API Design Assignment 3

Evelyn Wu (zhixuanw@andrew.cmu.edu)

November 15, 2023

Link to repo

https://github.com/zhixuanevelynwu/api-a3-gRPC.git

Protocol Buffer Definitions

JavaScript

```
message User {
  // a human readable id
  string id = 1;
}

message Post {
  // a human readable id
  string id = 1;

  // title, some text, one of video or image url
  string title = 2;
  string content = 3;
  oneof media {
    string image_url = 4;
    string video_url = 5;
  }

  // optional author
  optional string user_id = 6;

  // a score, can be negative
  int32 score = 7;

  // a state: normal, locked, hidden
  enum PostState {
    NORMAL = 0;
    LOCKED = 1;
    HIDDEN = 2;
  }
  PostState state = 8;

  // a publication date
  string publication_date = 9;
}

message Comment {
  // a human readable id
  string id = 1;

  // the post or comment it is under
  oneof parent {
    string parent_post_id = 2;
    string parent_comment_id = 3;
  }

  // must have an author
  string user_id = 4;

  // must have some text
  string content = 5;

  // have a score
  int32 score = 6;

  // can be normal or hidden
  enum CommentState {
    NORMAL = 0;
    HIDDEN = 1;
  }
  CommentState state = 7;

  // a publication date
  string publication_date = 8;
}

message CreatePostRequest {
  // title, some text, one of video or image url
  string title = 1;
  string content = 2;
  oneof media {
    string image_url = 3;
    string video_url = 4;
  }

  // optional author
  optional string user_id = 5;
}

message VotePostRequest {
  // the post id
  string id = 1;
}

message RetrievePostRequest {
  // the post id
  string id = 1;
}

message CreateCommentRequest {
  // the post or comment it is under
  oneof parent {
    string parent_post_id = 1;
    string parent_comment_id = 2;
  }

  // must have an author
  string user_id = 3;

  // must have some text
  string content = 4;
}

message VoteCommentRequest {
  // the comment id
  string id = 1;
}

message RetrieveTopNCommentsRequest {
  // post id to retrieve comments for
  string id = 1;

  // how many comments to retrieve
  int32 N = 2;
}

message CommentWithHasReplies {
  Comment comment = 1;
  bool has_replies = 2;
}

message RetrieveTopNCommentsResponse {
  repeated CommentWithHasReplies comments_with_has_replies = 1;
}

message ExpandCommentBranchRequest {
  // comment id to expand
  string id = 1;

  // how many comments to retrieve
  int32 N = 2;
}

message CommentWithReplies {
  Comment comment = 1;
  repeated Comment replies = 2;
}

message ExpandCommentBranchResponse {
  // top n comments with top n replies
  repeated CommentWithReplies comments_with_replies = 1;
}
```

Service definitions, alongside with input/output Protocol Buffers

JavaScript

```
service RedditService {
  rpc CreatePost(CreatePostRequest) returns (Post);
  rpc UpvotePost(VotePostRequest) returns (Post);
  rpc DownvotePost(VotePostRequest) returns (Post);
  rpc RetrievePostContent(RetrievePostRequest) returns (Post);
  rpc CreateComment(CreateCommentRequest) returns (Comment);
  rpc UpvoteComment(VoteCommentRequest) returns (Comment);
  rpc DownvoteComment(VoteCommentRequest) returns (Comment);
  rpc RetrieveTopNComments(RetrieveTopNCommentsRequest) returns
(RetrieveTopNCommentsResponse);
  rpc ExpandCommentBranch(ExpandCommentBranchRequest) returns
(ExpandCommentBranchResponse);
}
```

Storage backend being used

I used a dummy storage (in-memory structures). I generated around 100 posts, 150 comments, and various replies for each comment. You can find the data generation code at db.py.

Two links, pointing to implementation of the Server and the Client

- Server: https://github.com/zhixuanevelynwu/api-a3-gRPC/blob/main/reddit/reddit_server.py
- Client: https://github.com/zhixuanevelynwu/api-a3-gRPC/blob/main/reddit/reddit_client.py

Links to the implementation of the high level function and its test

- Function: https://github.com/zhixuanevelynwu/api-a3-gRPC/blob/main/reddit/highlevel.py
- Test: https://github.com/zhixuanevelynwu/api-a3-gRPC/blob/main/reddit/test_highlevel.py