

# Publish/Subscribe Protocol

This document describes the protocol governing the communication between server and clients for the NSD assignments. The document first details the order of exchanges dictated by the protocol. Then the format of individual requests and responses is specified (by examples).

## Order of exchanges (per session)

1. Client sends Open request, identifying the channel to publish on.
2. Server responds with Success.
3. Client sends Publish, Subscribe, Unsubscribe or Get request.
4. In case of Get, server responds with MessageList, otherwise server responds with Success or Error.
5. Go to 3.

## JSON encoding

All client/server exchanges (as well as the messages contained within) are encoded as JSON objects. All these objects have a mandatory field `_class`, the String value of which indicates the type of object. JSON-encoded requests and responses are transmitted as single lines of text.

## Messages

A message is encoded as a JSON object with fields `from` (the sender of the message), `when` (a non-negative integer serving as the timestamp when the message was received by the server) and `body` (the text of the message).

```
{"_class":"Message", "from":"Bob", "when":47, "body":"Hello again!"}
```

Beyond these fields, a message may have additional fields, such as the field `pic` below (for a picture attachment, base64-encoded).

```
{"_class":"Message", "from":"Alice", "when":53, "body":"pic attached,  
base64 encoded, LOL",  
"pic":"iVBORw0KGgoAAAANSUhEUgAAABAAAAQCAMAAAAoLQ9TAAAx1BMVEUAAADWAEX/"}
```

## Requests

The server understands 5 types of requests. All requests include an `identity` field identifying the channel associated with the client.

- An **Open** request establishes the client's identity and creates a channel by that name if it does not already exist. The request also subscribes the client to its own channel. The request always succeeds.

```
{"_class":"OpenRequest", "identity":"Alice"}
```

- A **Publish** request publishes a message on the client's channel. The timestamp of the message is ignored; the server will issue its own timestamp for the message when storing it. The request fails if the channel does not exist or if the message is too big.

```
{"_class":"PublishRequest", "identity":"Alice",
"message":{"_class":"Message", "from":"Bob", "when":0, "body":"Hello
again!"}}
```

- A **Subscribe** request subscribes the client to the named channel. The request fails if the channel does not exist.

```
{"_class":"SubscribeRequest", "identity":"Alice", "channel":"Bob"}
```

- An **Unsubscribe** request unsubscribes the client from the named channel. The request fails if the channel does not exist.

```
{"_class":"UnsubscribeRequest", "identity":"Alice", "channel":"Bob"}
```

- A **Get** request retrieves all messages that were published on any channel to which the client is currently subscribed, in the order in which they were published. The field `after` specifies a timestamp, and the server will only retrieve messages that were published strictly after that timestamp. If the timestamp is 0 the server will retrieve all messages from the subscribed channels. The request never fails, but it may return an empty list of messages.

```
{"_class":"GetRequest", "identity":"Alice", "after":42}
```

## Responses

The server replies to requests with 3 types of responses.

- A **Success** response simply indicates that the request succeeded.

```
{"_class":"SuccessResponse"}
```

- An **Error** response indicates that the request failed and provides a reason why.

```
{"_class":"ErrorResponse", "error":"NO SUCH CHANNEL: Bob"}
{"_class":"ErrorResponse", "error":"MESSAGE TOO BIG: 1234
characters"}
{"_class":"ErrorResponse", "error":"INVALID REQUEST: [{}]"}
```

- A **MessageList** response is sent in reply to a **Get** request. The response provides a list of messages ordered by timestamp. The list of messages may be empty.

```
{"_class":"MessageListResponse", "messages":[]}
{"_class":"MessageListResponse", "messages":[{"_class":"Message",
"from":"Bob", "when":47, "body":"Hello again!"}, {"_class":"Message",
"from":"Alice", "when":53, "body":"pic attached, base64 encoded,
LOL",
"pic":"iVBORw0KGgoAAAANSUhEUgAAABAAAAQCAMAAAAoLQ9TAAAx1BMVEUAAADWAE
X/"}]}
```

