

KẾ HOẠCH LÀM VIỆC TUẦN 2

**Viện Công nghệ thông tin và Truyền thông
2021**

Nhắc lại

Tổng cộng có các API sau:

signup, login, logout, add_post, get_post, get_list_posts, check_new_item, edit_post, delete_post, report, set_comment, get_comment, like, edit_comment, del_comment, search, set_request_friend, get_requested_friend, set_accept_friend, get_user_friends, change_password, set_block_user, set_block_diary, get_conversation, delete_message, get_list_conversation, delete_conversation

Các API thêm vào:

set_official_account, check_verify_code, del_saved_search,
get_list_suggested_friends, get_verify_code,
get_saved_search,

admin_api: get_admin_permission, get_user_list, set_role,
get_analyst_result, set_user_state, delete_user,
get_user_basic_info

Nhắc lại

- Đã nắm được tổng quan đề tài và quy trình bảo vệ
- Đã nắm được một số API thuộc nhóm đăng ký-đăng nhập
- Tuần này sẽ giới thiệu API đầu tiên của nhóm API đăng bài

Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

Mục lục

1. **API add post**
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

Add post

| Tên parameter | Type | NN | Mô tả |
|---------------|---------------|----|--|
| token | string | O | |
| image | array of file | X | mảng các file ảnh, yêu cầu có ảnh hoặc video |
| video | file | X | file video, hoặc url video |
| described | string | O | |

| Tên parameter | Type | NN | Mô tả |
|---------------|--------|----|----------------------------------|
| code | string | O | (tham khảo phần response common) |
| message | string | O | |
| data | string | O | |
| id | string | O | id bài |
| url | string | O | url bài |

Mô tả Add post

Chức năng cho phép đăng một bài viết lên tường người dùng.

Input:

- **image** (không bắt buộc, ở dạng File)
- **token**
- **video** (không bắt buộc, đã có ảnh thì không video và ngược lại)
- **described** (văn bản)

Kết quả: 1000|OK - Nếu đăng thành công thì sẽ hiển thị bài viết mới tại vị trí bài đăng đầu tiên trên giao diện nhật ký hoặc tại bài đăng đầu tiên trên giao diện trang cá nhân tùy theo vị trí đăng bài tương ứng.

Mục lục

1. API add post
2. **Giao diện add post**
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

Giao diện Add post(1)



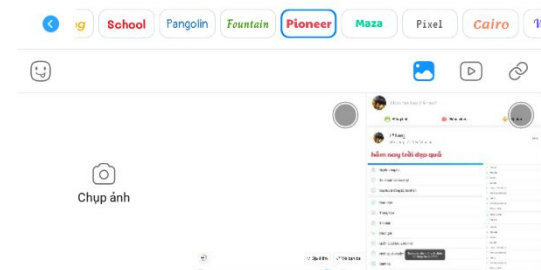
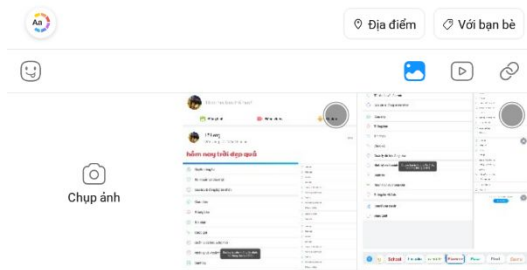
Chỉ có một chế độ hiển thị công khai.

Không gắn thẻ bạn bè.

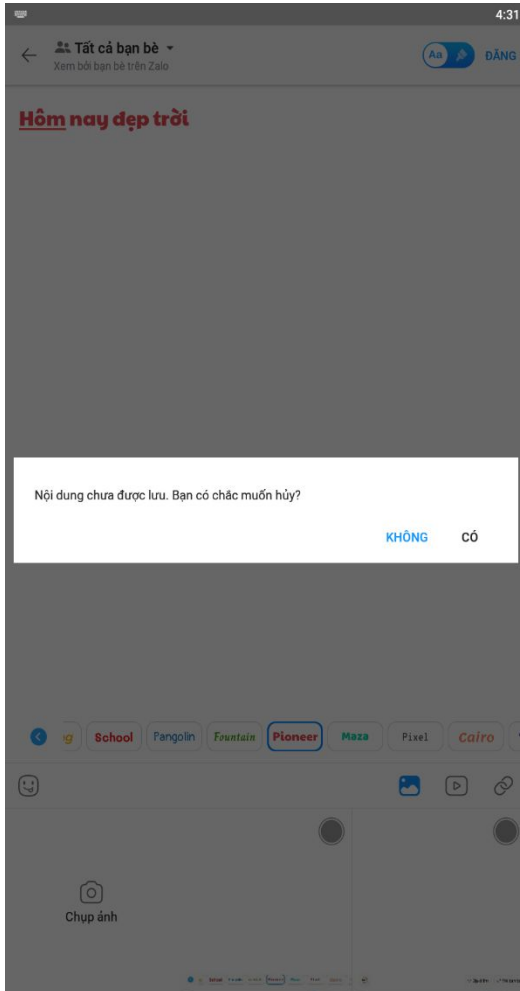
Không có chức năng checkin.

Không tùy chọn màu nền của bài viết.

Không tùy chọn màu chữ, font chữ.



Giao diện Add post(2)

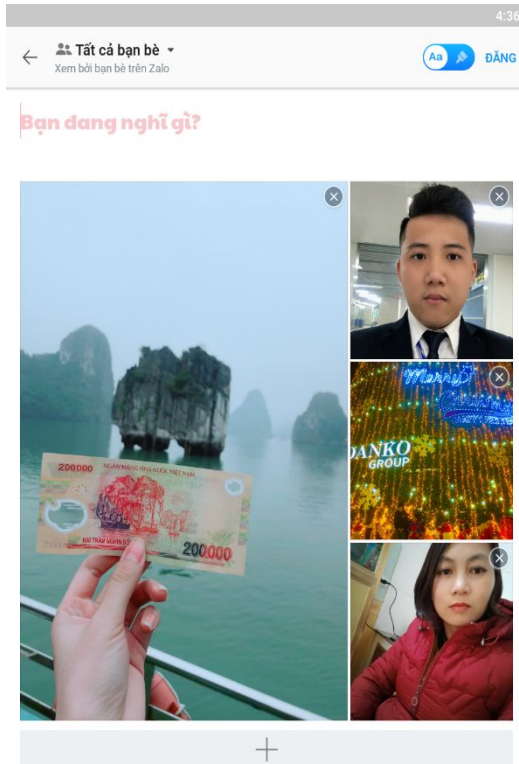


Nếu đang biên soạn bài viết nhưng muốn chuyển sang màn hình khác; sẽ có popup menu hiện ra với hai tùy chọn Tiếp tục hoặc Hủy bài viết.

Một khi bài viết được đăng, sẽ được chuyển sang các giao diện sau đây:

- Trang chủ với bài viết vừa đăng ở trên đầu tiên nếu đăng ở trang chủ.
- Trang cá nhân với bài viết vừa đăng ở trên đầu tiên nếu đăng ở trang cá nhân.

Giao diện Add post(3)



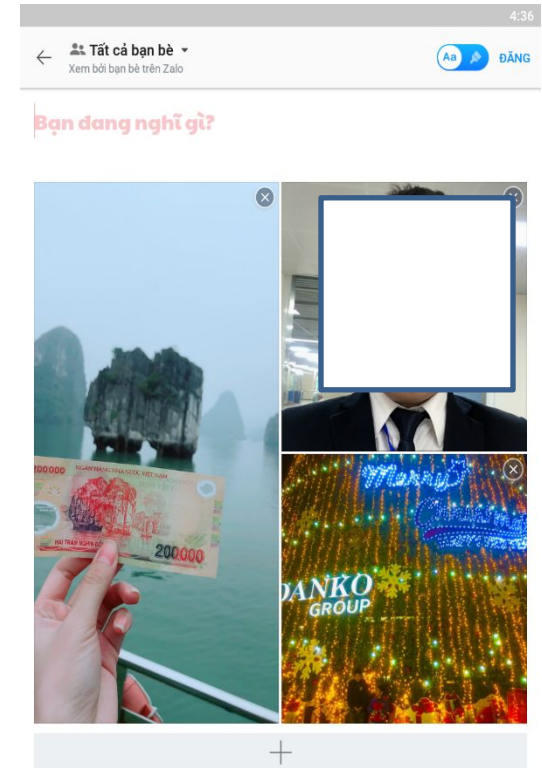
Chỉ có một chế độ hiển thị công khai.

Không gắn thẻ bạn bè.

Không có chức năng checkin.

Không tùy chọn màu nền của bài viết.

Không tùy chọn màu chữ, font chữ.



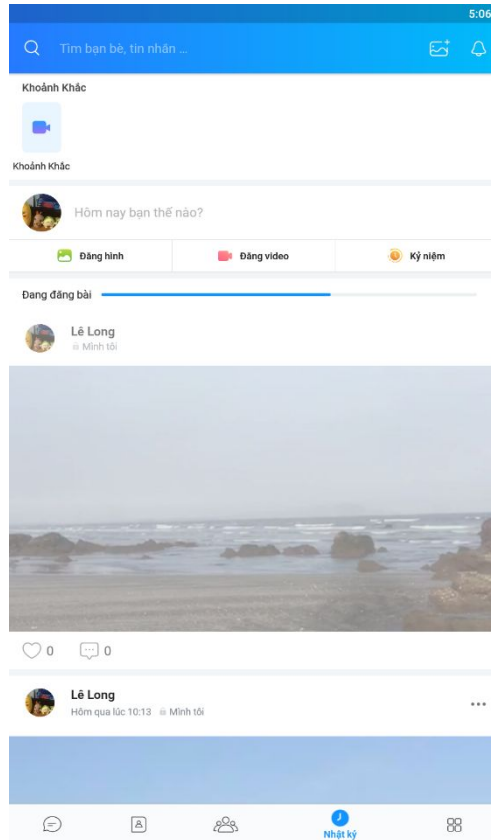
Giao diện Add post(4)

Nếu nhấn nút đăng xong mà người dùng chuyển sang trang khác thì việc đăng lên sẽ được hoạt động ngầm?

Khi đăng xong, sẽ kiểm tra xem màn hình hiện tại là màn hình nào để ứng dụng có thể có các cách chèn bài (hoặc xử lý) cho phù hợp.

Không cho phép đăng các video quá ngắn (chưa được 1s thời lượng).

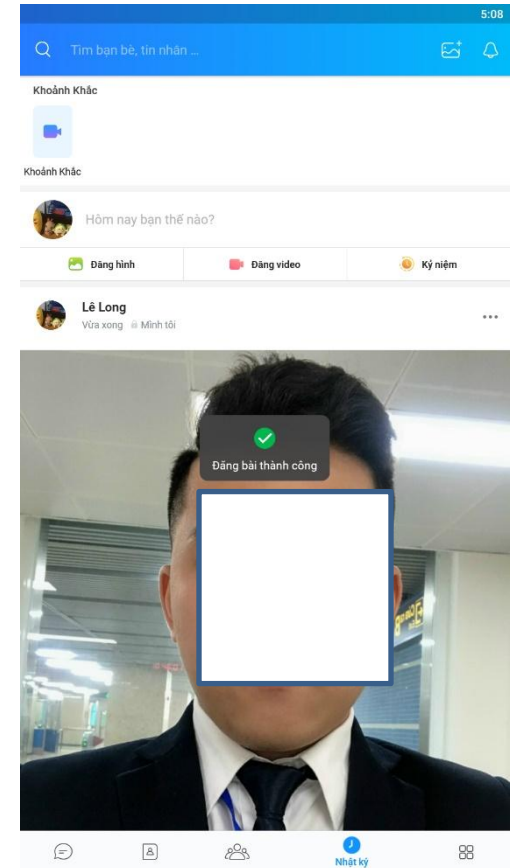
Giao diện Add post(5)



Với những bài có nhiều ảnh và video có dung lượng lớn thì việc tải lên sẽ mất nhiều thời gian.

Nếu được, có thể làm kiểu màn hình con ở góc trên có hiển thị progress bar dạng hình tròn (hoặc thẳng).

Khi quá trình tải lên kết thúc, sẽ hiện ra popup thông báo “Đăng bài thành công”.



Giao diện Add post(6)

Các bức ảnh được đăng trong bài viết tối đa là 4 ảnh

Chỉ có một bố cục bố trí các ảnh là bố cục cổ điển (dạng lưới).

Khi đăng bài, người viết bài có khả năng bổ sung thêm hoặc bỏ bớt đi các ảnh trong bài.

Một khi bức ảnh được đăng lên, hiển thị phần trung tâm của ảnh.

Giữa các ảnh có bố cục một đường viền màu trắng

Có thể sinh viên sẽ gặp khó khăn khi thực hiện việc bố cục lại các bức ảnh sau mỗi lần người dùng thêm hoặc bớt đi một bức ảnh.

Giao diện Add post(7)

Khi đó cho phép sinh viên thực hiện việc bố cục 4 bức ảnh vào 4 ô cạnh nhau được sắp xếp sẵn. Lúc ấy 4 ô này khi chưa nạp ảnh khác vào thì chúng sẽ hiển thị một ảnh mặc định nào đó. Với dấu + ở chính giữa bức ảnh.

Các bức ảnh được đăng trong bài viết tối đa là 4 ảnh

Chỉ có một bố cục bố trí các ảnh là bố cục cổ điển (dạng lưới).

Khi đăng bài, người viết bài có khả năng bổ sung thêm hoặc bỏ bớt đi các ảnh trong bài.

Một khi bức ảnh được đăng lên, hiển thị phần trung tâm của ảnh.

Giao diện Add post(8)

Giữa các ảnh có bố cục một đường viền màu trắng

Có thể sinh viên sẽ gặp khó khăn khi thực hiện việc bố cục lại các bức ảnh sau mỗi lần người dùng thêm hoặc bớt đi một bức ảnh.

Khi đó cho phép sinh viên thực hiện việc bố cục 4 bức ảnh vào 4 ô cạnh nhau được sắp xếp sẵn. Lúc ấy 4 ô này khi chưa nạp ảnh khác vào thì chúng sẽ hiển thị một ảnh mặc định nào đó. Với dấu + ở chính giữa bức ảnh.

Giao diện Add post(9)

Nhắc lại, nếu kích thước bức ảnh không vừa khít với ô thì sẽ hiển thị phần trung tâm của bức ảnh (không bóp méo bức ảnh cho vừa với khung hình).

Cần chú ý về thứ tự của ảnh được đăng lên và thứ tự của ảnh được hiển thị (có liên quan đến màn hình hiển thị bài).

Tình huống như sau: người dùng tải lên các ảnh theo thứ tự trước sau là: ảnh A, ảnh B, ảnh C.

Khi đó, server phải ghi nhớ thứ tự các ảnh này mà trả về kết quả cho các màn hình hiển thị trang cá nhân (hoặc thậm chí trang chủ).

Các test case cho Add post(1)

1. Người dùng nhập file đúng định dạng, nội dung không trống và dưới số lượng từ cực đại.
Kết quả mong đợi: Đăng bài thành công.
2. Người dùng nhập các dữ liệu khác hợp lệ nhưng ảnh không hợp lệ do tổng dung lượng quá lớn.
Kết quả mong đợi: ứng dụng cần kiểm tra ngay phía mình để đảm bảo không gửi dữ liệu không chính xác.
Thông báo cần hiển thị là dung lượng ảnh quá lớn.

Các test case cho Add post(2)

3. Người dùng truyền mã phiên hợp lệ, các dữ liệu khác hợp lệ nhưng video không hợp lệ do thời lượng hoặc dung lượng quá lớn.
Kết quả mong đợi: ứng dụng cần kiểm tra ngay phía mình để đảm bảo không gửi dữ liệu không chính xác. Thông báo cần hiển thị là thời lượng hoặc dung lượng video quá lớn.
4. Người dùng đã nhập đủ các dữ liệu khác hợp lệ nhưng trong quá trình đăng bài thì mạng bị ngắt kết nối.
Kết quả mong đợi: ứng dụng cần kiểm tra ngay phía mình để đảm bảo đưa ra thông báo về kết nối mạng bị ngắt.

Mục lục

1. API add post
2. Giao diện add post
3. **API get post**
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

get_post (1)

| Tên parameter | Type | NN | Mô tả |
|---------------|--------|----|------------|
| token | string | 0 | |
| id | string | 0 | id của bài |

| Tên parameter | Type | NN | Mô tả |
|---------------|--------|----|--|
| code | string | 0 | (tham khảo phần response common) |
| message | string | 0 | |
| data | string | 0 | |
| id | string | 0 | |
| described | string | 0 | |
| created | string | 0 | |
| modified | string | 0 | |
| like | string | 0 | số lượng like bài |
| comment | string | 0 | số lượng comment bài |
| is_liked | string | 0 | kiểm tra User đã like bài này chưa. 1: đã like, 0: chưa like |

get_post (2)

| | | | |
|-------------|--------|---|---|
| image | string | X | url hình ảnh |
| id | string | X | id của ảnh |
| url | string | X | url của ảnh |
| video | string | X | |
| url | string | X | url video, nếu có video thì thôi ảnh, có ảnh thì thôi video |
| thumb | string | X | thumbnail của video |
| author | string | 0 | |
| id | string | 0 | |
| name | string | 0 | |
| avatar | string | 0 | |
| state | string | 0 | trạng thái của người viết |
| is_blocked | string | 0 | kiểm tra author có block userid không |
| can_edit | string | 0 | kiểm tra token có thể edit bài hay không |
| banned | string | 0 | |
| can_comment | string | X | có thể comment được bài viết không |

Mô tả get_post (1)

API thực hiện việc đọc một bài viết của một tài khoản của người dùng

Request dạng **POST**

Tham số: **token**, **id** (của bài viết)

Kết quả đầu ra: Nếu không thành công thì sẽ có các thông báo lỗi tương ứng. Khi thành công sẽ trả về mã 1000 | OK và giá trị dưới đây.

Các giá trị trả về bao gồm:

- **id**: của bài viết
- **described**: nội dung bài viết

Mô tả get_post (2)

Các giá trị trả về bao gồm (tiếp):

- **created:** thời điểm tạo bài viết. Chú ý cần hiển thị như zalo
- **modified:** thời điểm chỉnh sửa bài viết
- **like:** số lượng người dùng đã thích bài viết
- **comment:** số lượng các comment của bài viết

Mô tả `get_post` (3)

Các giá trị trả về bao gồm (tiếp):

- **is_liked**: người dùng đã thích bài viết này chưa?
- **image**: danh sách các hình ảnh cùng id và url, id này sẽ giúp ứng dụng phân biệt được người dùng muốn xóa/thay thế ảnh nào
- **video**: video tải lên và id cùng url
- **author**: thông tin của người chủ bài viết, bao gồm cả trạng thái avatar

Mô tả `get_post` (4)

Các giá trị trả về bao gồm (tiếp):

- **state**: Hiện tại trường này có thể bỏ qua không trả về
- **is_blocked**: kiểm tra xem người chủ bài viết có chặn người dùng không?
- **can_edit**: kiểm tra xem người dùng có sửa được bài viết hay không? Không sửa được nếu không phải chủ nhân hoặc đang bị khóa bài viết
- **banned**: Hiện tại trường này có thể bỏ qua không trả về
- **can_comment**: người chủ bài viết đã khóa tính năng bình luận cho toàn bài

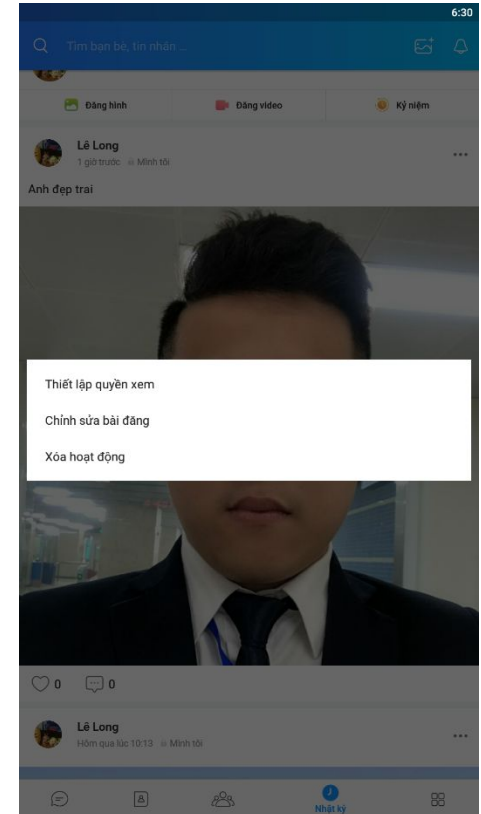
Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. **Giao diện get post**
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

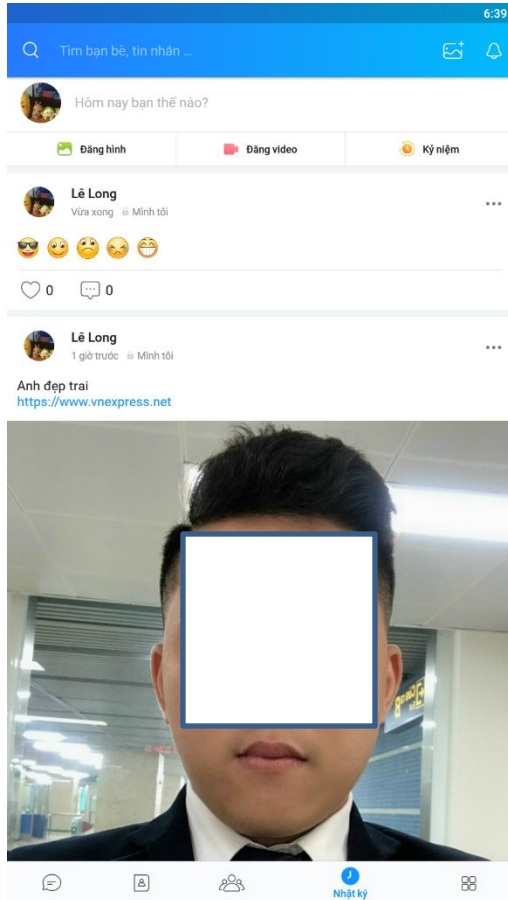
Giao diện get post (1)

Hiển thị đầy đủ thông tin bài viết.

Nếu bài viết có link liên kết thì cần phải hiển thị link và cho phép click vào sẽ được chuyển trang sang một “Webview” để đọc tin tức.



Giao diện get post (2)



Ứng dụng cần hiển thị đầy đủ cả các emoticons.

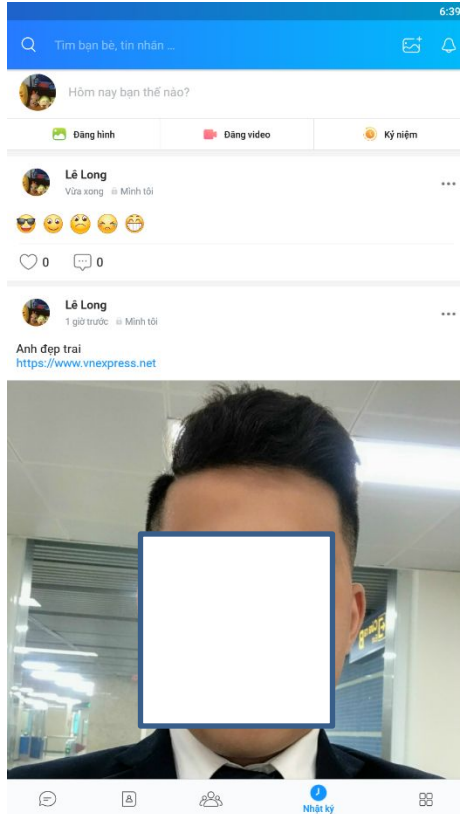
Nếu được, nên hiển thị bài viết theo chế độ căn ngang (dàn đều ra hai bên).

Còn không có thể hiển thị theo chế độ căn lề trái.

Font chữ nên là Roboto.

Nếu thời gian tạo bài viết còn gần thời điểm hiện tại thì hiển thị “Vừa xong”.

Giao diện get post (3)



Nếu thời gian đăng bài cách thời điểm hiện tại dưới 1 phút thì hiển thị “Vừa xong”.

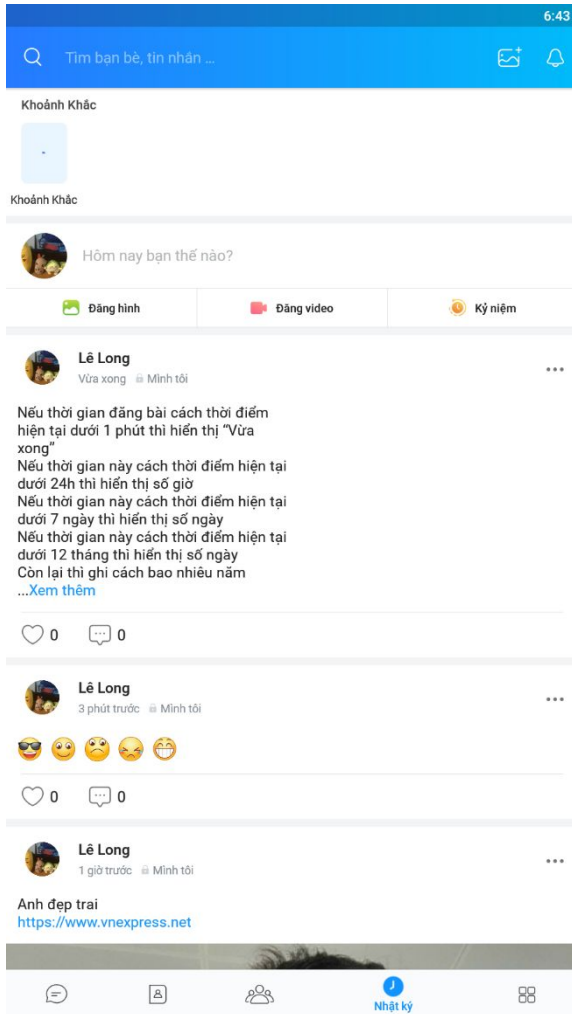
Nếu thời gian này cách thời điểm hiện tại dưới 24h thì hiển thị số giờ.

Nếu thời gian này cách thời điểm hiện tại dưới 7 ngày thì hiển thị số ngày.

Nếu thời gian này cách thời điểm hiện tại dưới 12 tháng thì hiển thị số ngày.

Còn lại thì ghi cách bao nhiêu năm.

Giao diện get post (4)



Ứng dụng cần hiển thị dấu ba chấm và nút nhấn “Xem thêm” nếu bài viết dài.

Nếu nhấn vào “Xem thêm” sẽ mở rộng bài viết với đầy đủ nội dung.

Nếu được, nên có nút “Thu gọn” sau khi đã làm mở rộng bài viết ra.

Giao diện get post (5)



Icon hình trái tim hiển thị hình màu xanh khi người dùng thích bài viết.

Hiển thị số lượng lượt thích và comment, có thể xem danh sách đầy đủ người đã thích bài viết.

Giao diện get post (6)



Khi click vào bài viết sẽ chuyển sang màn hình khác hiển thị toàn bộ bài viết, comment.

Có sự hiển thị khác nhau khi chính người chủ đã thích bài viết hoặc người chủ không thích bài viết.

Có hiển thị số lượng bình luận.

Nếu số lượng like và bình luận lớn, cần có sự phân cách (dấu chấm hoặc dấu phẩy) để người đọc phân biệt được các số hàng ngàn, hàng triệu.

Không có sự khác biệt về giao diện giữa việc chính người dùng có bình luận hay không.

Giao diện get post (7)

Ứng dụng cần hỗ trợ chèn các **emoticons** trong bài viết, không nhất thiết phải chèn ngay lập tức khi đang viết.

Nhóm nào làm được việc chuyển đổi ngay lập tức từ ký hiệu sang biểu diễn **emoticons** sẽ được cộng điểm (cần nhấn mạnh khi báo cáo BTL).

Các bức ảnh **emoticons** có thể được lưu trữ sẵn trong ứng dụng.

Khi sửa bài viết thì các **emoticons** sẽ chuyển thành dạng text.

Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. **API get list posts**
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

get_list_posts (1)

| Tên parameter | Type | NN | Mô tả |
|---------------|--------|----|----------------------------------|
| token | string | 0 | |
| last_id | string | 0 | last_id trả về lần trước |
| index | string | 0 | Index start select |
| count | string | 0 | Số bài trong một lần gửi yêu cầu |

get_list_posts (2)

| Tên parameter | Type | NN | Mô tả |
|---------------|--------|----|--|
| code | string | 0 | (tham khảo phần response common) |
| message | string | 0 | |
| data | string | 0 | |
| posts | string | 0 | |
| id | string | 0 | |
| name | string | 0 | |
| image | string | X | url hình ảnh |
| video | string | X | |
| url | string | X | nếu không có video thì bằng trống |
| thumb | string | X | nếu không có video thì bằng trống |
| described | string | 0 | |
| created | string | 0 | thời gian đăng |
| like | string | 0 | số like của sản phẩm |
| comment | string | 0 | số comment của sản phẩm |
| is_liked | string | 0 | kiểm tra User đã like chưa. 1: đã like, 0: chưa like |
| is_blocked | string | 0 | |
| 2 message | string | 0 | |

get_list_posts (3)

| | | | | |
|--|-------------|--------|---|--------------------------|
| | can_comment | string | 0 | |
| | can_edit | string | 0 | |
| | banned | string | 0 | |
| | state | string | 0 | |
| | author | string | 0 | |
| | id | string | 0 | |
| | username | string | 0 | |
| | avatar | string | 0 | |
| | online | string | X | trạng thái online |
| | new_items | string | 0 | số bài viết mới đăng |
| | last_id | string | 0 | id bài viết cuối |
| | in_campaign | string | 0 | cập nhật in_campaign |
| | campaign_id | string | 0 | cập nhật id của campaign |

Mô tả get_list_posts (1)

Ứng dụng không bao giờ yêu cầu lấy hết dữ liệu mà sẽ chỉ lấy theo từng phần.

- Mặc định từ phía ứng dụng, tất cả các yêu cầu gửi đến server có giá trị chỉ số thì count luôn bằng 20.
- Tuy vậy phía server luôn tâm niệm rằng giá trị này có thể thay đổi.

Kết quả đầu ra: nếu lấy dữ liệu thành công thì sẽ có gửi đầy đủ các thông tin cần thiết cho ứng dụng. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng.

Mô tả `get_list_posts` (2)

Dữ liệu trả về trong trường data (nếu thành công) là mảng các bài viết

- Có id của bài viết, ảnh bài viết (nếu có) hoặc video của bài viết (nếu có)
- Có name lưu trữ danh sách những người bạn có tương tác với bài viết (thích/bình luận)
- Có nội dung của bài viết trong trường `described`, thời gian đăng bài viết
- Có số lượt thích bài viết, có số bình luận của bài viết
- Có trường `can_comment` để xem bài viết đó có được bình luận hay không?

Mô tả `get_list_posts` (3)

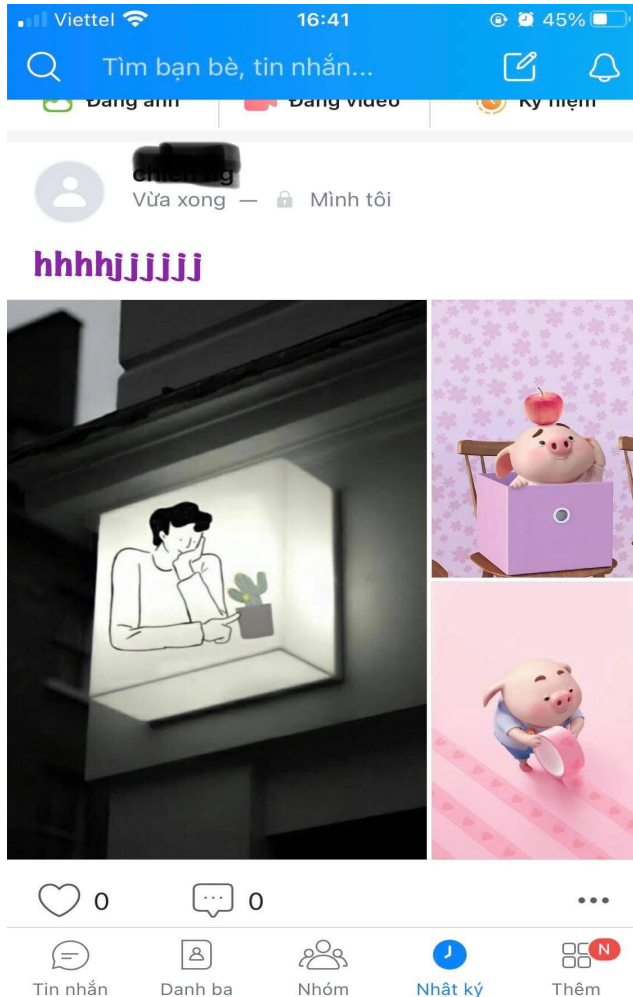
Dữ liệu trả về trong trường `data` (nếu thành công) là mảng các bài viết (tiếp)

- `banned`, `state`: Các trường này tạm thời chưa cần trả về
- `author`: trường thông tin liên quan đến tác giả bài viết, có đầy đủ `id`, `tên`, ảnh đại diện và trạng thái online
- `new_items`: số lượng các bài viết mới đăng bị bỏ lỡ
- `last_id`: cập nhật tham số để gửi lên lần sau (nếu lấy thêm bài)
- Có trường `can_edit` để bài viết đó có được chỉnh sửa hay không (nếu đang bị khóa sửa bài thì không sửa được)

Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. **Giao diện get list posts**
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

Giao diện get_list_posts



- Khi nhấp chuột vào từng bức ảnh của một bài viết trong danh sách, ứng dụng sẽ chuyển sang “Trang hiển thị bài viết” để hiển thị lại các ảnh theo danh sách dạng List View.
- Nếu mạng không có kết nối, ứng dụng có thể hiển thị lại dữ liệu trong cache.

Các testcase cho app (1)

1. Người dùng truyền đúng mã phiên đăng nhập, các tham số khác cũng đầy đủ và đúng đắn.
Kết quả mong đợi: 1000 | OK (Thông báo thành công), gửi cho ứng dụng các thông tin cần thiết.
2. Người dùng gửi sai mã phiên đăng nhập (mã bị trống hoặc quá ngắn hoặc mã phiên đăng nhập cũ).
Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập.

Các testcase cho app (2)

3. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác. Nhưng không còn bài viết nào
Kết quả mong đợi: mã lỗi báo không còn dữ liệu nữa, nhưng ứng dụng không hiển thị thông báo lỗi này, chỉ đơn giản thông báo người dùng có thể kết bạn thêm.
4. Người dùng truyền đúng mã phiên đăng nhập, id bài viết. Nhưng người dùng đã bị chặn (do hệ thống chặn).
Kết quả mong đợi: người dùng bị đăng xuất, ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập, tất cả các dữ liệu lưu trữ trước nếu có đều bị xóa.

Các testcase cho app(3)

5. Người dùng truyền đúng mã phiên đăng nhập, id bài viết. Nhưng dữ liệu trả về có trường nội dung bài viết bị lỗi. Kết quả mong đợi: ứng dụng không hiển thị những bài viết bị lỗi (sai định dạng khiến không chuyển đổi). Nếu tất cả bài mới lấy về bị lỗi thì ứng dụng hiển thị như thể không còn bài viết mới nào nữa.
6. Người dùng truyền đúng mã phiên đăng nhập, id bài viết. Nhưng dữ liệu trả về có trường like hoặc comment hoặc is_liked bị lỗi. Kết quả mong đợi: ứng dụng sẽ phải căn cứ dữ liệu trả về mà hiển thị được các thông tin cần thiết. Các trường bị lỗi coi như nhận giá trị mặc định (bằng 0).

Các testcase cho app (4)

7. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác. Nhưng dữ liệu trả về có trường can_comment thông báo bình luận bị khóa hoặc trường can_comment bị lỗi về giá trị.
Kết quả mong đợi: ứng dụng không hiển thị ô để nhập bình luận.
8. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác. Nhưng dữ liệu trả về có trường author bị lỗi về id.
Kết quả mong đợi: ứng dụng không hiển thị bài viết có thông tin tác giả bị lỗi về id. Các trường khác của author bị lỗi thì vẫn có thể nhận giá trị mặc định để hiển thị bài viết.

Các testcase cho app (5)

9. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác. Nhưng dữ liệu trả về có trường described và trường đa phương tiện (image hoặc video) bị lỗi.
Kết quả mong đợi: không hiển thị bài viết.
10. Người dùng truyền đúng mã phiên đăng nhập và các tham số khác. Nhưng dữ liệu trả về có một trong hai trường described và trường đa phương tiện (image hoặc video) bị lỗi.
Kết quả mong đợi: ứng dụng vẫn hiển thị bài viết cho các trường thông tin không bị lỗi. Các trường bị lỗi thì vẫn có thể nhận giá trị mặc định để hiển thị bài viết.

Các testcase cho app (6)

11. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác. Có bài viết cần phải che đi hình ảnh/video không thích hợp nhưng ứng dụng bị lỗi trong việc hiển thị màn che.
Kết quả mong đợi: phải bắt được sự kiện lỗi hiển thị màn che để cuối cùng là xóa đi bài viết trong danh sách.
12. Người dùng truyền đúng mã phiên đăng nhập và các tham số khác. Nhưng dữ liệu trả về có từ một trong hai trường in_campaign và campaign_id bị lỗi.
Kết quả mong đợi: ứng dụng vẫn hiển thị bài viết cho các trường thông tin không bị lỗi. Các trường bị lỗi thì vẫn có thể nhận giá trị mặc định (hoặc giá trị cũ) để tiếp tục gửi yêu cầu.

Các testcase cho app (7)

13. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác nhưng tham số last_id bị lỗi (không có hoặc không đúng định dạng).
Kết quả mong đợi: mã lỗi sai giá trị dữ liệu của tham số.
Ứng dụng không nhận được thêm danh sách nào cả.
14. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác nhưng tham số index hoặc count bị lỗi (không có hoặc không đúng định dạng).
Kết quả mong đợi: mã lỗi sai giá trị dữ liệu của tham số.
Ứng dụng không nhận được thêm danh sách nào cả.

Các testcase cho app (8)

15. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác nhưng các tham số tọa độ bị lỗi (không có hoặc không đúng định dạng).
Kết quả mong đợi: server sử dụng lại tham số lần gần nhất.
16. Người dùng truyền đúng mã phiên đăng nhập, và các tham số khác nhưng tham số user_id của người dùng bị lỗi.
Kết quả mong đợi: ứng dụng sẽ phải đẩy người dùng sang trang đăng nhập.

Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. **API check new item**
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

Check_new_item

Input

| NO | Tên parameter | Type | NN | Mô tả |
|----|---------------|--------|----|-------|
| | last_id | string | 0 | |
| | category_id | string | X | |

Output

| NO | Tên parameter | Type | NN | Mô tả |
|----|---------------|--------|----|----------------------------------|
| 1 | code | string | 0 | (tham khảo phần response common) |
| 2 | message | string | 0 | |
| 3 | data | string | 0 | |
| | new_items | string | 0 | |

Mô tả check_new_item

API thực hiện việc kiểm tra xem có phần tử mới không?

Request dạng **POST**.

Tham số: **last_id** và **category_id** (không bắt buộc, có giá trị ban đầu bằng 0).

Kết quả đầu ra: nếu thành công thì sẽ được nhận số phần tử mới chưa được hiển thị. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng.

Trường category_id có giá trị từ 0 đến 3. Nếu không truyền thì mặc định sẽ là 0.

Các testcase check_new_item (1)

1. Người dùng truyền đúng mã last_id, category_id đúng định dạng.

Kết quả mong đợi: 1000 | OK (Thông báo thành công), gửi cho ứng dụng số lượng phần tử mới của tab. Ứng dụng tự hiển thị một nút nổi ngay trên phần đầu tiên của danh sách bài viết.

2. Chú ý không hiển thị cho người dùng nhưng vẫn hiển thị trong giai đoạn phát triển.

Kết quả mong đợi: ứng dụng sẽ kiểm tra trước để không gửi dữ liệu không hợp lệ lên. Tuy vậy nếu lỡ gửi lên thì khi hệ thống trả về lỗi, ứng dụng sẽ không cần hiển thị thông báo cho người dùng bình thường biết.

Các testcase check_new_item (2)

3. Người dùng truyền các tham số hợp lệ nhưng giá trị new_items trả về không hợp lệ.
Kết quả mong đợi: Ứng dụng coi như giá trị new_items bằng 0.
4. Người dùng truyền các dữ liệu tham số hợp lệ và new_items trả về hợp lệ nhưng người dùng không còn ở trong giao diện trang chủ nữa.
Kết quả mong đợi: Ứng dụng lưu new_items vào trong cache, nhưng kể cả khi quay lại màn hình trang chủ, thì vẫn gửi yêu cầu check_new_item.

Các testcase check_new_item (3)

5. Người dùng truyền các dữ liệu tham số hợp lệ và new_items trả về hợp lệ nhưng người dùng đã bị đẩy sang trang đăng nhập.
Kết quả mong đợi: Ứng dụng bỏ qua không xử lý dữ liệu được trả về này.
6. Ứng dụng đã truyền đủ các dữ liệu tham số hợp lệ nhưng server bị lỗi không lấy được giá trị new_items này.
Kết quả mong đợi: mã lỗi trả về, ứng dụng vẫn ở nguyên màn hình trang chủ.

Các testcase check_new_item (3)

7. Ứng dụng đã truyền đủ các dữ liệu tham số hợp lệ nhưng trong quá trình kiểm tra thì mạng bị ngắt kết nối.
Kết quả mong đợi: ứng dụng cần kiểm tra ngay phía mình để đảm bảo đưa ra thông báo về kết nối mạng bị ngắt.

Mục lục

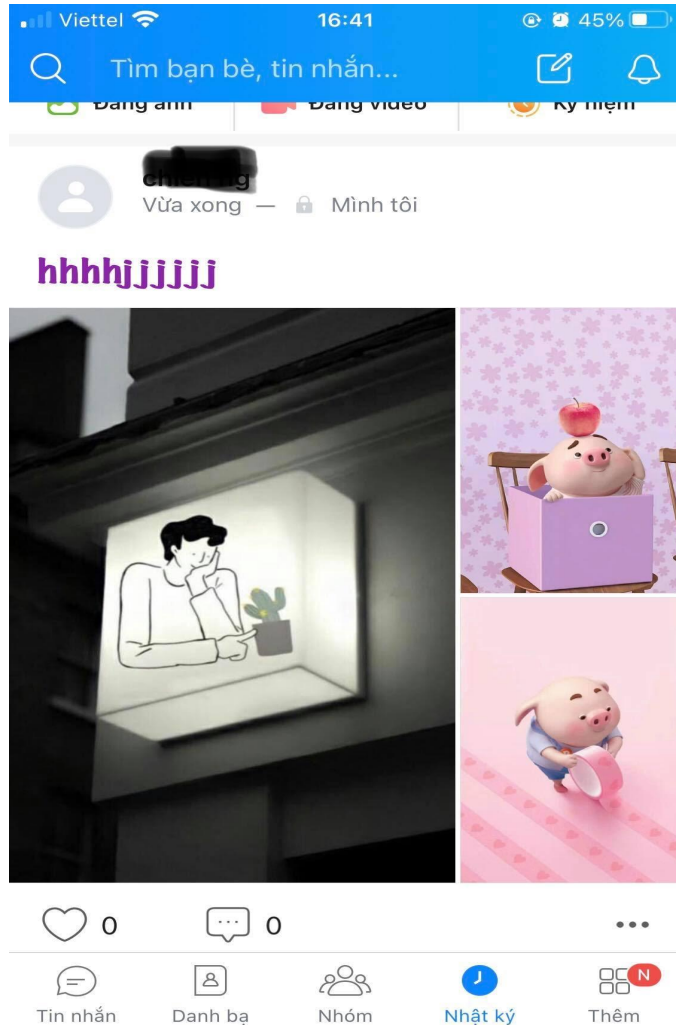
1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. **Giao diện check new item**
9. API edit post
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

Giao diện check_new_item

Để tạo trải nghiệm cho người dùng rằng ứng dụng nạp dữ liệu rất nhanh cho màn trang chủ:

- Khi vào một tab nào của màn trang chủ, nếu đã có cache thì không gửi yêu cầu get_list_post (hoặc các api khác của các tab). Ngoại trừ tab liên quan đến push và tab cài đặt.
- Nếu đã có cache thì gửi yêu cầu check_new_item để kiểm tra các phần tử mới chưa được hiện trong tab.
- Nếu một lý do cache bị lỗi không chuyển đổi từ JSON sang đối tượng được thì xóa đi và gửi yêu cầu nạp dữ liệu
- Có lẽ không nên dùng CSDL để lưu cache này.

Giao diện check_new_item



- Với cache, người dùng sẽ thấy ứng dụng hiển thị được gần như ngay lập tức dữ liệu bài viết.
- Cache giúp ứng dụng vẫn hiển thị được dữ liệu ngay khi không có mạng.
- Khi người dùng có thao tác thích hoặc bỏ thích một bài nào đó, cũng sẽ cập nhật ngay vào cache (nếu server hồi đáp thành công).

Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. **API edit post**
10. Giao diện edit post
11. API delete post
12. Giao diện delete post

Edit post

Input

| Tên parameter | Type | NN | Mô tả |
|---------------|---------------|----|---|
| token | string | 0 | |
| id | string | 0 | id bài |
| described | string | X | |
| image | array of file | X | danh sách các file ảnh mới sẽ upload |
| image_del | string | X | danh sách id của ảnh sẽ xóa |
| image_sort | string | X | thứ tự index vị trí của ảnh sẽ upload tính từ 0 |
| video | file | X | file video, hoặc url video |

Output

| NO | Tên parameter | Type | NN | Mô tả |
|----|---------------|--------|----|----------------------------------|
| 1 | code | string | 0 | (tham khảo phần response common) |
| 2 | message | string | 0 | |

Mô tả chức năng edit post

Chức năng cho phép sửa bài viết do người dùng đăng tải. Tuy nhiên chỉ có thể sửa nội dung của bài đăng, ảnh hoặc video không sửa được (mà chỉ thay thế được).

Giải thích các tham số:

image_del (danh sách id của ảnh sẽ xóa): cần nói rõ cho server biết rằng người dùng đã xóa những ảnh nào?

image_sort (thứ tự index vị trí của ảnh sẽ upload từ vị trí 0): cần nói rõ cho server biết là các ảnh đã thay thuộc vị trí nào trong danh sách các ảnh đã upload

Mô tả chức năng edit post

Kết quả: nếu đăng thành công thì sẽ hiển thị bài viết đã sửa tại vị trí cũ. Đồng thời hiện lên thông báo sửa bài viết thành công

Các test case cho Edit post(1)

1. Người dùng nhập file đúng định dạng, nội dung không trống và dưới số lượng từ cực đại.
Kết quả mong đợi: Đăng bài thành công.
2. Người dùng nhập các dữ liệu khác hợp lệ nhưng ảnh không hợp lệ do tổng dung lượng quá lớn.
Kết quả mong đợi: ứng dụng cần kiểm tra ngay phía mình để đảm bảo không gửi dữ liệu không chính xác. Thông báo cần hiển thị là dung lượng ảnh quá lớn.

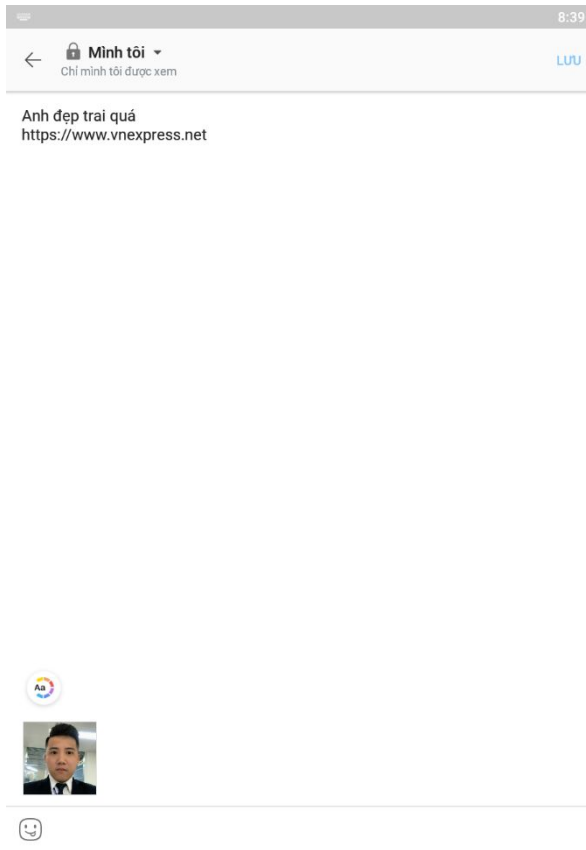
Các test case cho Edit post(2)

3. Người dùng nhập các dữ liệu hợp lệ nhưng video không hợp lệ do thời lượng hoặc dung lượng quá lớn.
Kết quả mong đợi: ứng dụng cần kiểm tra ngay phía mình để đảm bảo không gửi dữ liệu không chính xác. Thông báo cần hiển thị là thời lượng hoặc dung lượng video quá lớn.
4. Người dùng đã nhập đủ các dữ liệu khác hợp lệ nhưng trong quá trình đăng bài thì mạng bị ngắt kết nối.
Kết quả mong đợi: ứng dụng cần kiểm tra ngay phía mình để đảm bảo đưa ra thông báo về kết nối mạng bị ngắt.

Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
- 10. Giao diện edit post**
11. API delete post
12. Giao diện delete post

Giao diện Edit post



Chỉ có một chế độ hiển thị công khai

Không sửa được bài viết đăng dưới dạng video.

Chỉ thay đổi được nội dung dạng text của bài viết.

Các link liên kết được chuyển về dưới dạng văn bản.

Khi ấn nút back thì trả về giao diện cũ ngay mà không yêu cầu xác nhận.

Khi có sự thay đổi bài đăng thì màu chữ “Lưu” mới sáng lên.

Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. **API delete post**
12. Giao diện delete post

delete_post

Input

| NO | Tên parameter | Type | NN | Mô tả |
|----|---------------|--------|----|-------------|
| 1 | token | string | 0 | |
| 2 | id | string | 0 | id bài viết |

Output

| NO | Tên parameter | Type | NN | Mô tả |
|----|---------------|--------|----|----------------------------------|
| 1 | code | string | 0 | (tham khảo phần response common) |
| 2 | message | string | 0 | |

Mô tả delete_post

API thực hiện việc xóa một bài viết của một tài khoản của người dùng

Request dạng **POST**

Tham số: **token, id** (của bài viết)

Kết quả đầu ra: Nếu thành công thì mã thông báo thành công được trả về, bài viết bị xóa. Nếu không thành công thì sẽ có các thông báo lỗi tương ứng.

Chức năng cho phép xóa bài viết của một tài khoản của người dùng.

Kết quả: nếu thành công sẽ bài viết sẽ biến mất khỏi giao diện. Dữ liệu về bài viết cũng được xóa khỏi database.

Các test case cho delete_post

1. Người dùng chọn xóa bài viết.
Kết quả mong đợi: ứng dụng xóa dữ liệu bài viết và thông báo xóa bài viết thành công.
2. Người dùng chọn xóa bài viết, nhưng hệ thống không thể xóa bài viết đi được do lỗi database.
Kết quả mong đợi: Hệ thống không thể xóa được do lỗi DB. Nhưng ở phía ứng dụng, cần hiện thông báo khác, chẳng hạn “Không thể kết nối Internet”.

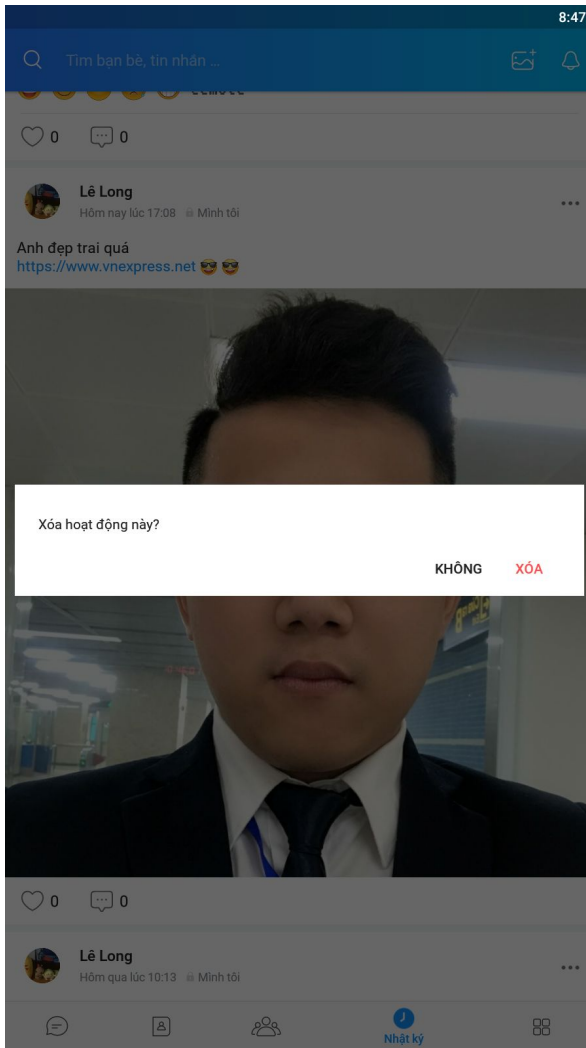
Các test case cho delete_post

3. Người dùng chọn xóa bài viết. Nhưng đang kết nối thì mạng Internet bị ngắt.
Kết quả mong đợi: ứng dụng cần hiện thông báo “Không thể kết nối Internet” càng sớm càng tốt.

Mục lục

1. API add post
2. Giao diện add post
3. API get post
4. Giao diện get post
5. API get list posts
6. Giao diện get list posts
7. API check new item
8. Giao diện check new item
9. API edit post
10. Giao diện edit post
11. API delete post
12. **Giao diện delete post**

Giao diện delete_post



Giao diện này được xuất hiện sau khi người dùng chọn lấy mục xóa hoạt động trong danh sách các chức năng nâng cao (ở góc trên tay phải mỗi bài viết)

Khi được chọn, chức năng này yêu cầu người dùng phải khẳng định một lần nữa để đảm bảo chắc chắn.

HẾT TUẦN 2