DB Practical Work 4:
The Notification model

January 8, 2018

Abstract

The following subject aims at implementing the data handling for notifications in a twitter-like web-application. Implementations are to be done in the file `model_student/notification.php`

Contents

1 Requirement

To fulfill this work, you will need the following elements:

- A working environment with db connection to both app and test databases (see `0setup.pdf`).
- At least the tables modeling user related tables and **posts**, **mentions**, **likes** and **hashtags** in `sql/schemas.sql`.

2 Work to do

You have to fill out the functions defined in the file `model_student/notification.php`

These functions are used in the application to get access to the database. Therefore, these functions must observe some rules about both input data (the formats of the parameters of the functions) and output data (the returned values).

In the functions, you can access to the PDO object by using the following instruction:

```
$db = \Db::dbc();
```

Then, you can perform queries using `$db` like a PDO object:

```
$db = \Db::dbc();
$result = $db->query('SELECT * FROM mention');
```

When you completed all the functions, you can check them by using the available unit tests.

3 The notifications entities

3.1 Presentation

Notifications are useful to inform users they have either a new post mentioning them, someone liking one of their posts or someone following them.

For each type of notification, there should be a way to store two temporal pieces of information :

- A notification datetime, which is the moment when the notification has been created
- A read datetime, which is the moment when the notification a user saw the notification.

Every datetime fields must be stored with `\DateTime` objects. The second date field (the read datetime) must be `null` if the notification hasn't been read yet.

3.2 Handling like notification

3.2.1 `get_liked_notifications($uid)`

`get_liked_notifications` returns the "liked" notifications (i.e. when someone liked one of the user's posts). It should return an array of objects which form is as following :

```
(object) array(
       "type" => "liked",
       "post" => $post_object,
       "liked_by" => $user_object,
       "date" => $liked_date,
       "reading_date" => $reading_date
)
```

3.2.2 `liked_notification_seen($pid, $uid)`

`liked_notification_seen` updates a "liked" notification as seen.

3.3 Handling mention notification

3.3.1 `get_mentioned_notifications($uid)`

`get_mentioned_notifications` returns the "mentioned" notifications (i.e. when a post mentions the user). It should return an array of objects which form is as following :

```
(object) array(
    "type" => "mentioned",
    "post" => $post_object,
    "mentioned_by" => $author,
    "date" => $post_date,
    "reading_date" => $reading_date
)
```

3.3.2 `mentioned_notification_seen($pid, $uid)`

`mentioned_notification_seen` updates a "mentioned" notification as seen.

3.4 Handling follow notification

3.4.1 `get_followed_notifications($uid)`

`get_mentioned_notifications` returns the "followed" notifications (i.e. when someone is following the user). It should return an array of objects which form is as following :

```
(object) array(
    "type" => "followed",
    "user" => \Model\User\get(1),
    "date" => $following_date,
    "reading_date" => $reading_date
)
```

3.4.2 `followed_notification_seen($followed_id, $follower_id)`

`followed_notification_seen` updates a "followed" notification as seen.