User photo

This page describes some design decisions behind user photo.

Requirements:

- In a buddy request message, the user photo should be shown
- In all messages from buddies the user photo should be shown

Questions/answers:

- Q: In what format should the photo be uploaded?
- A: The default is multipart/form-data uploads, so let's implement this. We expect that the app will resize the photo to prevent large uploads. We at least use a separate upload/download endpoint and do not include it in the user JSON, because it is a larger request so good to send/receive separately.
- Q: Should it be possible to upload a photo before creating the account (so before the user id is known)? Or only after the account has been submitted?
- A: Not necessary. Otherwise it is an easy target for DOS attacks.

There are multiple possible flavors for the REST API between which we have to choose. These are explained below.

Option 1 - Attached to user

The path for the photo link is a subresource of the user.

Path	Method	Description	Password required?
/users/{userID}/photo	PUT	Upload a new photo	✅
/users/{userID}/photo	DELETE	Remove the current photo	✅
/users/{userID}/photo	GET	Download your own photo	✅
/users/{userID}/buddies/{buddyID}/photo	GET	Download the photo of a buddy	✅

Pro/con:

- ➖ Cannot be used until the user created their account.
- ➕ Makes the one-to-one relation of user to photo easy to maintain
- ➖ Fetching a photo requires authentication, so standard HTTP proxy caching cannot be used
- ➖ The photo is not accessible from a buddy request message/user delete message/buddy disconnect message
- ➕ User photo can be removed immediately when updated/deleted

Option 2 - Detached

Path	Method	Description	Password required?
/userPhotos/	POST	Upload a new photo ⚠️ Can be accessed unauthorized, so DOS prevention should be enabled	❌
/userPhotos/{userPhotoId}	GET	Download a photo by ID	❌
/userPhotos/{userPhotoId}	DELETE	Remove a photo by ID ❓ How to authorize for this action? Send the password header and retrieve the attached user ID from the UserPhoto entity?	❓
EXISTING /users/	POST	Allow to link a photo on user account creation by URL in the _links section	✅
EXISTING /users/{userID}	PUT	Allow to link/unlink a photo on user update by URL in the _links section	✅

Pro/con:

- ➖ Authorization gets more complicated
- ➖ DOS prevention required
- ➕ Can be used before user account creation has finished
- ➕ Fetching a photo does not require authentication, so standard HTTP proxy caching can be used

- ⛔ User photo can not be cleaned up immediately when updated/deleted, because the app may have cached messages which have the link to the photo before update/delete
- ➕ User photo can still be accessed for some time from user deleted messages/buddy disconnect messages

Option 3 - Attached to user, separately retrievable BEST OPTION

Path	Method	Description	Password required?
/users/{userID}/photo	PUT	Upload a new photo	✅
/users/{userID}/photo	DELETE	Remove the current photo	✅
/userPhotos/{userPhotoId}	GET	Download a photo.	❌
EXISTING /users/{userID}	GET	Fetch the photo link when fetching the user (whether buddy or own user)	✅

Pro/con:

- ⛔ Cannot be used until the user created their account
- ➕ Makes the one-to-one relation of user to photo easy to maintain
- ➕ Fetching a photo does not require authentication, so standard HTTP proxy caching can be used
- ➕ No need for DOS-prevention mechanism
- ⛔ User photo can not be cleaned up immediately when updated/deleted, because the app may have cached messages which have the link to the photo before update/delete
- ➕ User photo can still be accessed for some time from user deleted messages/buddy disconnect messages