
BeaverManager Documentation

Release 0.4

James Seden Smith

February 09, 2016

1	beaver_manager package	3
1.1	Submodules	3
1.2	beaver_manager.decorators module	3
1.3	beaver_manager.email module	3
1.4	beaver_manager.forms module	3
1.5	beaver_manager.models module	4
1.6	beaver_manager.views module	8
1.7	beaver_manager.logic module	12
2	config module	15
3	API Requests	17
4	Indices and tables	23
	Python Module Index	25
	HTTP Routing Table	27

Contents:

beaver_manager package

Beaver Management app using flask

`beaver_manager.app`
Flask – Flask application instance

`beaver_manager.db`
SQLAlchemy – SQLAlchemy database instance

1.1 Submodules

1.2 beaver_manager.decorators module

`beaver_manager.decorators.async(f)`
 Decorator that will make any function asynchronous

1.3 beaver_manager.email module

Module to control email

`beaver_manager.email.send_email(subject, sender, recipients, text_body, html_body)`
 Sends email to recipient using the flask email module

Args: subject (str): Plain text sender (str): Sender of email in plain text recipients (str): Recipient email. Can be list of strings text_body (str): Body of message without HTML formatting html_body (str): Body of message with HTML formatting

1.4 beaver_manager.forms module

```
class beaver_manager.forms.BeaverAttendanceForm(formdata=<class
    'flask_wtf.form._Auto'>,      obj=None,
    prefix='',      csrf_context=None,      se-
    cret_key=None,      csrf_enabled=None,
    *args, **kwargs)
```

Bases: flask_wtf.form.Form

`beavers = <UnboundField(MultiCheckboxField, ('Present: ',), {'coerce': <class 'int'>})>`

```
class beaver_manager.forms.BeaverSortForm(formdata=<class 'flask_wtf.form._Auto'>,
                                           obj=None, prefix='', csrf_context=None, se-
                                           cret_key=None, csrf_enabled=None, *args,
                                           **kwargs)

    Bases: flask_wtf.form.Form

    sort_on = <UnboundField(SelectField, ('Sort By'), {'choices': [(('surname', 'Surname'), ('first_name', 'First Name'), ('l
class beaver_manager.forms.EmailForm(formdata=<class 'flask_wtf.form._Auto'>, obj=None,
                                      prefix='', csrf_context=None, secret_key=None,
                                      csrf_enabled=None, *args, **kwargs)

    Bases: flask_wtf.form.Form

    message = <UnboundField(TextAreaField, (), {})>
    subject = <UnboundField(StringField, (), {})>

class beaver_manager.forms.MultiCheckboxField(label=None, validators=None, co-
                                              erce=<class 'str'>, choices=None,
                                              **kwargs)

    Bases: wtforms.fields.core.SelectMultipleField

    A multiple-select, except displays a list of checkboxes.

    Iterating the field will produce subfields, allowing custom rendering of the enclosed checkbox fields.

    option_widget = <wtforms.widgets.core.CheckboxInput object>
    widget = <wtforms.widgets.core.ListWidget object>

class beaver_manager.forms.TripSortForm(formdata=<class 'flask_wtf.form._Auto'>, obj=None,
                                         prefix='', csrf_context=None, secret_key=None,
                                         csrf_enabled=None, *args, **kwargs)

    Bases: flask_wtf.form.Form

    sort_on = <UnboundField(SelectField, ('Sort By'), {'choices': [(('date', 'Date'), ('location', 'Location'))]])>
```

1.5 beaver_manager.models module

This module contains SQLAlchemy model definitions to create SQL tables

```
class beaver_manager.models.Attendance(**kwargs)
    Bases: flask_sqlalchemy.Model

    A model for a meeting. Linked to a Criterion to allow automatic completion of badges. If beaver was
    present for the meeting then the criterion is completed.

    id
        int – Unique Primary Key.

    date
        DateTime – Date of meeting

    criterion_id
        int – Foreign key for Criterion

    criterion
        Criterion – Provides a link to the Criterion that Attendance is associated with

    description
        str – Brief description of evening
```

organiser

str – Name of organiser for that meeting

class beaver_manager.models.**Badge** (**kwargs)

Bases: flask_sqlalchemy.Model

Model for a Badge

id

int – Unique Primary Key.

name

str – Name of the badge

img_url

str – URL for the badge image

beavers_badges

list[[BeaverBadge](#)] – A list of beavers_badges associated with the lodge

class beaver_manager.models.**BadgeCriterion** (criterion_id, badge_id, completed)

Bases: flask_sqlalchemy.Model

Model for a BadgeCriterion. Acts as an intermediary between [BeaverBadge](#) and [Criterion](#)

Parameters

- **criterion_id** (*int*) – Foreign key for [Criterion](#)
- **badge_id** (*int*) – Foreign key for [BeaverBadge](#)
- **completed** (*bool*) – Represents whether the criterion has been completed or not

id

int – Unique Primary Key.

criterion

[Criterion](#) – Provides a link to the [Criterion](#) that criterion is associated with

badge

[BeaverBadge](#) – Provides a link to the [BeaverBadge](#) that criterion is associated with

description

str – Description of criterion

class beaver_manager.models.**Beaver** (**kwargs)

Bases: flask_sqlalchemy.Model

Model for a beaver

id

int – Unique Primary Key.

first_name

str – The beaver's first name

surname

str – The beaver's surname

dob

DateTime – The beaver's date of birth

lodge_id

int – Foreign key for [Lodge](#) table

lodge

Lodge – Provides direct access to the *Lodge* that is that is linked to the beaver

contacts

list[*EmergencyContact*] – A list of *EmergencyContact* objects associated with the beaver

badges

list[*BeaverBadge*] – A list of *BeaverBadge* associated with the beaver

beaver_attendances

list[*BeaverAttendance*] – A list of *BeaverAttendance* objects associated with the beaver

class beaver_manager.models.**BeaverAttendance** (*attendance_id*, *beaver_id*, *present*)

Bases: flask_sqlalchemy.Model

Model for a BeaverAttendance. Acts as an intermediary between *Beaver* and *Attendance*

Parameters

- **attendance_id** (*int*) – Foreign key for *Attendance*
- **beaver_id** (*int*) – Foreign key for *Beaver*
- **present** (*bool*) – Represents whether the beaver was been presernt or not

id

int – Unique Primary Key.

attendance

Attendance – Provides a link to the *Attendance* that BeaverAttendance is associated with

beaver

Beaver – Provides a link to the *Beaver* that BeaverAttendance is associated with

class beaver_manager.models.**BeaverBadge** (*beaver_id*, *badge_id*, *completed*)

Bases: flask_sqlalchemy.Model

Model for a BeaverBadge. Acts as an intermediary between *Beaver* and *Badge*

Parameters

- **beaver_id** (*int*) – Foreign key for *Beaver*
- **badge_id** (*int*) – Foreign key for *Badge*
- **completed** (*bool*) – Represents whether the badge has been completed or not

id

int – Unique Primary Key.

beaver

Beaver – Provides a link to the *Beaver* that badge is associated with

class beaver_manager.models.**BeaverTrip** (*beaver_id*, *trip_id*, *permission*, *paid*)

Bases: flask_sqlalchemy.Model

Model for a BeaverTrip. Acts as intermediary between *Beaver* and *Trip*

Parameters

- **beaver_id** (*int*) – Foreign key for *Beaver*
- **trip_id** (*int*) – Foreign key for *Trip*
- **permission** (*bool*) – Represents whether the criterion has been completed or not
- **paid** (*bool*) – Represents whether the criterion has been completed or not

id

int – Unique Primary Key.

beaver

Beaver – Provides a link to the *Beaver* that BeaverTrip is associated with

trip

Trip – Provides a link to the *Trip* that BeaverTrip is associated with

class beaver_manager.models.**Criterion** (***kwargs*)

Bases: flask_sqlalchemy.Model

Model for a Criterion.

id

int – Unique Primary Key.

badge_id

int – Foreign key for *Badge*

badge

Badge – Provides a link to the *Badge* that criterion is associated with

description

str – Description of criterion

class beaver_manager.models.**EmergencyContact** (***kwargs*)

Bases: flask_sqlalchemy.Model

Model for an Emergency contact

id

int – Unique Primary Key.

beaver_id

int – Foreign Key for the *Beaver* table

beaver

Beaver – Provides direct access to the *Beaver* that is

linked to the contact

first_name

str – The contact's first name

surname

str – The contact's surname

email

str – The contact's email

phone_number

str – The contact's phone number

address_line1

str – The contact's first address line

address_line2

str – The contact's first second address line

town

str – The town in which the Contact lives

county

str – The county in which the Contact lives

postcode*str* – The contact’s postcode**class** beaver_manager.models.Lodge (**kwargs)

Bases: flask_sqlalchemy.Model

Model for a Lodge (group of beavers)

id*int* – Unique Primary Key.**name***str* – Name of the lodge**beavers**

list[Beaver] – A list of beavers associated with the lodge

class beaver_manager.models.Trip (**kwargs)

Bases: flask_sqlalchemy.Model

A model for a Trip.

id*int* – Unique Primary Key.**location***str* – Place Name**cost***numeric* – Cost of trip [Max Value of 99999.99]**date***DateTime* – Date of trip**overnight***bool* – Whether trip is overnight**number_of_nights***int* – Number of nights the trip is

1.6 beaver_manager.views module

This module contains the views for the app.

Using flask’s `route()` function to route the request it then executes the necessary code return a rendered template at the end.

```
class beaver_manager.views.AttendanceModelView(model, session, name=None, category=None, endpoint=None, url=None, static_folder=None, menu_class_name=None, menu_icon_type=None, menu_icon_value=None)
```

Bases: flask_admin.contrib.sqla.view.ModelView

Custom view for Flask-Admin. Adds BeaverAttendance as an inline view

action_view()

Mass-model action view.

ajax_lookup()

```

ajax_update()
    Edits a single column of a record in list view.

create_view()
    Create model view

delete_view()
    Delete model view. Only POST method is allowed.

details_view()
    Details model view

edit_view()
    Edit model view

export_csv()
    Export a CSV of records.

form_excluded_columns = 'beaver_attendances'

index_view()
    List view

class beaver_manager.views.BadgeModelView(model, session, name=None, category=None,
                                           endpoint=None, url=None, static_folder=None,
                                           menu_class_name=None, menu_icon_type=None,
                                           menu_icon_value=None)
    Bases: flask_admin.contrib.sqla.view.ModelView

    Custom view for Flask-Admin. Adds criterion as inline model and modifies update behaviour

action_view()
    Mass-model action view.

ajax_lookup()

ajax_update()
    Edits a single column of a record in list view.

create_view()
    Create model view

delete_view()
    Delete model view. Only POST method is allowed.

details_view()
    Details model view

edit_view()
    Edit model view

export_csv()
    Export a CSV of records.

form_excluded_columns = 'beaver_badges'

index_view()
    List view

inline_models = (<class 'beaver_manager.models.Criterion'>,)

on_model_change(form, model, is_created)
    When a Badge record is created or modified ensures that it has a BeaverBadge record for all Beavers and
    that the BeaverBadge has the correct criteria

```

on_model_delete (*model*)

When a Badge is deleted, delete all :class:‘BeaverBadge’s associated with it.

```
class beaver_manager.views.BeaverBadgeModelView(model, session, name=None, category=None, endpoint=None, url=None, static_folder=None, menu_class_name=None, menu_icon_type=None, menu_icon_value=None)
```

Bases: flask_admin.contrib.sqla.view.ModelView

Only used for debugging. Custom view for Flask-Admin. Adds BadgeCriterion as an inline view

action_view ()

Mass-model action view.

ajax_lookup ()

ajax_update ()

Edits a single column of a record in list view.

create_view ()

Create model view

delete_view ()

Delete model view. Only POST method is allowed.

details_view ()

Details model view

edit_view ()

Edit model view

export_csv ()

Export a CSV of records.

index_view ()

List view

inline_models = (<class ‘beaver_manager.models.BadgeCriterion’>,)

```
class beaver_manager.views.BeaverModelView(model, session, name=None, category=None, endpoint=None, url=None, static_folder=None, menu_class_name=None, menu_icon_type=None, menu_icon_value=None)
```

Bases: flask_admin.contrib.sqla.view.ModelView

Custom view for Flask-Admin. Adds emergency contacts as inline model and modifies update behaviour

action_view ()

Mass-model action view.

ajax_lookup ()

ajax_update ()

Edits a single column of a record in list view.

create_view ()

Create model view

delete_view ()

Delete model view. Only POST method is allowed.

```

details_view()
    Details model view

edit_view()
    Edit model view

export_csv()
    Export a CSV of records.

form_excluded_columns = ('badges', 'beaver_attendances', 'trips')

index_view()
    List view

inline_models = (<class 'beaver_manager.models.EmergencyContact'>,)

on_model_change (form, model, is_created)
    When a Beaver record is created or modified ensures that it has a BeaverBadge record for all Badges and
    that the BeaverBadge has the correct criteria

on_model_delete (model)
    When a beaver is deleted delete all :class:'BeaverBadge's, :class:'BeaverTrip's,
    :class:'BeaverAttendance's and :class:'EmergencyContact's associated with it.

class beaver_manager.views.ChoiceObj (name, choices)
    Bases: object

class beaver_manager.views.MyHomeView (name=None, category=None, endpoint=None,
                                         url=None, template='admin/index.html',
                                         menu_class_name=None, menu_icon_type=None,
                                         menu_icon_value=None)
    Bases: flask_admin.base.AdminIndexView

index()

class beaver_manager.views.TripModelView (model, session, name=None, category=None,
                                             endpoint=None, url=None, static_folder=None,
                                             menu_class_name=None, menu_icon_type=None,
                                             menu_icon_value=None)
    Bases: flask_admin.contrib.sqla.view.ModelView

    Custom view for Flask-Admin modifying update behaviour

action_view()
    Mass-model action view.

ajax_lookup()

ajax_update()
    Edits a single column of a record in list view.

create_view()
    Create model view

delete_view()
    Delete model view. Only POST method is allowed.

details_view()
    Details model view

edit_view()
    Edit model view

```

export_csv()

Export a CSV of records.

form_excluded_columns = 'trips'

index_view()

List view

on_model_change(*form, model, is_created*)

When a Trip record is created or modified ensures that it has a BeaverTrip record for all Beavers.

on_model_delete(*model*)

When a Trip is deleted, delete all :class:'BeaverTrip's associated with it.

`beaver_manager.views.beaver`(*beaver_id*)

Displays Information about the given beaver

Parameters **beaver_id**(*int*) – The ID number of the beaver record

`beaver_manager.views.beavers`()

Queries the database for all beavers then displays a list of them

`beaver_manager.views.index`(*success=None, error=None*)

Displays the homepage

Parameters

- **success**(*str*) – Message to be displayed in toastr success pop up
- **error**(*str*) – Message to be displayed in toastr error pop up

`beaver_manager.views.internal_error`(*error*)

`beaver_manager.views.not_found_error`(*error*)

`beaver_manager.views.register`(*attendance_id*)

Displays a form to record beavers attendance.

Parameters **attendance_id**(*int*) – The ID number of the attendance record

`beaver_manager.views.registers`()

Displays a list displaying dates for which registers can be taken

`beaver_manager.views.trip`(*trip_id*)

Displays information about the trip and allows user to select whether or a beaver has paid for the trip and has permission for the trip

Parameters **trip_id**(*int*) – The ID number of the trip record

`beaver_manager.views.trips`()

Displays a list of all trips linking to an individual view showing more data

1.7 beaver_manager.logic module

This module contains any logic that does not belong in any other module

`beaver_manager.logic.to_percent`(*value, total*)

Works out the percentage of a value from a total.

Parameters

- **value**(*int*) – The value to be converted
- **total**(*int*) – The total to be used in conversion

Returns Number between 0.0 and 100.0

Return type (int)

`beaver_manager.logic.update_beaver_badge(beaver_badge)`

Checks to see if all the criteria are completed for `beaver_badge` and if so sets `beaver_badge.completed` to True

Parameters

- **beaver_badge** (`BeaverBadge`) – The beaver badge which need
- **checking** –

`beaver_manager.logic.update_criterion(attendance)`

Check whether a beaver was present for a given attendance and if so sets the corresponding `criterion.completed` to True

Parameters **attendance** (`Attendance`) – The attendance for which criterion need updating

config module

Contains the configuration options for the flask app.

`config.basedir`

str – Base directory of the flask app

`config.SQLALCHEMY_DATABASE_URI`

str – Location of SQLAlchemy database

`config.SQLALCHEMY_MIGRATE_REPO`

str – Directory containing database migrations and history

`config.WTF_CSRF_ENABLED`

bool – Whether CSRF is enabled for forms, protecting data

`config.SECRET_KEY`

str – A secret key

API Requests

GET /admin/beaverattendance/export/csv/
Export a CSV of records.

POST /admin/beaverattendance/ajax/update/
Edits a single column of a record in list view.

GET /admin/badgecriterion/export/csv/
Export a CSV of records.

POST /admin/badgecriterion/ajax/update/
Edits a single column of a record in list view.

GET /admin/beaverbadge/export/csv/
Export a CSV of records.

POST /admin/beaverbadge/ajax/update/
Edits a single column of a record in list view.

GET /admin/attendance/export/csv/
Export a CSV of records.

GET /admin/beavertrip/export/csv/
Export a CSV of records.

POST /admin/attendance/ajax/update/
Edits a single column of a record in list view.

POST /admin/beavertrip/ajax/update/
Edits a single column of a record in list view.

GET /admin/criterion/export/csv/
Export a CSV of records.

POST /admin/criterion/ajax/update/
Edits a single column of a record in list view.

GET /admin/beaver/export/csv/
Export a CSV of records.

POST /admin/beaver/ajax/update/
Edits a single column of a record in list view.

GET /admin/badge/export/csv/
Export a CSV of records.

GET /admin/lodge/export/csv/
Export a CSV of records.

POST /admin/badge/ajax/update/
Edits a single column of a record in list view.

POST /admin/lodge/ajax/update/
Edits a single column of a record in list view.

GET /admin/trip/export/csv/
Export a CSV of records.

POST /admin/trip/ajax/update/
Edits a single column of a record in list view.

GET /admin/beaverattendance/details/
Details model view

POST /admin/beaverattendance/action/
Mass-model action view.

POST /admin/beaverattendance/delete/
Delete model view. Only POST method is allowed.

POST /admin/beaverattendance/edit/
Edit model view

GET /admin/beaverattendance/edit/
Edit model view

POST /admin/beaverattendance/new/
Create model view

GET /admin/beaverattendance/new/
Create model view

GET /admin/badgecriterion/details/
Details model view

POST /admin/badgecriterion/action/
Mass-model action view.

POST /admin/badgecriterion/delete/
Delete model view. Only POST method is allowed.

POST /admin/badgecriterion/edit/
Edit model view

GET /admin/badgecriterion/edit/
Edit model view

POST /admin/badgecriterion/new/
Create model view

GET /admin/badgecriterion/new/
Create model view

GET /admin/beaverbadge/details/
Details model view

POST /admin/beaverbadge/action/
Mass-model action view.

POST /admin/beaverbadge/delete/
Delete model view. Only POST method is allowed.

POST /admin/beaverbadge/edit/
Edit model view

GET /admin/beaverbadge/edit/
Edit model view

POST /admin/beaverbadge/new/
Create model view

GET /admin/beaverbadge/new/
Create model view

GET /admin/attendance/details/
Details model view

GET /admin/beavertrip/details/
Details model view

POST /admin/attendance/action/
Mass-model action view.

POST /admin/attendance/delete/
Delete model view. Only POST method is allowed.

POST /admin/beavertrip/action/
Mass-model action view.

POST /admin/beavertrip/delete/
Delete model view. Only POST method is allowed.

POST /admin/attendance/edit/
Edit model view

GET /admin/attendance/edit/
Edit model view

POST /admin/beavertrip/edit/
Edit model view

GET /admin/beavertrip/edit/
Edit model view

POST /admin/attendance/new/
Create model view

GET /admin/attendance/new/
Create model view

POST /admin/beavertrip/new/
Create model view

GET /admin/beavertrip/new/
Create model view

GET /admin/criterion/details/
Details model view

POST /admin/criterion/action/
Mass-model action view.

POST /admin/criterion/delete/
Delete model view. Only POST method is allowed.

POST `/admin/criterion/edit/`
Edit model view

GET `/admin/criterion/edit/`
Edit model view

POST `/admin/criterion/new/`
Create model view

GET `/admin/criterion/new/`
Create model view

GET `/admin/beaver/details/`
Details model view

POST `/admin/beaver/action/`
Mass-model action view.

POST `/admin/beaver/delete/`
Delete model view. Only POST method is allowed.

POST `/admin/beaver/edit/`
Edit model view

GET `/admin/beaver/edit/`
Edit model view

POST `/admin/beaver/new/`
Create model view

GET `/admin/beaver/new/`
Create model view

GET `/admin/badge/details/`
Details model view

GET `/admin/lodge/details/`
Details model view

POST `/admin/badge/action/`
Mass-model action view.

POST `/admin/badge/delete/`
Delete model view. Only POST method is allowed.

POST `/admin/lodge/action/`
Mass-model action view.

POST `/admin/lodge/delete/`
Delete model view. Only POST method is allowed.

POST `/admin/badge/edit/`
Edit model view

GET `/admin/badge/edit/`
Edit model view

POST `/admin/lodge/edit/`
Edit model view

GET `/admin/lodge/edit/`
Edit model view

POST /admin/badge/new/
Create model view

GET /admin/badge/new/
Create model view

POST /admin/lodge/new/
Create model view

GET /admin/lodge/new/
Create model view

GET /admin/trip/details/
Details model view

POST /admin/trip/action/
Mass-model action view.

POST /admin/trip/delete/
Delete model view. Only POST method is allowed.

POST /admin/trip/edit/
Edit model view

GET /admin/trip/edit/
Edit model view

POST /admin/trip/new/
Create model view

GET /admin/trip/new/
Create model view

GET /admin/beaverattendance/
List view

GET /admin/badgecriterion/
List view

GET /admin/beaverbadge/
List view

GET /admin/attendance/
List view

GET /admin/beavertrip/
List view

GET /admin/criterion/
List view

GET /admin/beaver/
List view

GET /admin/badge/
List view

GET /admin/lodge/
List view

GET /admin/trip/
List view

GET /registers

Displays a list displaying dates for which registers can be taken

GET /beavers

Queries the database for all beavers then displays a list of them

POST /index

POST /

Displays the homepage

Args: success (str): Message to be displayed in toastr success pop up error (str): Message to be displayed in toastr error pop up

GET /index

GET /

Displays the homepage

Args: success (str): Message to be displayed in toastr success pop up error (str): Message to be displayed in toastr error pop up

GET /trips/

Displays a list of all trips linking to an individual view showing more data

GET /admin/static/ (path: filename)

Function used internally to send static files from the static folder to the browser.

New in version 0.5.

POST /registers/ (attendance_id)

Displays a form to record beavers attendance.

Args: attendance_id (int): The ID number of the attendance record

GET /registers/ (attendance_id)

Displays a form to record beavers attendance.

Args: attendance_id (int): The ID number of the attendance record

GET /beavers/ (beaver_id)

Displays Information about the given beaver

Args: beaver_id (int): The ID number of the beaver record

POST /trips/ (trip_id)

Displays information about the trip and allows user to select whether or a beaver has paid for the trip and has permission for the trip

Args: trip_id (int): The ID number of the trip record

GET /trips/ (trip_id)

Displays information about the trip and allows user to select whether or a beaver has paid for the trip and has permission for the trip

Args: trip_id (int): The ID number of the trip record

Indices and tables

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

b

`beaver_manager`, [3](#)
`beaver_manager.decorators`, [3](#)
`beaver_manager.email`, [3](#)
`beaver_manager.forms`, [3](#)
`beaver_manager.logic`, [12](#)
`beaver_manager.models`, [4](#)
`beaver_manager.views`, [8](#)

c

`config`, [15](#)

/	GET /admin/beavertrip/edit/, 19
GET /, 22	GET /admin/beavertrip/export/csv/, 17
POST /, 22	GET /admin/beavertrip/new/, 19
/admin	GET /admin/criterion/, 21
GET /admin/attendance/, 21	GET /admin/criterion/details/, 19
GET /admin/attendance/details/, 19	GET /admin/criterion/edit/, 20
GET /admin/attendance/edit/, 19	GET /admin/criterion/export/csv/, 17
GET /admin/attendance/export/csv/, 17	GET /admin/criterion/new/, 20
GET /admin/attendance/new/, 19	GET /admin/lodge/, 21
GET /admin/badge/, 21	GET /admin/lodge/details/, 20
GET /admin/badge/details/, 20	GET /admin/lodge/edit/, 20
GET /admin/badge/edit/, 20	GET /admin/lodge/export/csv/, 17
GET /admin/badge/export/csv/, 17	GET /admin/lodge/new/, 21
GET /admin/badge/new/, 21	GET /admin/static/ (path:filename), 22
GET /admin/badgecriterion/, 21	GET /admin/trip/, 21
GET /admin/badgecriterion/details/, 18	GET /admin/trip/details/, 21
GET /admin/badgecriterion/edit/, 18	GET /admin/trip/edit/, 21
GET /admin/badgecriterion/export/csv/, 17	GET /admin/trip/export/csv/, 18
GET /admin/badgecriterion/new/, 18	GET /admin/trip/new/, 21
GET /admin/beaver/, 21	POST /admin/attendance/action/, 19
GET /admin/beaver/details/, 20	POST /admin/attendance/ajax/update/, 17
GET /admin/beaver/edit/, 20	POST /admin/attendance/delete/, 19
GET /admin/beaver/export/csv/, 17	POST /admin/attendance/edit/, 19
GET /admin/beaver/new/, 20	POST /admin/attendance/new/, 19
GET /admin/beaverattendance/, 21	POST /admin/badge/action/, 20
GET /admin/beaverattendance/details/, 18	POST /admin/badge/ajax/update/, 17
GET /admin/beaverattendance/edit/, 18	POST /admin/badge/delete/, 20
GET /admin/beaverattendance/export/csv/, 17	POST /admin/badge/edit/, 20
GET /admin/beaverattendance/new/, 18	POST /admin/badge/new/, 20
GET /admin/beaverbadge/, 21	POST /admin/badgecriterion/action/, 18
GET /admin/beaverbadge/details/, 18	POST /admin/badgecriterion/ajax/update/, 17
GET /admin/beaverbadge/edit/, 19	POST /admin/badgecriterion/delete/, 18
GET /admin/beaverbadge/export/csv/, 17	POST /admin/badgecriterion/edit/, 18
GET /admin/beaverbadge/new/, 19	POST /admin/badgecriterion/new/, 18
GET /admin/beavertrip/, 21	POST /admin/beaver/action/, 20
GET /admin/beavertrip/details/, 19	POST /admin/beaver/ajax/update/, 17
	POST /admin/beaver/delete/, 20
	POST /admin/beaver/edit/, 20
	POST /admin/beaver/new/, 20

POST /admin/beaverattendance/action/,
18
POST /admin/beaverattendance/ajax/update/,
17
POST /admin/beaverattendance/delete/,
18
POST /admin/beaverattendance/edit/, 18
POST /admin/beaverattendance/new/, 18
POST /admin/beaverbadge/action/, 18
POST /admin/beaverbadge/ajax/update/,
17
POST /admin/beaverbadge/delete/, 18
POST /admin/beaverbadge/edit/, 18
POST /admin/beaverbadge/new/, 19
POST /admin/beavertrip/action/, 19
POST /admin/beavertrip/ajax/update/, 17
POST /admin/beavertrip/delete/, 19
POST /admin/beavertrip/edit/, 19
POST /admin/beavertrip/new/, 19
POST /admin/criterion/action/, 19
POST /admin/criterion/ajax/update/, 17
POST /admin/criterion/delete/, 19
POST /admin/criterion/edit/, 19
POST /admin/criterion/new/, 20
POST /admin/lodge/action/, 20
POST /admin/lodge/ajax/update/, 18
POST /admin/lodge/delete/, 20
POST /admin/lodge/edit/, 20
POST /admin/lodge/new/, 21
POST /admin/trip/action/, 21
POST /admin/trip/ajax/update/, 18
POST /admin/trip/delete/, 21
POST /admin/trip/edit/, 21
POST /admin/trip/new/, 21

/beavers

GET /beavers, 22
GET /beavers/(beaver_id), 22

/index

GET /index, 22
POST /index, 22

/registers

GET /registers, 21
GET /registers/(attendance_id), 22
POST /registers/(attendance_id), 22

/trips

GET /trips/, 22
GET /trips/(trip_id), 22
POST /trips/(trip_id), 22

A

[action_view\(\)](#) (beaver_manager.views.AttendanceModelView method), 8
[action_view\(\)](#) (beaver_manager.views.BadgeModelView method), 9
[action_view\(\)](#) (beaver_manager.views.BeaverBadgeModelView method), 10
[action_view\(\)](#) (beaver_manager.views.BeaverModelView method), 10
[action_view\(\)](#) (beaver_manager.views.TripModelView method), 11
[address_line1](#) (beaver_manager.models.EmergencyContact attribute), 7
[address_line2](#) (beaver_manager.models.EmergencyContact attribute), 7
[ajax_lookup\(\)](#) (beaver_manager.views.AttendanceModelView method), 8
[ajax_lookup\(\)](#) (beaver_manager.views.BadgeModelView method), 9
[ajax_lookup\(\)](#) (beaver_manager.views.BeaverBadgeModelView method), 10
[ajax_lookup\(\)](#) (beaver_manager.views.BeaverModelView method), 10
[ajax_lookup\(\)](#) (beaver_manager.views.TripModelView method), 11
[ajax_update\(\)](#) (beaver_manager.views.AttendanceModelView method), 8
[ajax_update\(\)](#) (beaver_manager.views.BadgeModelView method), 9
[ajax_update\(\)](#) (beaver_manager.views.BeaverBadgeModelView method), 10
[ajax_update\(\)](#) (beaver_manager.views.BeaverModelView method), 10
[ajax_update\(\)](#) (beaver_manager.views.TripModelView method), 11
[app](#) (in module beaver_manager), 3
[async\(\)](#) (in module beaver_manager.decorators), 3
[attendance](#) (beaver_manager.models.BeaverAttendance attribute), 6
[Attendance](#) (class in beaver_manager.models), 4
[AttendanceModelView](#) (class in beaver_manager.views), 8

B

[badge](#) (beaver_manager.models.BadgeCriterion attribute), 5
[badge](#) (beaver_manager.models.Criterion attribute), 7
[Badge](#) (class in beaver_manager.models), 5
[badge_id](#) (beaver_manager.models.Criterion attribute), 7
[BadgeCriterion](#) (class in beaver_manager.models), 5
[BadgeModelView](#) (class in beaver_manager.views), 9
[badges](#) (beaver_manager.models.Beaver attribute), 6
[basedir](#) (in module config), 15
[beaver](#) (beaver_manager.models.BeaverAttendance attribute), 6
[beaver](#) (beaver_manager.models.BeaverBadge attribute), 6
[beaver](#) (beaver_manager.models.BeaverTrip attribute), 7
[beaver](#) (beaver_manager.models.EmergencyContact attribute), 7
[Beaver](#) (class in beaver_manager.models), 5
[beaver\(\)](#) (in module beaver_manager.views), 12
[beaver_attendances](#) (beaver_manager.models.Beaver attribute), 6
[beaver_id](#) (beaver_manager.models.EmergencyContact attribute), 7
[beaver_manager](#) (module), 3
[beaver_manager.decorators](#) (module), 3
[beaver_manager.email](#) (module), 3
[beaver_manager.forms](#) (module), 3
[beaver_manager.logic](#) (module), 12
[beaver_manager.models](#) (module), 4
[beaver_manager.views](#) (module), 8
[BeaverAttendance](#) (class in beaver_manager.models), 6
[BeaverAttendanceForm](#) (class in beaver_manager.forms), 3
[BeaverBadge](#) (class in beaver_manager.models), 6
[BeaverBadgeModelView](#) (class in beaver_manager.views), 10
[BeaverModelView](#) (class in beaver_manager.views), 10

beavers (beaver_manager.forms.BeaverAttendanceForm attribute), 3
 beavers (beaver_manager.models.Lodge attribute), 8
 beavers() (in module beaver_manager.views), 12
 beavers_badges (beaver_manager.models.Badge attribute), 5
 BeaverSortForm (class in beaver_manager.forms), 3
 BeaverTrip (class in beaver_manager.models), 6

C

ChoiceObj (class in beaver_manager.views), 11
 config (module), 15
 contacts (beaver_manager.models.Beaver attribute), 6
 cost (beaver_manager.models.Trip attribute), 8
 county (beaver_manager.models.EmergencyContact attribute), 7
 create_view() (beaver_manager.views.AttendanceModelView method), 9
 create_view() (beaver_manager.views.BadgeModelView method), 9
 create_view() (beaver_manager.views.BeaverBadgeModelView method), 10
 create_view() (beaver_manager.views.BeaverModelView method), 10
 create_view() (beaver_manager.views.TripModelView method), 11
 criterion (beaver_manager.models.Attendance attribute), 4
 criterion (beaver_manager.models.BadgeCriterion attribute), 5
 Criterion (class in beaver_manager.models), 7
 criterion_id (beaver_manager.models.Attendance attribute), 4

D

date (beaver_manager.models.Attendance attribute), 4
 date (beaver_manager.models.Trip attribute), 8
 db (in module beaver_manager), 3
 delete_view() (beaver_manager.views.AttendanceModelView method), 9
 delete_view() (beaver_manager.views.BadgeModelView method), 9
 delete_view() (beaver_manager.views.BeaverBadgeModelView method), 10
 delete_view() (beaver_manager.views.BeaverModelView method), 10
 delete_view() (beaver_manager.views.TripModelView method), 11
 description (beaver_manager.models.Attendance attribute), 4
 description (beaver_manager.models.BadgeCriterion attribute), 5
 description (beaver_manager.models.Criterion attribute), 7

details_view() (beaver_manager.views.AttendanceModelView method), 9
 details_view() (beaver_manager.views.BadgeModelView method), 9
 details_view() (beaver_manager.views.BeaverBadgeModelView method), 10
 details_view() (beaver_manager.views.BeaverModelView method), 10
 details_view() (beaver_manager.views.TripModelView method), 11
 dob (beaver_manager.models.Beaver attribute), 5

E

edit_view() (beaver_manager.views.AttendanceModelView method), 9
 edit_view() (beaver_manager.views.BadgeModelView method), 9
 edit_view() (beaver_manager.views.BeaverBadgeModelView method), 10
 edit_view() (beaver_manager.views.BeaverModelView method), 11
 edit_view() (beaver_manager.views.TripModelView method), 11
 email (beaver_manager.models.EmergencyContact attribute), 7
 EmailForm (class in beaver_manager.forms), 4
 EmergencyContact (class in beaver_manager.models), 7
 export_csv() (beaver_manager.views.AttendanceModelView method), 9
 export_csv() (beaver_manager.views.BadgeModelView method), 9
 export_csv() (beaver_manager.views.BeaverBadgeModelView method), 10
 export_csv() (beaver_manager.views.BeaverModelView method), 11
 export_csv() (beaver_manager.views.TripModelView method), 11

F

first_name (beaver_manager.models.Beaver attribute), 5
 first_name (beaver_manager.models.EmergencyContact attribute), 7
 form_excluded_columns (beaver_manager.views.AttendanceModelView attribute), 9
 form_excluded_columns (beaver_manager.views.BadgeModelView attribute), 9
 form_excluded_columns (beaver_manager.views.BeaverModelView attribute), 11
 form_excluded_columns (beaver_manager.views.TripModelView attribute), 12

I

id (beaver_manager.models.Attendance attribute), 4
 id (beaver_manager.models.Badge attribute), 5

- id (beaver_manager.models.BadgeCriterion attribute), 5
 - id (beaver_manager.models.Beaver attribute), 5
 - id (beaver_manager.models.BeaverAttendance attribute), 6
 - id (beaver_manager.models.BeaverBadge attribute), 6
 - id (beaver_manager.models.BeaverTrip attribute), 6
 - id (beaver_manager.models.Criterion attribute), 7
 - id (beaver_manager.models.EmergencyContact attribute), 7
 - id (beaver_manager.models.Lodge attribute), 8
 - id (beaver_manager.models.Trip attribute), 8
 - img_url (beaver_manager.models.Badge attribute), 5
 - index() (beaver_manager.views.MyHomeView method), 11
 - index() (in module beaver_manager.views), 12
 - index_view() (beaver_manager.views.AttendanceModelView method), 9
 - index_view() (beaver_manager.views.BadgeModelView method), 9
 - index_view() (beaver_manager.views.BeaverBadgeModelView method), 10
 - index_view() (beaver_manager.views.BeaverModelView method), 11
 - index_view() (beaver_manager.views.TripModelView method), 12
 - inline_models (beaver_manager.views.BadgeModelView attribute), 9
 - inline_models (beaver_manager.views.BeaverBadgeModelView attribute), 10
 - inline_models (beaver_manager.views.BeaverModelView attribute), 11
 - internal_error() (in module beaver_manager.views), 12
- ## L
- location (beaver_manager.models.Trip attribute), 8
 - lodge (beaver_manager.models.Beaver attribute), 5
 - Lodge (class in beaver_manager.models), 8
 - lodge_id (beaver_manager.models.Beaver attribute), 5
- ## M
- message (beaver_manager.forms.EmailForm attribute), 4
 - MultiCheckboxField (class in beaver_manager.forms), 4
 - MyHomeView (class in beaver_manager.views), 11
- ## N
- name (beaver_manager.models.Badge attribute), 5
 - name (beaver_manager.models.Lodge attribute), 8
 - not_found_error() (in module beaver_manager.views), 12
 - number_of_nights (beaver_manager.models.Trip attribute), 8
- ## O
- on_model_change() (beaver_manager.views.BadgeModelView method), 9
 - on_model_change() (beaver_manager.views.BeaverModelView method), 11
 - on_model_change() (beaver_manager.views.TripModelView method), 12
 - on_model_delete() (beaver_manager.views.BadgeModelView method), 9
 - on_model_delete() (beaver_manager.views.BeaverModelView method), 11
 - on_model_delete() (beaver_manager.views.TripModelView method), 12
 - option_widget (beaver_manager.forms.MultiCheckboxField attribute), 4
 - organiser (beaver_manager.models.Attendance attribute), 4
 - overnight (beaver_manager.models.Trip attribute), 8
- ## P
- phone_number (beaver_manager.models.EmergencyContact attribute), 7
 - postcode (beaver_manager.models.EmergencyContact attribute), 7
- ## R
- register() (in module beaver_manager.views), 12
 - registers() (in module beaver_manager.views), 12
- ## S
- SECRET_KEY (in module config), 15
 - send_email() (in module beaver_manager.email), 3
 - sort_on (beaver_manager.forms.BeaverSortForm attribute), 4
 - sort_on (beaver_manager.forms.TripSortForm attribute), 4
 - SQLALCHEMY_DATABASE_URI (in module config), 15
 - SQLALCHEMY_MIGRATE_REPO (in module config), 15
 - subject (beaver_manager.forms.EmailForm attribute), 4
 - surname (beaver_manager.models.Beaver attribute), 5
 - surname (beaver_manager.models.EmergencyContact attribute), 7
- ## T
- to_percent() (in module beaver_manager.logic), 12
 - town (beaver_manager.models.EmergencyContact attribute), 7
 - trip (beaver_manager.models.BeaverTrip attribute), 7
 - Trip (class in beaver_manager.models), 8
 - trip() (in module beaver_manager.views), 12
 - TripModelView (class in beaver_manager.views), 11
 - trips() (in module beaver_manager.views), 12
 - TripSortForm (class in beaver_manager.forms), 4

U

`update_beaver_badge()` (in module `beaver_manager.logic`), [13](#)

`update_criterion()` (in module `beaver_manager.logic`), [13](#)

W

`widget` (`beaver_manager.forms.MultiCheckboxField` attribute), [4](#)

`WTF_CSRF_ENABLED` (in module `config`), [15](#)