# Just after the connection:

* The API sends all the entries to this new user:

[

{"id": 1, "description": "a", "number": 3, "last\_modif":"2020-10-01 12:00:00"},

{"id": 2, "description": "b", "number": 55, "last\_modif":"2020-10-01 18:52:27"}

]

* The client sends his pseudo:

{"userName","3noix"}

# For a lock request

Needed only for updates and deletions.

* The user asks like this:

{"rqtType": "lock", "rqtData": 2}

* The API answers to this user only:

{"type": "lock", "id": 2, "status": "success"}

* In case of error:

{"type": "lock", "id": 2, "status": "failure", "msg": "an error message"}

# For an unlock request

Needed only for updates.

* The user asks like this:

{"rqtType": "unlock", "rqtData": 2}

* The API answers to this user only:

{"type": "unlock", "id": 2, "status": "success"}

* In case of error:

{"type": "unlock", "id": 2, "status": "failure", "msg": "an error message"}

# For an insertion

* The user asks like this:

{

"rqtType": "insert",

"rqtData": {"description": "c", "number": 3}

}

* The API notifies all the users:

{

"type": "insert",

"entry": {"id": 5, "description": "c", "number": 3, "last\_modif": "2020-10-01 12:00:00"}

}

* In case of error the API sends such a message only to the original user:

{

"originalMsg": "insert request",

"errorMsg": "an error message"

}

# For an update

* The user asks like this:

{

"rqtType": "update",

"rqtData": {"id": 2, "description": "c", "number": 3}

}

* The API notifies all the users:

{

"type": "update",

"entry": {"id": 2, "description": "c", "number": 3, "last\_modif": "2020-10-01 12:00:00"}

}

* In case of error the API sends such a message only to the original user:

{

"originalMsg": "update request",

"errorMsg": "an error message"

}

# For a deletion

* The user asks like this:

{

"rqtType": "delete",

"rqtData": 2

}

* The API notifies all the users:

{

"type": "delete",

"id": 2

}

* In case of error the API sends such a message only to the original user:

{

"originalMsg": "delete request",

"errorMsg": "an error message"

}

# Other errors

* In case the message sent by the client was invalid, the API answers to him only:

{

"originalMsg": "what the client sent",

"errorMsg": "Invalid input data"

}