Web Studio: Week 8 Turn-in #7

Name: Ivy Chen
UNI: ic2389
Due: Friday 10/26 at 2PM
COMS6998 Adv. Web Studio

Original Goals

High Level Goal

1. Allow users to save messages to their personal list.

Low Level Goal

1. Disable messages from being sent to everyone
2. Enable users to add messages to their own list
3. Present messages in reverse chronological order
4. Limit number of messages on screen at one time
5. Show who a reply is directed towards
6. Add infinite scrolling for messages
7. Enable direct messaging

Lessons Learned Through Iteration

Report on 3 of the things that you learned that caused you to iterate. Consider the following format: My plan was to __x___. But I ran into problem __y___. And I solved it by doing __z____.
Where z = "I added a new subgoal", "I changed my high level goal", "I removed a subgoal" Each of the three things should take 1-2 paragraphs to describe. Images are encouraged.

1. I think this week I spent the most time iterating on a problem domain to tackle. My first task was to identify a domain of communication where I could fix a problem or enhance an existing practice. I initially thought about chat apps and the features they provided, and thought of ideas such as implementing additional

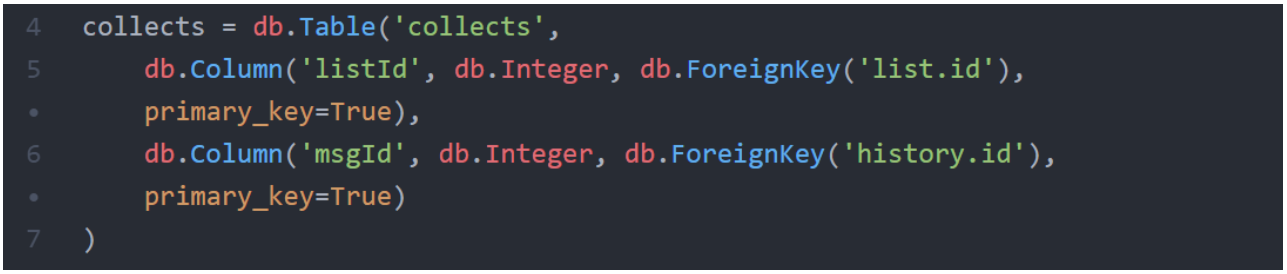
features in chat. After observing some of my own issues with chat apps and asking friends about their problems I realized that some of the major complaints were summarizing key takeaways/action items from chats, and collaboration.

Thus, after iterating on a couple ideas (collaborative food ordering, design canvas, etc.), my high level goal is to enable users to add messages to a personal list ie. a todo-list or just a log of favorite messages. (My stretch goal is to allow for collaborative to-do lists within a group chat, but I'm not sure about implementation at the moment.)

2. One of my low-level goals was to enable direct messaging and disable broadcast messages from appearing on all individual user message pages. I searched for information on how to accomplish this and found some resources for socket.io's rooms and keeping track of users and their sessions. But I wasn't sure how to integrate rooms and namespaces (also seemed like an overhaul of the existing structure) and didn't have enough time to investigate on top of adding other features.

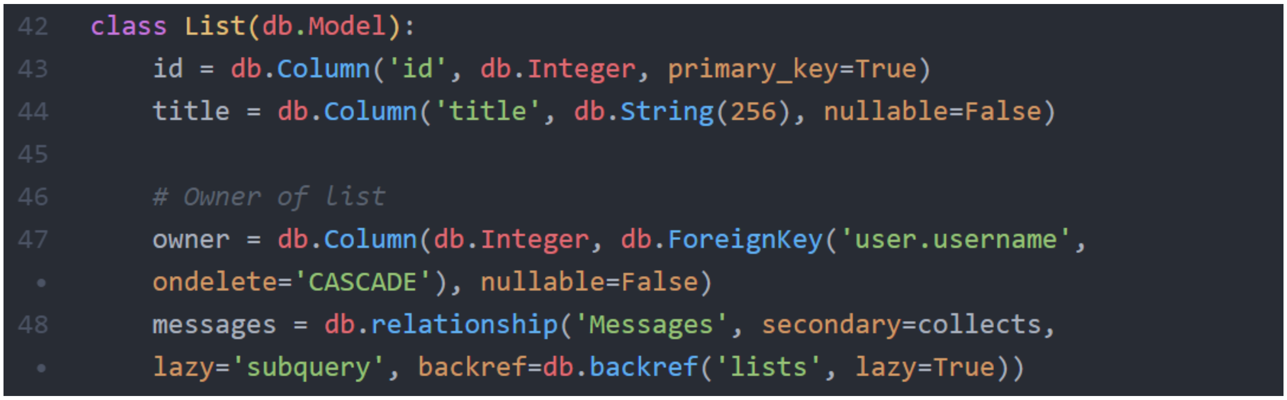
   I decided to make enabling direct messaging and fixing the broadcast bug a stretch-goal/defer it while I first build out the list feature that directly accomplishes my high-level goal.

3. One of my low level goals was "Enable users to add messages to their own list". I realized that it was still pretty high-level as a goal and can be broken down into further smaller goals that can help me organize the implementation better.
   So I broke it up into two smaller goals:
   - Create user lists and table for messages in list
   - Create button and event to add message to list



```
collects = db.Table('collects',
    db.Column('listId', db.Integer, db.ForeignKey('list.id'),
    primary_key=True),
    db.Column('msgId', db.Integer, db.ForeignKey('history.id'),
    primary_key=True)
)
```

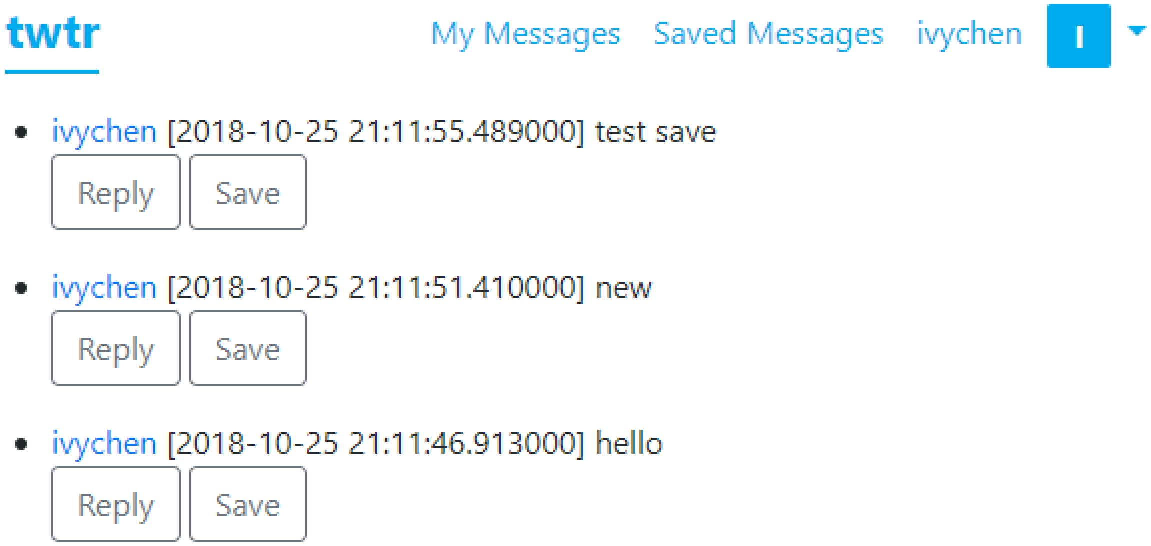
Links messages to lists



```
class List(db.Model):
    id = db.Column('id', db.Integer, primary_key=True)
    title = db.Column('title', db.String(256), nullable=False)

    # Owner of list
    owner = db.Column(db.Integer, db.ForeignKey('user.username', ondelete='CASCADE'), nullable=False)
    messages = db.relationship('Messages', secondary=collects, lazy='subquery', backref=db.backref('lists', lazy=True))
```

Lists owned by user



- ivychen [2018-10-25 21:11:55.489000] test save

- ivychen [2018-10-25 21:11:51.410000] new

- ivychen [2018-10-25 21:11:46.913000] hello

Save button

I think in a future iteration it would be cool to give users/groups more flexibility and customization with their lists as well.

Goal Progress

Status Summary

✓ = DONE
🔜 = PARTIAL/DEFERRED
✗ = NOT DONE (REMOVED)

High Level Goal	Status
1. Allow users to save messages to their personal list.	✓

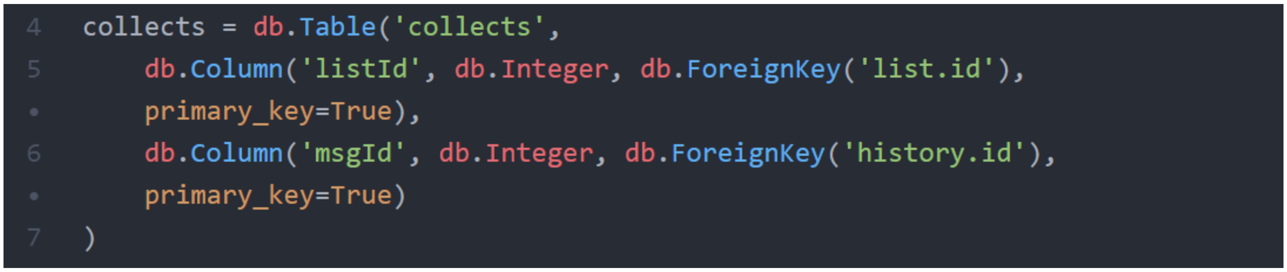
Low Level Goals	Status
1. Disable messages from being sent to everyone	🔜
2. Create user lists and table for messages in list	✓
3. Present messages in reverse chronological order	✓
4. Limit number of messages on screen at one time	✓
5. Show who a reply is directed towards	✓
6. Add infinite scrolling for messages	✓
7. Enable direct messaging	🔜
8. Create button and event to add message to list	✓

High Level Goal

1. Allow users to save messages to their personal list.

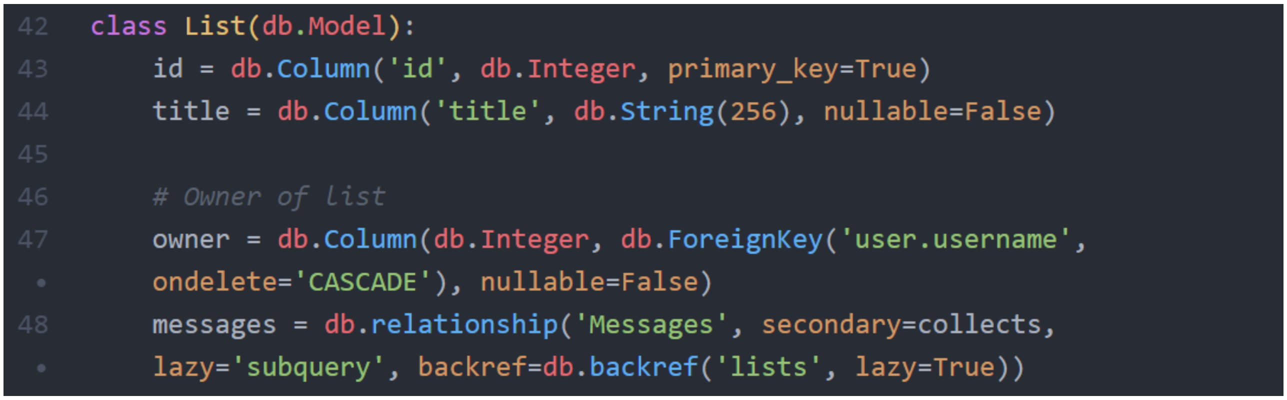
Low Level Goal

1. Disable messages from being sent to everyone (Not Done)
2. Create user lists and table for messages in list
   Created List and collects table to represent user lists and collections of messages within a list. When displayed on a list’s page, the messages are sorted in reverse chronological order.



```
collects = db.Table('collects',
    db.Column('listId', db.Integer, db.ForeignKey('list.id'),
    primary_key=True),
    db.Column('msgId', db.Integer, db.ForeignKey('history.id'),
    primary_key=True)
)
```

Links messages to lists

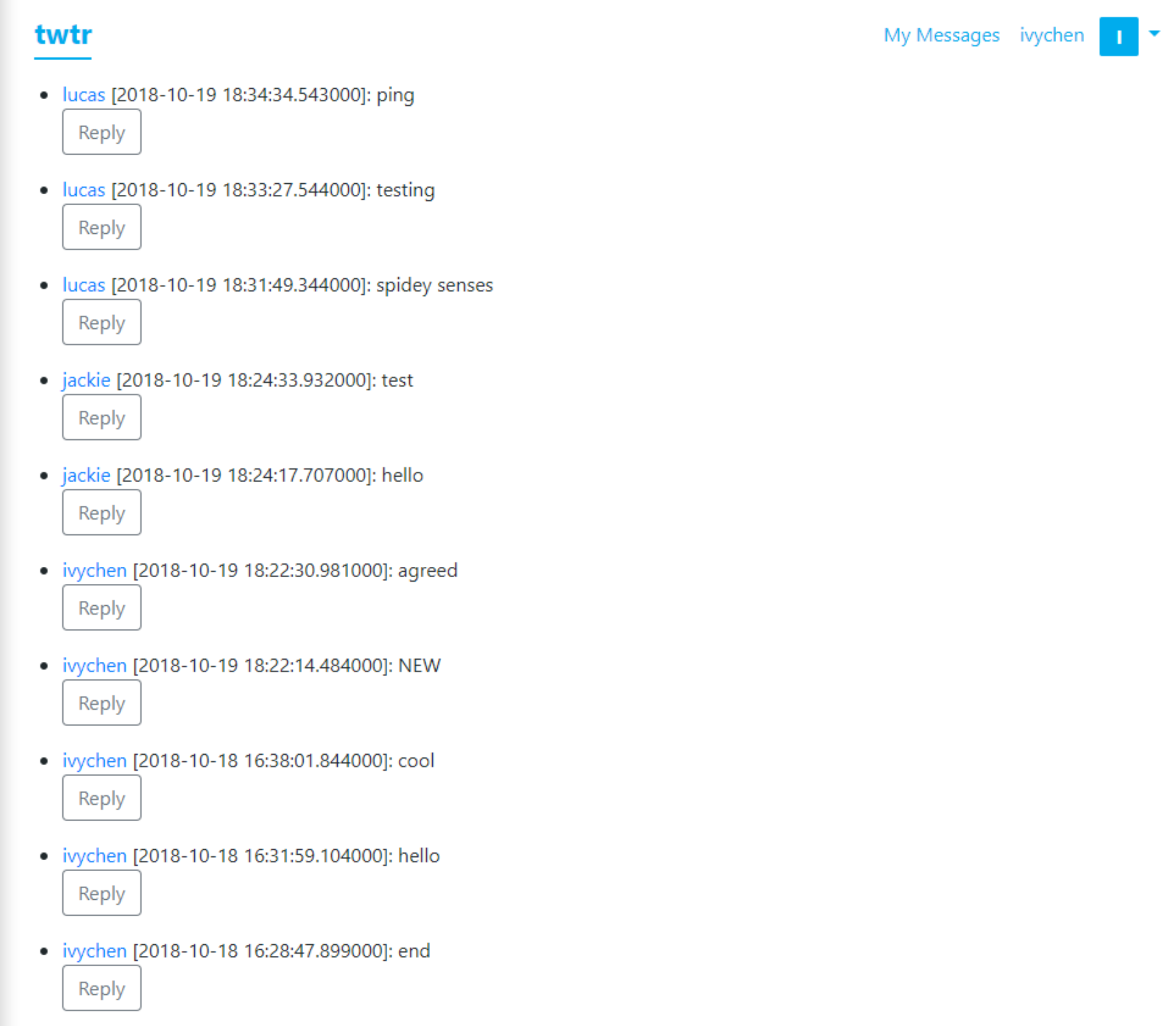


```
class List(db.Model):
    id = db.Column('id', db.Integer, primary_key=True)
    title = db.Column('title', db.String(256), nullable=False)

    # Owner of list
    owner = db.Column(db.Integer, db.ForeignKey('user.username',
    ondelete='CASCADE'), nullable=False)
    messages = db.relationship('Messages', secondary=collects,
    lazy='subquery', backref=db.backref('lists', lazy=True))
```

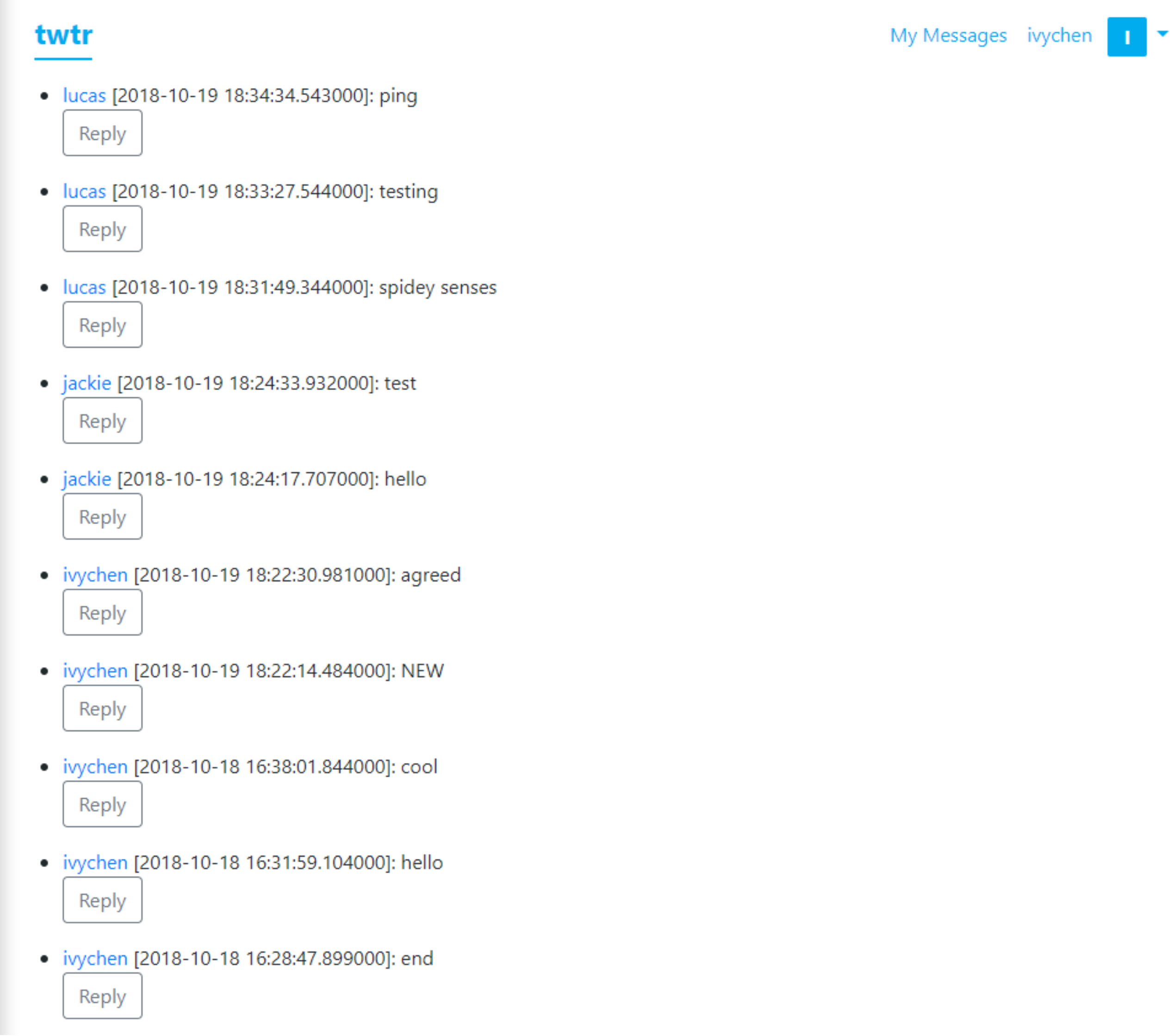
Lists owned by user

3. Present messages in reverse chronological order
   Used SQLAlchemy to reverse the message order by timestamp.



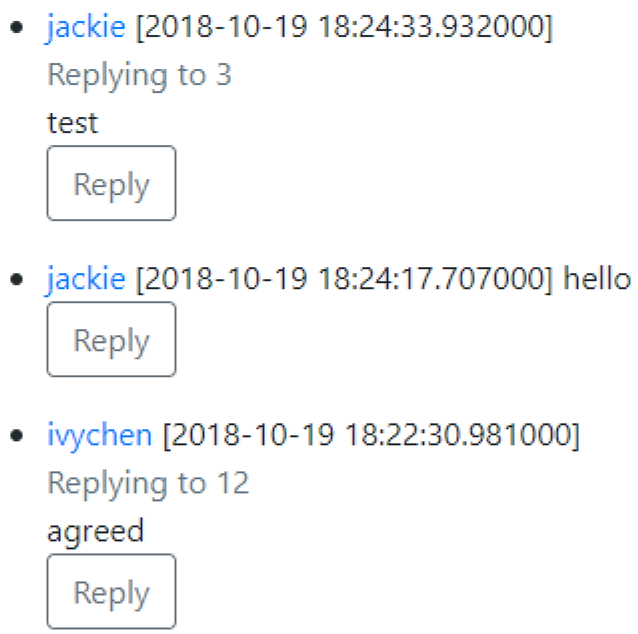
- lucas [2018-10-19 18:34:34.543000]: ping
- lucas [2018-10-19 18:33:27.544000]: testing
- lucas [2018-10-19 18:31:49.344000]: spidey senses
- jackie [2018-10-19 18:24:33.932000]: test
- jackie [2018-10-19 18:24:17.707000]: hello
- ivychen [2018-10-19 18:22:30.981000]: agreed
- ivychen [2018-10-19 18:22:14.484000]: NEW
- ivychen [2018-10-18 16:38:01.844000]: cool
- ivychen [2018-10-18 16:31:59.104000]: hello
- ivychen [2018-10-18 16:28:47.899000]: end

4. Limit number of messages on screen at one time
   I limited the number of messages on the page to 8 using the SQLAlchemy pagination feature.



- lucas [2018-10-19 18:34:34.543000]: ping
  Reply
- lucas [2018-10-19 18:33:27.544000]: testing
  Reply
- lucas [2018-10-19 18:31:49.344000]: spidey senses
  Reply
- jackie [2018-10-19 18:24:33.932000]: test
  Reply
- jackie [2018-10-19 18:24:17.707000]: hello
  Reply
- ivychen [2018-10-19 18:22:30.981000]: agreed
  Reply
- ivychen [2018-10-19 18:22:14.484000]: NEW
  Reply
- ivychen [2018-10-18 16:38:01.844000]: cool
  Reply
- ivychen [2018-10-18 16:31:59.104000]: hello
  Reply
- ivychen [2018-10-18 16:28:47.899000]: end
  Reply

5. Show who a reply is directed towards
   Added a reply tag for messages that are replies in order to show that a message is a reply to someone else. At the moment, the reply tag shows the message id of its parent, but I'll modify it to be a more helpful identifier (ie. name)

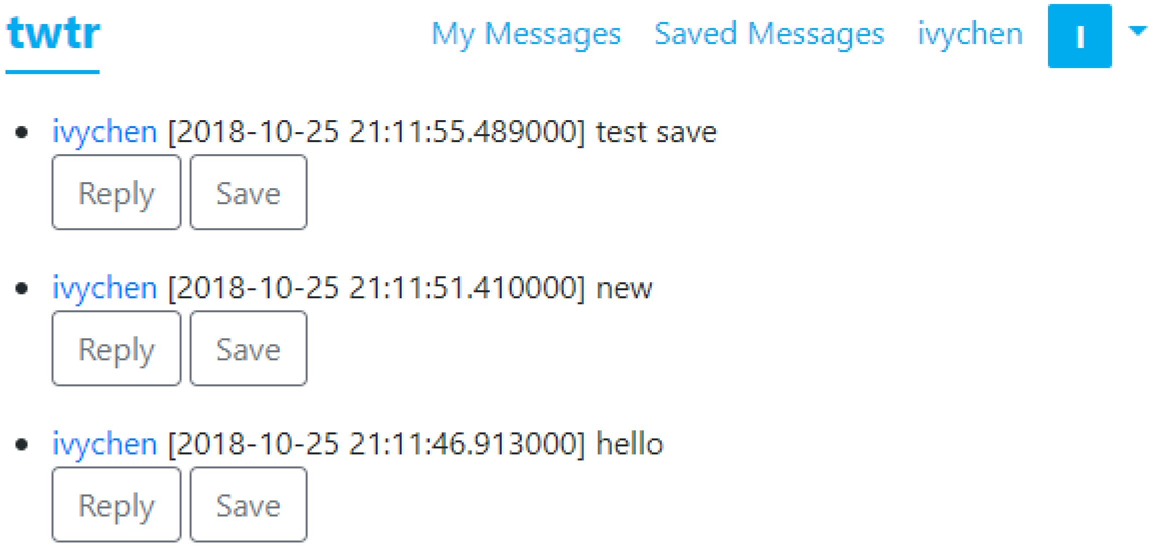


- jackie [2018-10-19 18:24:33.932000]
  Replying to 3
  test
- jackie [2018-10-19 18:24:17.707000] hello
- ivychen [2018-10-19 18:22:30.981000]
  Replying to 12
  agreed

6. Add infinite scrolling for messages
   Added pagination according the guide here: https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-ix-pagination/page/0. However, in order to use infinite scrolling, I used the Infinite-Scroll JS library here: https://infinite-scroll.com.

7. Enable direct messaging (Not Done)

8. Create button and event to add message to list



- ivychen [2018-10-25 21:11:55.489000] test save
- ivychen [2018-10-25 21:11:51.410000] new
- ivychen [2018-10-25 21:11:46.913000] hello

Created save button and a page that has all saved messages