DVM_Network

V 1.2.0

Server 객체 사용법

```java
package DVM_Server;
import Model.Message;
import javafx.collections.ListChangeListener;


public interface messageListChangeListner extends ListChangeListener<Message>
{
    void onChanged(Change<? extends Message> change);
}
```

Interface : messageListChangeListener

Server 객체 내의 ObservableList<Message>에 변화가 생길 때 Trigger되는 eventListener
ListChangeListener를 상속 받아 사용. onChanged method 만 추상메서드로 존재

Server 객체 사용법

예제

```
import DVM_Server.messageListChangeListner;
import Model.Message;



public class Main  implements messageListChangeListner {


    @Override
    public void onChanged(Change<? extends Message> change) {
        //다음 이벤트가 존재하는 동안
        while(change.next()){
            //새로운 메시지가 추가되면
            if(change.wasAdded()){
              //List의 크기
                int size = change.getList().size();
                System.out.println(change.getList().get(size-1).getMsgType());
            }
       }
    }
}
```

onChange에 각 팀별로 client 메시지에 대한 응답 혹은 처리 로직을 작성 , 그림의 예제는 새로 메시지가 들어올 경우 가장 마지막 메시지의 타입을 반환하여 출력하는 예제.

while (change.next)
다음 이벤트가 존재할 경우 무한루프

If(change.wasAdded)
List에 객체가 추가될 경우

Change.getList()
변화가 발생한 리스트를 반환

Server 객체 사용법

예제

```
import DVM_Server.DVMServer;

public class Test {
    public static void main(String[] args) throws Exception {
        DVMServer server = new DVMServer();
        server.setMessageListner(new Main());
        server.run();
    }
}
```

예시와 같이 server 가동전에 원하는 동작을 작성하여 Listener 추가 후 서버 동작

질문 및 오류 관련 연락처

union93@konkuk.ac.kr