PROJECT 1: BACKEND DOCUMENTATION

CSE 331
Fall 2017
Professor: Eric P. S. Baumer {ericpsb@lehigh.edu}

PHOTO STORAGE

Each photo is stored with four pieces of data.

`id`

This is a unique positive integer assigned to each photo.

`description`

The textual description of the photo.

`src`

The (relative) path to the source file of the photo on the server.

`tn_src`

The (relative) path to a smaller (100 pixel wide) thumbnail of the photo on the server.

BACKEND STRUCTURE

Photos are accessed via a backend that consists of five end points. These endpoints correspond roughly with the main functionality your interface should provide.

UPLOADING

`/uploadPhoto.php`

Takes a photo and description from a web form and uploads them to the site. Requires a form with the following fields:

`fupload`

The photo to be uploaded. This is selected using a form field of type `file`.

`description`

Text entry with a description for the photo.

Return:

This endpoint returns a JSON with the `id` of the photo that was just uploaded. The key is `LAST_INSERT_ID()` and the value is the `id`.

VIEWING

Two endpoints are provided for viewing photos, one for viewing all thumbnails, and one for viewing the details of a photo.

`/getPhotos.php`

Returns a JSON with the `id` and `tn_src` for each photo currently uploaded to the site.

`/viewPhoto.php`

Returns a JSON with the `description` and `src` for a single photo. Requires the following parameter:

`id`

The `id` of the photo to be viewed.

UPDATING

One endpoint is used to change the description of an existing photo.

`/updatePhoto.php`

This endpoint changes the description for a photo. It requires two parameters:

`id`

The id of the photo whose description should be changed.

`description`

The text to which the photo’s description should be set.

Returns a JSON with a textual message indicating whether the photo was updated successfully or there was a problem.

DELETING

One endpoint is used to delete a photo.

`/deletePhoto.php`

This endpoint deletes a photo. It requires one parameter:

`id`

The `id` of the photo to be deleted.

Returns a JSON with a textual message indicating whether the photo was deleted successfully or there was a problem.