

Network Protocol

Index

- Network Data
- Network Messages
- Network Messages Protocols
 - Clients to server
 - * Patient to Server
 - Patient login
 - Push measurement
 - * Doctor to Server
 - Doctor login
 - Get patient measures
 - Push measurement comment
 - * Administrator to server
 - Admin login
 - Get Doctors
 - Register Patient
 - Register Doctor
 - Register Administrator
 - Server shutdown
 - Server cancel shutdown
 - Server to clients
 - * Server to Patient
 - * Server to Doctor
 - Push patient measures
 - * Server to Admin
 - Push doctors
 - * Server to any
 - Login accept
 - Login deny
 - ACK
 - Error
 - From either one
 - * Disconnect
 - * Server shutdown confirm

Network data

- Server Address: **TO BE DETERMINED**
 - Server Port: **22333**
-

Network Messages

The server listens to objects of the type **NetworkMessage**. These messages have a mandatory **Protocol** attribute which will determine what the message purpose is. The type “Protocol” is an enumeration of the different protocols the server recognizes. Depending on the protocol attribute, the **NetworkMessage** object will have to contain the other attributes. — ## **NetworkMessages Protocol Enumeration** Here we will see what each protocol implies and what should be expected in the **NetworkMessage** object with it.

From Patient to Server

Protocol PATIENT_LOGIN

- **Usage:** After a successful TCP connection, this should be the first message sent to the server by any patient client. Its intention is to check if the user is registered in the database as a patient and with the right password.
- **Requirements:** Requires a **Patient** object with at least DNI and password.
- **Possible answers:**
 - *LOGIN_ACCEPT*
 - *LOGIN_DENY*
 - *ERROR*

Protocol: PUSH_MEASUREMENT

- **Usage:** Once the client has logged in with a patient DNI and password, they can upload as many measurements as they want. It can't be done before login in with *PATIENTLOGIN_*
- **Requirements:** Requires a **ArrayList< Measurement >** of at least 1 measurement.
- **Possible answers:**
 - *ACK*
 - *ERROR*

From Server to Patient

Currently there are no specific messages from the server to the patient client.

From Doctor to Server

Protocol: DOCTOR_LOGIN

- **Usage:** After a successful TCP connection, this should be the first message sent to the server by any Doctor client. Its intention is to check if the user is registered in the database as a doctor and with the right password.
- **Requirements:** Requires a **Doctor** object with at least DNI and password.
- **Possible answers:**
 - *LOGIN_ACCEPT*
 - *LOGIN_DENY*
 - *ERROR*

Protocol: GET_PATIENT_MEASURES

- **Usage:** When the Doctor chooses a patient to look the measurements of, this protocol is called to retrieve the measurements of the patient.
- **Requirements:** Requires a **Patient** object.
- **Possible answers:**
 - *PUSH_PATIENT_MEASURES*
 - *ERROR*

Protocol: PUSH_MEASUREMENT_COMMENT

- **Usage:** to add a comment to the measurement of a patient.
- **Requirements:** Requires a **Measurement** object.
- **Possible answers:**
 - *ACK*
 - *ERROR*

From Server to Doctor

Protocol: PUSH_PATIENT_MEASURES

- **Usage:** Answers when the doctor wants the measurements of a particular patient.

- **Requirements:** Requires an **ArrayList< Measurement >** to return. This list could be empty.
- **Answers to:**
 - *GET_PATIENT_MEASURES*

From Administrator to Server

Protocol: ADMIN_LOGIN

- **Usage:** After a successful TCP connection, this should be the first message sent to the server by any admin client. Its intention is to check if the user is registered in the database as a admin and with the right password.
- **Requirements:** Requires an **Administrator** object with at least DNI and password.
- **Possible answers:**
 - *LOGIN_ACCEPT*
 - *LOGIN_DENY*
 - *ERROR*

Protocol: GET_DOCTORS

- **Usage:** retrieve all doctors' ids and names so a new patient can be assigned to one of them
- **Requirements:** no requirements;
- **Possible answers:**
 - *PUSH_DOCTORS*
 - *ERROR*

Protocol: REGISTER_PATIENT

- **Usage:** to add a new patient to the database by an admin
- **Requirements:** Requires a **Patient** object with the database requirements; when provided the server answers with *ACK*, otherwise with *ERROR*.
- **Possible answers:**
 - *ACK*
 - *ERROR*

Protocol: REGISTER_DOCTOR

- **Usage:** to add a new patient to the database by an admin
- **Requirements:** Requires a **Doctor** object with the database requirements; when provided the server answers with *ACK*, otherwise with *ERROR*.
- **Possible answers:**
 - *ACK*
 - *ERROR*

Protocol: REGISTER_ADMIN

- **Usage:** to add a new patient to the database by an admin
- **Requirements:** Requires a **Administrator** object with the database requirements; when provided the server answers with *ACK*, otherwise with *ERROR*.
- **Possible answers:**
 - *ACK*
 - *ERROR*

Protocol: SERVER_SHUTDOWN

- **Usage:** When the administrator wants to shutdown the server
- **Requirements:** An **Administrator** object with the id and the password of the administrator requesting the shutdown, for extra security.
- **Possible answers:**
 - *ACK*
 - *SERVER_SHUTDOWN_CONFIRM*: Used in case there are other clients connected.
 - *ERROR*

Protocol: SERVER_CANCEL_SHUTDOWN

- **Usage:** When there are other clients connected to the server and the administrator decides not to shutdown the server.
- **Requirements:** none.
- **Possible answers:**
 - *ACK*
 - *ERROR*

From Server to Administrator

Protocol: PUSH_DOCTORS

- **Usage:** To answer *GETDOCTORS_*
- **Returns:** A `LinkedList< Doctor >`. The list might be empty if no doctors have been registered.
- **Answers to:**
 - *GET_DOCTORS*

From Server to ANY Client

Protocol: LOGIN_ACCEPT

- **Usage:** When the DNI and password provided by a protocol *LOGIN* matches with those registered in the database, this protocol will be used as answer.
- **Returns:**
 - A *Patient* object will be returned, with all the information required, included the measures taken in previous sessions if this was called *PATIENTLOGIN_*
 - A *Doctor* object and a list of patients under their care if this was called from *DOCTORLOGIN_*
- **Answers to:**
 - *PATIENT_LOGIN*
 - *DOCTOR_LOGIN*
 - *ADMIN_LOGIN*

Protocol: LOGIN_DENY

- **Usage:** When the DNI and password provided by a protocol *LOGIN* doesn't match with those registered in the database, this protocol will be used as answer. When this protocol is sent, the connection will be closed, so the client needs to connect again.
- **Returns:** just the protocol.
- **Answers to:**
 - *PATIENT_LOGIN*
 - *DOCTOR_LOGIN*
 - *ADMIN_LOGIN*

Protocol: ACK

- **Usage:** Answer to protocols that don't require anything else, when there was no error executing the protocol.
- **Returns:** just the protocol.
- **Answers to:**
 - *PUSH_MEASUREMENT*
 - *PUSH_MEASUREMENT_COMMENT*
 - *REGISTER_PATIENT*
 - *REGISTER_DOCTOR*
 - *REGISTER_ADMIN*
 - *SERVER_SHUTDOWN*
 - *SERVER_CANCEL_SHUTDOWN*

Protocol: ERROR

- **Usage:** Answer to messages from the client when there is any error not related to them, usually this might be database errors. Meant to inform the client to try again some other time.
- **Returns:** just the protocol.
- **Answers to:**
 - *Any message from a client to the server*

Messages from either end

Protocol: DISCONNECT

- **Usage:** Whenever one side wants to close the connection for some reason.
- **Returns:** just the protocol.

Protocol: SERVER_SHUTDOWN_CONFIRM

- **Usage:** with administrator clients. Shut down the server even though other clients are connected if answering to SERVER_SHUTDOWN_CONFIRM and works as an acknowledgement if answering to SERVER_SHUTDOWN
- **Requirements:** none
- **Answers to:**
 - *SERVER_SHUTDOWN_CONFIRM*
 - *SERVER_SHUTDOWN*