

Programming Exercise: Publish/Subscribe Server

What We're Looking For

- Your goal is to demonstrate how you would develop a production-quality software project, within the limits of an exercise. This includes the code you write but also the explanations you will include.
- This is not a fail/pass test. Different people have different skills, and we want to see what you're good at.

Hints

- Please ask us if you have any questions!
- We realize you're doing other things with your life. There is no calendar deadline for this exercise.
- This is just an exercise; you should spend no more than a day worth of effort to complete it. You can always, and ought to, add some text at the end explaining things you've left out or how you might improve it.
- Use whatever programming language, libraries and tools you're most comfortable with and will help you solve the problem in a reasonable amount of time.

What You'll be Building

You'll be implementing a server which exposes an HTTP-based API that can be used to subscribe to specific topics. When a client publishes messages to a specific topic, all subscribers receive that message.

Once a message has been published:

- A message must persist until all subscribers at the time of publishing have received it.
- A message should be removed once all subscribers have received it.

Messages need not persist across server restarts.

For example:

1. Alice and Bob subscribe to the kittens_and_puppies topic.
2. Charles sends a message to kittens_and_puppies saying "http://cuteoverload.files.wordpress.com/2014/10/unnamed23.jpg?w=750&h=1000".
3. Alice checks to see if she has any new messages in that topic and receives a copy of the message.
4. If Alice checks again, that message will not be there anymore.
5. The message is kept around until Bob checks for new messages or unsubscribes.
6. Once both Alice and Bob have received it, the message is deleted.

API

The HTTP server should implement the following API:

Subscribe to a topic

Request: POST /<topic>/<username>

Response codes:

- 200: Subscription succeeded.

Unsubscribe from a topic

Request: DELETE /<topic>/<username>

Response codes:

- 200: Unsubscribe succeeded.
- 404: The subscription does not exist.

Publish a message

Request: POST /<topic>

Request body: The message being published.

Response codes:

- 200: Publish succeeded.

Retrieve the next message from a topic

Request: GET /<topic>/<username>

Response codes:

- 200: Retrieval succeeded.
- 204: There are no messages available for this topic on this user.
- 404: The subscription does not exist.

Response body: The body of the next message, if one exists.

Deliverables

Please include:

- Source code for your solution.
- Instructions on deploying and running it.
- An explanation of the steps you took to ensure this is production-quality code as well as what you left out and would do given more time.