
Nestegg Documentation

Release 1.0

Casey Weed, Dax Lloyd, William McMullen, Christopher Wilson

April 29, 2015

CONTENTS

1	public blueprint	3
2	users blueprint	5
3	Indices and tables	9
	HTTP Routing Table	11

Contents:

PUBLIC BLUEPRINT

GET **/privacy**

Return static Privacy Policy.

GET **/about**

Return static About page.

GET **/tos**

Return static Terms of Service page.

GET **/**

Return home page, as well as a single random photo to be used as the background image.

USERS BLUEPRINT

GET `/user/edit-profile/security`

Edit password for user.

Form Parameters

- **current_password** – current password for user
- **password** – new password for user
- **confirm** – new password confirmation

POST `/user/edit-profile/security`

Edit password for user.

Form Parameters

- **current_password** – current password for user
- **password** – new password for user
- **confirm** – new password confirmation

GET `/user/edit-profile/about`

Edit basic information for user profile.

Form Parameters

- **first_name** – first name of user
- **last_name** – last name of user

POST `/user/edit-profile/about`

Edit basic information for user profile.

Form Parameters

- **first_name** – first name of user
- **last_name** – last name of user

GET `/user/control-panel`

Display basic control panel for user.

POST `/user/control-panel`

Display basic control panel for user.

GET `/user/unsubscribe`

Unsubscribe current user using stripe ID from database.

Parameters

- **stripe_token** (*str*) – stripe ID

GET /user/register

Register user based on form parameters.

Form Parameters

- **username** – username
- **password** – password
- **email** – email address for user

POST /user/register

Register user based on form parameters.

Form Parameters

- **username** – username
- **password** – password
- **email** – email address for user

GET /user/gallery/

Personal gallery for user wherein they can delete and edit photos.

Parameters

- **page** (*int*) – page number

GET /user/upload

Upload file given information from form.

Form Parameters

- **photo** – file to be uploaded
- **title** – title
- **desc** – description

POST /user/upload

Upload file given information from form.

Form Parameters

- **photo** – file to be uploaded
- **title** – title
- **desc** – description

GET /user/logout

Logout the current user.

POST /user/search

Initiate search based on whether or not search form parameters are validated.

GET /user/login

Login user using credentials from form.

Form Parameters

- **username** – username
- **password** – password

POST /user/login

Login user using credentials from form.

Form Parameters

- **username** – username
- **password** – password

POST /user/hook/

Receive webhooks from Stripe for subscription events.

GET /user/pro

Allow user to manage their subscription. This includes both signing up for a subscription and removing it.

POST /user/pro

Allow user to manage their subscription. This includes both signing up for a subscription and removing it.

GET /user/search_results/ (*query*) /

int: *page* Return paginated search results.

Parameters

- **query** (*str*) – search terms
- **page** (*int*) – page number

GET /user/ (*username*) /gallery/

int: *page* Public paginated gallery for user.

Parameters

- **username** (*str*) – username
- **page** (*int*) – page number

GET /user/search_results/ (*query*)

Return paginated search results.

Parameters

- **query** (*str*) – search terms
- **page** (*int*) – page number

GET /user/remove-photo/ (**int:** *image_id*)

Delete image ID if image belongs to current user.

Parameters

- **image_id** (*int*) – image ID

GET /user/view-profile/ (*username*)

Present public profile of user.

Parameters

- **username** (*str*) – username

GET /user/edit-photo/ (**int:** *image_id*)

Edit image properties, including title and description.

Parameters

- **image_id** (*int*) – image ID

POST /user/edit-photo/ (**int:** *image_id*)

Edit image properties, including title and description.

Parameters

- **image_id** (*int*) – image ID

GET /user/gallery/ (**int:** *page*)

Personal gallery for user wherein they can delete and edit photos.

Parameters

- **page** (*int*) – page number

GET /user/photo/ (**int:** *photo_id*)

View the specified photo.

Parameters

- **photo_id** (*int*) – photo ID

GET /user/ (*username*) /gallery

Public paginated gallery for user.

Parameters

- **username** (*str*) – username
- **page** (*int*) – page number

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

HTTP ROUTING TABLE

```

/
GET /,3
POST /user/register,6
POST /user/search,6
POST /user/upload,6

/about
GET /about,3

/privacy
GET /privacy,3

/tos
GET /tos,3

/user
GET /user/(username)/gallery,8
GET /user/(username)/gallery/(int:page),
7
GET /user/control-panel,5
GET /user/edit-photo/(int:image_id),7
GET /user/edit-profile/about,5
GET /user/edit-profile/security,5
GET /user/gallery/,6
GET /user/gallery/(int:page),8
GET /user/login,6
GET /user/logout,6
GET /user/photo/(int:photo_id),8
GET /user/pro,7
GET /user/register,5
GET /user/remove-photo/(int:image_id),
7
GET /user/search_results/(query),7
GET /user/search_results/(query)/(int:page),
7
GET /user/unsubscribe,5
GET /user/upload,6
GET /user/view-profile/(username),7
POST /user/control-panel,5
POST /user/edit-photo/(int:image_id),7
POST /user/edit-profile/about,5
POST /user/edit-profile/security,5
POST /user/hook/,7
POST /user/login,6
POST /user/pro,7

```