These are the JSON message standards

I will be making a jSON api

**Receiving information(from the server):**

**Picture Post**

* “Picture” : Base 64 encoded string of a buffered Image
* “ID” : String Post ID
* “Date” : String Date
* “Tags” : String[] of tags
* “Rating” : int for rating based on votes

**Video Post**

* “Video” : BAse 664 encoded string of the video
* “ID” : String Post ID
* “Date” : String Date
* “Tags” : String[] of tags
* “Rating” : int for rating based on votes

**Text Post**

* “Text” : String of text to be converted to image
* “ID” : String Post ID
* “Date” : String Date
* “Tags” : String[] of tags
* “Rating” : int for rating based on votes

**Comments**

* “Comments[]” : String [] of comments
* “ID” : String Post ID

**Ranking**

* “Rank” : int ranking
* “ID” : String Post ID

**Sending Information(sending to Server):**

**Picture Post**

* “Picture” : Base 64 encoded string of a buffered Image
* “Lat” : Double value of the latitude
* “Lon” : Double value of the longitude
* “ID” : String Post ID
* “User” : String ID of the User
* “Date” : String Date
* “Tags” : String[] of tags

**Video Post**

* “Video” : BAse 664 encoded string of the video
* “Lat” : Double value of the latitude
* “Lon” : Double value of the longitude
* “ID” : String Post ID
* “User” : String ID of the User
* “Date” : String Date
* “Tags” : String[] of tags

**Text Post**

* “Text” : String of text to be converted to image
* “Lat” : Double value of the latitude
* “Lon” : Double value of the longitude
* “ID” : String Post ID
* “User” : String ID of the User
* “Date” : String Date
* “Tags” : String[] of tags

**Downvote**

* “Downvote” : No Needed information here
* “ID” : String Post ID
* “User” : String ID of the User
* “Previous” : String either “Up”, “Down”,”No”

**Upvote**

* “Upvote” : No Needed information here
* “ID” : String Post ID
* “User” : String ID of the User
* “Previous” : String either “Up”, “Down”,”No”

**UnUpvote**

* “UnUpvote” : No Needed information here
* “ID” : String Post ID
* “User” : String ID of the User

**UnDownvote**

* “UnDownvote” : No Needed information here
* “ID” : String Post ID
* “User” : String ID of the User

**Flag**

* “Flag” : No Needed Information here
* “ID” : String Post ID
* “User” : String ID of the User

**UnFlag**

* “UnFlag” : No Needed Information here
* “ID” : String Post ID
* “User” : String ID of the User

**Comment Post**

* “Comment” : String Comment text
* “ID” : String Post ID
* “User” : String Id of the User
* “Date” : String Date

**Request For Posts**

* “Bottom” : The String ID of the last post loaded on the client
* “Fliter” : String Filter
* “Lat” : Double value of the latitude
* “Lon” : Double value of the longitude

**Request for Comments**

* “Comments[]” : This is a place holder to direct packet
* “ID” : String Post ID

**Request to remove post**

* “Removal” : This is a place holder to direct packet
* “ID” : String Post ID

**Misc. Info:**

* A note about video posts: I have yet to nail down an efficient way to transmit the videos in packets. It may be needed to split the packet into several transmissions to make the transfer smoother. This is something that Will need to be tested before it can be nailed down and is more of a note for the future.
* A note about upvotes and downvotes: It will be important to track which users have up and down voted posts. I think that tracking a list of which posts have been up/downvoted by each user is the best way to do this. That way the information is personal to the user and they won't be able to up/down vote the same post multiple times. This goes for flags as well. Also, There will be a need to send an updated rating count after an upvote/downvote is sent to server.
* A note about undoing up and down votes: If a user wants to cancel out their previous vote, they should be able to tap the same vote, and it should adjust the rating accordingly and remove the post from their list of voted for content.
* A note about voting: If a user has previously voted, and presses the same vote again, nothing should happen. if they press the opposite vote, the system should undo the previous vote and update it to the new one.
* A note about comments: I’m not sure how were going to be storing comments yet. I think it will be ideal to store them separately with their own IDs. I think it would be rather simple to have the comment threads IDs be the same as the post ID but with a ‘C’ on the front of it. And if we want to be able to remove/access/change comments, we can have the specific comment ID be the comment threads ID followed by a ‘:’ and the the specific comment ID. For example: The post Id is 1a2b3c. The comment thread ID would be C1a2b3c. The fifth comment might have ID C1a2b3c:5.If this to be implemented, then the post ID cannot be allowed to have to start with a ‘C’. We will also want to tie a comment to the user if we choose to allow for comment removal/editing/or pretty much anything.
* A note about requesting posts: If it is the first time loading the app, there will be no bottom post and the string “Bottom” will be NULL. This will be the indicator that it is the first time and to grab from the top.
* A note about date format: it is important to keep the date consistent. (mm/dd/yyyy:hh:mm) could be a good format. It would also probably be easier to use whatever the android date/time format is by default.
* A note about Filters: We will need to nail down what Filters we want and what the format will be.
  + Top Rated : “Top”
  + Newest : “New”
  + My Posts : “<the user ID>”
  + Tag Lookup : “#<String name of tag>
* A note about looking up by tags: If we have a default 10 posts to load initially, and a specific tag may have fewer than that.
* A note about removing content: If a user wishes to remove their post, there should be a section in the app to see their posts that will have a remove button.This way only the poster can request to remove content and we don't have to change the ui for a post that the user posted in the main feed.