<Service Server>

디렉토리 구성

|  |  |
| --- | --- |
| Directory Name | Description |
| src | \*.java Files |
| lib | External Library |
| bin | \*.class Files |
| properties | Properties Text Files |
| Image | Image Files(서버 구동 시, 클라이언트에서 받아오는 이미지 파일들) |

패키지 구성

|  |  |
| --- | --- |
| Packege Name | Description |
| kr.android.hellodrinking.server | 서버의 생성 및 기능에 관련한 클래스를 제공합니다. |
| kr.android.hellodrinking.server.db | 데이터 베이스의 연결 및 접근에 관련한 클래스를 제공합니다. |
| kr.android.hellodrinking.server.db.connection | 데이터 베이스의 연결에 관련한 클래스를 제공합니다. |
| kr.android.hellodrinking.transmission | 네트워크 어플리케이션의 구현을 위한 클래스를 제공합니다. |
| kr.android.hellodrinking.transmission.dto | Bean 의 생성에 필요한 클래스를 제공합니다. |
| kr.android.hellodrinking.transmission.exception | 통신 중 일어날 수 있는 예외에 대해 정의한 클래스를 제공합니다. |

클래스 관계 개요도

Server

ClientThread

DAO

Resonce

Client

DB

Request

Data Model

|  |  |  |
| --- | --- | --- |
| Class | Description | 해당 Class |
| Server | Serversocket을 맴버로 가지고 있으며, 클라이언트 접속이 있을 시, ClientThread를 생성 | HelloDrinkingServer |
| ClientThread | 접속한 Client의 요청에 따른 처리를 담당하는 클래스 | ClientThread, RequestProcess |
| DAO | 데이터베이스에 접속하여 해당 처리를 수행하는 클래스 | DBAccess, Connectable, ConnectFromSettingFile |
| Data Model | Activity 또는 View 에 표현되는 정보를 담고 있는 클래스  \* 클라이언트와 동일한 클래스가 사용된다. | Bean, PostBean, UserBean |
| Request | 메서드를 호출하기 위해서 필요한 정보를 포함한 클래스  \* 클라이언트와 동일한 클래스가 사용된다. | Request, Requestable, RequestBeanPackege |
| Responce | Request를 받아 처리된 정보를 포함한 클래스  \* 클라이언트와 동일한 클래스가 사용된다. | ResponceBeanPackege |

Server Classes

|  |  |
| --- | --- |
| Class Name | Description |
| HelloDrinkingServer | Serversocket을 맴버로 가지고 있으며, 클라이언트 접속이 있을 시, ClientThread를 생성하는 클래스 |

**class** HelloDrinkingServer

|  |  |
| --- | --- |
| Constructor | Description |
| HelloDrinkingServer() |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + start() | void | 클라이언트 접속이 있을 시, ClientThread를 생성한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| + (s) (f) DEFAULT\_IMAGEFILE\_DIRECTORY | String | 이미지 파일이 저장되는 디렉토리 패스 : Image |
| + (s) (f) PROFERTIES\_FILE\_PATH | String | 설정 파일이 저장되는 디렉토리 패스 : properties/properties.txt |
| + (s) (f) DEFAULT\_DB\_PORT | int | 기본 DB port : 1521 |
| + (s) (f) DEFAULT\_DB\_IP | String | 기본 DB ip : 127.0.0.1 |
| + (s) (f) DEFAULT\_DB\_USER | String | 기본 DB user : Project |
| + (s) (f) DEFAULT\_DB\_PASSWOR | String | 기본 DB password :kitri |
| + (s) (f) DEFAULT\_JDBC\_ID | String | 기본 JDBC id : xe |
| + (s) (f) DEFAULT\_JDBC\_CLASS | String | 기본 JDBC class : oracle.jdbc.driver.OracleDriver |
| + (s) (f) DEFAULT\_JDBC\_DRIVER | String | 기본 JDBC driver : oracle:thin |

ClientThread Classes

|  |  |
| --- | --- |
| Class Name | Description |
| ClientThread | Socket 연결 및 IO 관리에 사용하는 클래스 |
| RequestProcess | Client의 Request 처리에 사용되는 클래스 |

**class** ClientThread **extends** Thread

|  |  |
| --- | --- |
| Constructor | Description |
| ClientThread(Socket) |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + run() | void | 요청에 따라 RequestProcess를 생성 및 실행한다. |
| + close() | void | 모든 자원을 반환한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| mSocket | Socket | 클라이언트 소켓를 생성 및 관리에 사용되는 객체 |
| mReader | InputStream | 소켓의 스트림 관리에 사용되는 객체 |
| mWriter | OutputStream | 소켓의 스트림 관리에 사용되는 객체 |
| mObjectReader | ObjectInputStream | 소켓의 스트림 관리에 사용되는 객체 |
| mObjectWriter | ObjectOutputStream | 소켓의 스트림 관리에 사용되는 객체 |
| mConnection | Connection | DB 연결에 사용되는 객체 |

**class** RequestProcess

|  |  |
| --- | --- |
| Constructor | Description |
| RequestProcess(RequestBeanPackege, Connection, ObjectOutputStream) |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + processing() | void | Request의 종류에 따른 분기를 실행한다. |
| - register() | boolean | 회원 가입을 위한 메서드 |
| - login() | boolean | 로그인을 위한 메서드 |
| - userModify() | boolean | 회원 정보 수정을 위한 메서드 |
| - getUser() | boolean | 회원의 정보를 받아오기 위한 메서드 |
| - post() | boolean | 게시물을 등록하기 위한 메서드 |
| - getPosts() | boolean | 게시물을 가져오기 위한 메서드 |
| - writerImageFileAndGetImageFilePath( String, byte[]) | String | 파일을 기록하고, 해당 파일의 경로를 반환한다. |
| sendResponce(ResponceBeanPackege) | boolean | ResponcePackege를 클라이언트에 전송한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| mRequest | Request | 요청 처리를 위해 사용되는 객체 |
| mObjectWriter | ObjectOutputStream | 소켓의 스트림 쓰기에 사용되는 객체 |
| mDBAccess | DBAccess | 데이터베이스에 접근하여 해당 요청을 처리하기 위해 사용되는 객체 |

DAO Classes

DBAccess

ConnectionFromSettingFile

Connectable

|  |  |
| --- | --- |
| Class Name | Description |
| DBAccess | 데이터베이스에 접근하는 DAO 클래스 |
| ConnectionFromSettingFile | 설정 파일로부터 정보를 읽어 Connection 객체를 생성하는 클래스 |
| Connectable | Connection 객체를 생성하기 위해 필요한 메서드를 정의한 인터페이스 |

**interface** Connectable

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + getConnection() | Connection | DB연결에 사용되는 Connection 객체를 반환한다. |

**class** ConnectionFromSettingFile

|  |  |
| --- | --- |
| Constructor | Description |
| - ConnectionFromSettingFile() |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + (s) getInstance() | ConnectFrom SettingFile | ConnectFromSettingFile 의 생성을 대신하여 사용된다. |
| + getConnection() | Connection | DB연결에 사용되는 Connection 객체를 반환한다. |
| - createDefaultFile(File) | void | HelloDrinkingServer의 변수를 통해 설정 파일을 만든다. |
| - setDefaultSetting() | void | HelloDrinkingServer의 변수를 통해 Connection 객체 생성에 필요한 변수를 초기화한다. |
| - readSettingFile(File) | void | 해당 파일 객체로부터 설정에 Connection 객체 생성에 필요한 변수를 초기화한다. |
| - getValue(StreamTokenizer) | String | StreamTokenizer 객체로부터 String 값을 구해 반환한다. |
| - readDBSetting(BufferedReader) | String | DB 설정에 관련한 변수를 초기화한다.  다음 행을 반환한다. |
| - readJDBCSetting(BufferedReader) | String | JDBC 설정에 관련한 변수를 초기화한다.  다음 행을 반환한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - (s) instance | ConnectFromSettingFile |  |
| - db\_ip | String | Connection 객체의 생성에 사용되는 DB ip |
| - db\_port | int | Connection 객체의 생성에 사용되는 DB port |
| - db\_user | String | Connection 객체의 생성에 사용되는 DB user |
| - db\_password | String | Connection 객체의 생성에 사용되는 DB password |
| - jdbc\_class | String | Connection 객체의 생성에 사용되는 JDBC class |
| - jdbc\_driver | String | Connection 객체의 생성에 사용되는 JDBC driver |
| - jdbc\_id | String | Connection 객체의 생성에 사용되는 JDBC id |

**class** DBAccess

|  |  |
| --- | --- |
| Constructor | Description |
| DBAccess (Connection) |  |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + addLoginHistory(String, boolean) | Exception | 로그인 기록을 추가한다. 시간, 아이디, 성공여부를 기록한다. |
| + getPosts(double, double, int) | Exceptioni | 지정된 경도, 위도에서 일정 거리 내의 게시물을 가져온다.  성공적으로 가져온 경우 kr.android. hellodrinking.transmission.exception.Message를 반환한다. |
| + getUser(String) | Exceptioni | 아이디를 통해 검색된 User의 정보를 가져온다.  성공적으로 가져온 경우 kr.android. hellodrinking.transmission.exception.Message를 반환한다. |
| + getUserPassword(String) | Exceptioni | 아이디를 통해 검색된 User의 암호를 가져온다.  성공적으로 가져온 경우 kr.android. hellodrinking.transmission.exception.Message를 반환한다. |
| + register(String, String, String, String, String, String, String, String) | Exceptioni | User를 등록한다. |
| + post(String, String, String, double, double) | Exceptioni | 게시물을 등록한다. |
| + setImageAtPostInfo(int, String) | Exceptioni | 지정 postnum의 게시물의 Image를 해당 파일 패스로 교체한다. |
| + setImageAtUserInfo(String, String) | Exceptioni | 지정 id의 사용자의 Image를 해당 파일 패스로 교체한다. |
| + userModify(String, String, String, String, String, String, String, String) | Exceptioni | 지정 id의 사용자의 정보를 갱신한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| mConnection | Connection | DB 연결에 사용되는 객체 |

Request Classes

RequestBeanPackege

Request

Requestable

|  |  |
| --- | --- |
| Class Name | Descripstion |
| Requestable | 가능한 요청의 종류를 정의하는 인터페이스 |
| Request | 요청을 구현하는 클래스 |
| RequestBeanPackege | 실제 요청되는 내부 메시지(Bean) 클래스 |

**interface** Requestable

* 모든 요청은 OptionalDataException, IOException 을 발생시킨다.

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + getPosts(double, double, int) | ResoponceBeanPackege | 지정된 경도, 위도로부터 지정 거리 내의 게시물을 가져온다. |
| + getUser(String) | ResoponceBeanPackege | 지정된 id를 가진 사용자의 정보를 가져온다. |
| + getUser(UserBean) | ResoponceBeanPackege | 지정된 id를 가진 사용자의 정보를 가져온다. |
| + login(String, String) | ResoponceBeanPackege | 지정된 id와 password가 옳바른지 확인한다. |
| + login(UserBean) | ResoponceBeanPackege | 지정된 id와 password가 옳바른지 확인한다. |
| + modifyUser(String, String, String, String, String, String, String, String) | ResoponceBeanPackege | 지정 id의 사용자의 정보를 갱신한다. |
| + modifyUser(UserBean) | ResoponceBeanPackege | 지정 id의 사용자의 정보를 갱신한다. |
| + post(PostBean) | ResoponceBeanPackege | 게시물을 등록한다. |
| + post(String, String, String, double, double) | ResoponceBeanPackege | 게시물을 등록한다. |
| + register(String, String, String, String, String, String, String, String) | ResoponceBeanPackege | 사용자를 등록한다. |
| + register(UserBean) | ResoponceBeanPackege | 사용자를 등록한다. |

**class** Request

|  |  |
| --- | --- |
| Constructor | Description |
| Request() | 기본 ip, port의 서버에 접근한다. |
| Request(String, int) | 지정 ip, port의 서버에 접근한다. |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + getPosts(double, double, int) | ResoponceBeanPackege | 지정된 경도, 위도로부터 지정 거리 내의 게시물을 가져온다. |
| + getUser(String) | ResoponceBeanPackege | 지정된 id를 가진 사용자의 정보를 가져온다. |
| + getUser(UserBean) | ResoponceBeanPackege | 지정된 id를 가진 사용자의 정보를 가져온다. |
| + login(String, String) | ResoponceBeanPackege | 지정된 id와 password가 옳바른지 확인한다. |
| + login(UserBean) | ResoponceBeanPackege | 지정된 id와 password가 옳바른지 확인한다. |
| + modifyUser(String, String, String, String, String, String, String, String) | ResoponceBeanPackege | 지정 id의 사용자의 정보를 갱신한다. |
| + modifyUser(UserBean) | ResoponceBeanPackege | 지정 id의 사용자의 정보를 갱신한다. |
| + post(PostBean) | ResoponceBeanPackege | 게시물을 등록한다. |
| + post(String, String, String, double, double) | ResoponceBeanPackege | 게시물을 등록한다. |
| + register(String, String, String, String, String, String, String, String) | ResoponceBeanPackege | 사용자를 등록한다. |
| + register(UserBean) | ResoponceBeanPackege | 사용자를 등록한다. |
| - createSocket() | void | Socket 객체를 생성한다. |
| - sendRequestAndGetResponce( RequestBeanPackege) | ResoponceBeanPackege | 지정된 RequestBeanPackege를 전송 후, ResponceBeanPackege를 전송 받아 반환한다. |
| + close() | void | 모든 자원을 반환한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| mSocket | Socket | 클라이언트 소켓를 생성 및 관리에 사용되는 객체 |
| mReader | InputStream | 소켓의 스트림 관리에 사용되는 객체 |
| mWriter | OutputStream | 소켓의 스트림 관리에 사용되는 객체 |
| mObjectReader | ObjectInputStream | 소켓의 스트림 관리에 사용되는 객체 |
| mObjectWriter | ObjectOutputStream | 소켓의 스트림 관리에 사용되는 객체 |

Data Model Classes

Bean

PostBean

UserBean

|  |  |
| --- | --- |
| Class Name | Descripstion |
| Bean | UserBean과 PostBean에서 공통적으로 사용되는 필드 및 메서드를 추상화시킨 추상 클래스 |
| UserBean | 사용자에 관련한 정보를 담고 있는 클래스 |
| PostBean | 게시물에 관련한 정보를 담고 있는 클래스 |

**abstract class** Bean **implements** Serializable

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + getBuffer() | byte[] | 파일 배열을 반환한다. |
| + getId() | String | 아이디를 반환한다. |
| + getImageFilePath() | String | 이미지 파일 경로를 반환한다. |
| + setBuffer(byte[]) | void | 파일 배열을 설정한다. |
| + setId(String) | void | 아이디를 설정한다. |
| + setImageFilePath(String) | void | 이미지 파일 경로를 설정한다. 그리고 convertToBytes()를 실행한다. |
| = convertToBytes() | void | 파일 경로를 통해 파일을 읽어 그 배열을 buffer에 저장시킨다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| = buffer | byte[] | 파일의 내용을 담고 있는 배열 |
| = id | String | 아이디를 가르키는 객체 |
| = imageFilePath | String | 파일 경로를 가르키는 객체 |

**class** UserBean **extends** Bean **implements** Serializable

|  |  |
| --- | --- |
| Constructor | Descripstion |
| + UserBean() |  |
| + UserBean(String) | 아이디를 가지고 객체를 생성한다. |
| + UserBean(String, String) | 아이디와 암호를 가지고 객체를 생성한다. |
| + UserBean(String, String, String, String, String, String, String, String) | 모든 정보를 가지고 객체를 생성한다. |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + getAge() | String | 나이를 반환한다. |
| + getJob() | String | 직업을 반환한다. |
| + getName() | String | 이름을 반환한다. |
| + getPassword() | String | 암호를 반환한다. |
| + getPhone() | String | 전화번호를 반환한다. |
| + getSex() | String | 성별을 반환한다. |
| + setAge(String) | void | 나이를 설정한다. |
| + setJob(String) | void | 직업를 설정한다. |
| + setName(String) | void | 이름를 설정한다. |
| + setPassword(String) | void | 암호를 설정한다. |
| + setPhone(String) | void | 전환번호를 설정한다. |
| + setSex(String) | void | 성별를 설정한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - age | String | 나이를 가르키는 객체 |
| - job | String | 직업을 가르키는 객체 |
| - name | String | 이름을 가르키는 객체 |
| - password | String | 암호를 가르키는 객체 |
| - phone | String | 전화번호를 가르키는 객체 |
| - sex | String | 성별을 가르키는 객체 |

**class** PostBean **extends** Bean **implements** Serializable

|  |  |
| --- | --- |
| Constructor | Descripstion |
| + PostBean(String, double, double) | 아이디, 경도, 위도 정보를 가지고 객체를 생성한다. |
| + PostBean(String, String, String, double, double) | 모든 정보를 가지고 객체를 생성한다. |

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| + getComment() | String | 코멘트를 반환한다. |
| + getLatitude() | double | 위도을 반환한다. |
| + getLongitude() | double | 경도을 반환한다. |
| + getPostnum() | int | 게시물번호를 반환한다. |
| + setComment(String) | void | 코멘트를 설정한다. |
| + setLatitude(double) | void | 위도를 설정한다. |
| + setLongitude(double) | void | 경도를 설정한다. |
| + setPostnum(int) | void | 게시물번호를 설정한다. |

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| - comment | String | 나이를 가르키는 객체 |
| - latitude | double | 위도을 가르키는 객체 |
| - longitude | double | 경도을 가르키는 객체 |
| - postnum | int | 게시물번호를 가르키는 객체 |