stork.properties

network.properties 예시

```
1 current.id=Team6
2 current.x=130
3 current.y=30
4 other.id=\
5     Team1,\
6     Team2,\
7     Team3
8 other.ip=\
9     192.168.1.1,\
10    192.168.1.64,\
11    192.168.64.60
```

- 우리 자판기 ID
- 우리 자판기 x좌표
- 우리 자판기 y좌표
- 다른 자판기 ID
{Team_ID[,]}{Team_ID[,]}{Team_ID} (심표 좌우 공백 없음에 주의!!) 형식
또는 {Team_ID[,₩]} 의 형식으로 줄바꿈 가능
- 다른 자판기 IP주소
- 위의 other.id 항목의 순서와 1대1 대응

network.properties 예시

```
public static NetworkService networkService() {
    if (networkService == null) {
        try (BufferedReader reader = new BufferedReader
            (new InputStreamReader(new
FileInputStream("src/main/resources/properties/network.properties"), StandardCharsets.UTF_8))) {
            // resources에 properties/?.properties 파일들 읽어서 세팅 -> 매번 빌드 안하기 위함
            Properties p = new Properties();
            p.load(reader);
            MessageFactory.setCurrentId(p.getProperty("current.id"));
            MessageFactory.setCurrentX(Integer.parseInt(p.getProperty("current.x")));
            MessageFactory.setCurrentY(Integer.parseInt(p.getProperty("current.y")));
            Sender.initDvmsNetworkInfo(p.getProperty("other.id").replace(" ", "").split(","),
                p.getProperty("other.ip").replace(" ", "").split(","));
        } catch (Exception e) {
            e.printStackTrace();
            String currentId = "Team6";
            int currentX = 30;
            int currentY = 50;
            String[] ids = new String[]{"Team1"};
            String[] ips = new String[]{"127.0.0.1"};
            MessageFactory.setCurrentId(currentId);
            MessageFactory.setCurrentX(currentX);
            MessageFactory.setCurrentY(currentY);
            Sender.initDvmsNetworkInfo(ids, ips);
        }
        System.out.println("-----현재 네트워크 정보-----");
        MessageFactory.printCurrentInfo();
        Sender.printDvmsNetworkInfo();
        System.out.println("-----");
        networkService = new NetworkService(itemService(), prepaymentService());
    }
    return networkService;
}
```

- network.properties 파일은 AppConfig 클래스의 networkService() 메소드에서 사용
- 형식 오류등의 이유로 읽어들이지 못하고 예외가 발생한 경우 기본 설정이 적용됨

stock.properties 예시

```
1 01=100
2 02=0
3 05=0
4 06=0
5 18=0
6 19=500
7 20=500|
```

- 우리 자판기에서 판매중인 7가지 음료의 종류와 재고 표시
- {음료코드=개수} 형식으로 사용
- PFR에 따라 음료코드는 01~20, 음료 개수는 0~999로 제한

stock.properties 예시

```
private ItemRepository() { // singleton 위해 생성자 접근 막음
    stock = new ConcurrentHashMap<>();
    try (BufferedReader reader = new BufferedReader
        (new InputStreamReader(new
FileInputStream("src/main/resources/properties/stock.properties"), StandardCharsets.UTF_8))) {
        Properties p = new Properties();
        p.load(reader);
        if (p.size() != 7) {
            logger.warning("자판기 음료 개수는 7개여야 합니다.");
            throw new Exception();
        }
        for (Object o : p.keySet()) {
            String key = (String) o;
            stock.put(key, Integer.parseInt(p.getProperty(key)));
        }
    } catch (Exception e) {
        logger.warning("stock.properties 이상. 기본 세팅으로 세팅합니다.");
        stock.put("01", 2);
        stock.put("02", 2);
        stock.put("03", 2);
        stock.put("04", 2);
        stock.put("05", 2);
        stock.put("06", 2);
        stock.put("10", 10);
    }
    printCurrentStock();
}
```

- stock.properties 파일은 ItemRepository 클래스의 생성자에서 사용
- 형식 오류등의 이유로 읽어들이지 못하고 예외가 발생한 경우 기본 설정이 적용됨