
anishare Documentation

Release 1.1

Holger Dinkel

Jun 06, 2018

Contents:

1	Introduction	1
2	Using the software	2
2.1	Main user interface	2
2.2	Main animal manager tasks	4
2.3	Main admin tasks	7
3	Installation	10
3.1	Requirements	10
3.2	First time setup	10
3.3	Importing existing data	12
3.4	Running Tests	12
4	API documentation	14
4.1	Admin	14
4.2	Models	18
4.3	Views	23
4.4	URLs	29
	Index	31

Chapter 1

Introduction

anishare is a webservice for research institutes to share animals with the goal to re-use animals and thus minimize total animal usage.

It has been developed at the [Leibniz institute for aging research](#) in Jena. This django app is meant to be used by researchers who want to share research animals with their colleagues. The basic idea is that animals are bred for experiments; however, sometimes, not all parts of the animal are used or sometimes an experiment gets cancelled for whatever reason. By sharing animals within the institute, less animals in total have to be sacrificed for research.

Anishare is a simple database of animals offered for reuse and a easy way to claim an animal with automatic generation of email messages as well as an RSS feed for updates.

AniShare

Animals

All Animals

Organis

Search for animals

Search

RSS Feed

Logged in as: admin

Logout

Add Animal

AniShare Animals

ID ↕	# ↕	Type ↕	Sex ↕	Date of Birth ↕	Age (w) ↕	Available From/To ↕	Line ↕	Mutations ↕	Location ↕	License# ↕	ext. Info ↕	Resp. Person ↕	Added By ↕	New Owner ↕
5	1	mouse	♂	01.01.2018	22	01.06.2018 01.07.2018	ko/ko		Outside	#007	ext. ID 123 LAB123	Roger Roger (Lab B Lab)	test user	Claim!
3	10	fish	?	06.05.2018	4	01.06.2018 30.06.2018	some line		Fish Facility	#154634	248962 LAB648	Fishy McFishface (Fish Lab)	test user	Claim!
2	1	mouse	♀	01.03.2018	13	29.05.2018 29.06.2018	ki/ki		Animal House 2	#900238	159758 LAB3584	Roger Roger (Lab B Lab)	test user	Claim!
1	2	mouse	♂	01.01.2018	22	29.05.2018 30.06.2018	ko/ko		Animal House 1	#900238	274628 LAB648	Mike Smith (Lab A Lab)	test user	Claim!

Showing 1 to 4 of 4 entries



Beutenbergstraße 11
D-07745 Jena • Germany

Phone: +49 3641 65-6000
Fax: +49 3641 65-6351

E-mail: info@leibniz-fli.de
www.leibniz-fli.de
[data privacy](#)



At the moment, the software/database is geared towards handling of mice, however, it can be adjusted to handle any kind of research animal.

Chapter 2

Using the software

AniShare

AnimalsAll AnimalsOrgans

Search for animals

Search

RSS

Feed

Logged in as: admin

Logout

Add Animal

AniShare Animals

ID ↕	# ↕	Type ↕	Sex ↕	Date of Birth ↕	Age (w) ↕	Available From/To ↕	Line ↕	Mutations ↕	Location ↕	License# ↕	ext. Info ↕	Resp. Person ↕	Added By ↕	New Owner ↕
5	1	mouse	♂	01.01.2018	22	01.06.2018 01.07.2018	ko/ko		Outside	#007	ext. ID 123 LAB123	Roger Roger (Lab B Lab)	test user	Claim!
3	10	fish	?	06.05.2018	4	01.06.2018 30.06.2018	some line		Fish Facility	#154634	248962 LAB648	Fishy McFishface	test user	Claim!

The webservice is split in two parts: The **animal input** method is via the Django Admin interface (See link “Add Animal” top right) and is meant for *animal managers* only. The **claim** method is via the normal web interface and is meant for normal users (who need to be authenticated, though).

2.1 Main user interface

2.1.1 Animals

The main user-facing site is the list of animals to be shared. A user can browse this list, sort it via the headers or search for a term using the search bar.

AniShare Animals All Animals Organs Search for animals Search

RSS Feed Logged in as: admin Logout Add Animal

AniShare Animals

ID ↕	# ↕	Type ↕	Sex ↕	Date of Birth ↕	Age (w) ↕	Available From/To ↕	Line ↕	Mutations ↕	Location ↕	License# ↕	ext. Info ↕	Resp. Person ↕	Added By ↕	New Owner ↕
5	1	mouse	♂	01.01.2018	22	01.06.2018 01.07.2018	ko/ko		Outside	#007	ext. ID 123 LAB123	Roger Roger (Lab B Lab)	test user	Claim!
3	10	fish	?	06.05.2018	4	01.06.2018 30.06.2018	some line		Fish Facility	#154634	248962 LAB648	Fishy McFishface (Fish Lab)	test user	Claim!
2	1	mouse	♀	01.03.2018	13	29.05.2018 29.06.2018	ki/ki		Animal House 2	#900238	159758 LAB3584	Roger Roger (Lab B Lab)	test user	Claim!
1	2	mouse	♂	01.01.2018	22	29.05.2018 30.06.2018	ko/ko		Animal House 1	#900238	274628 LAB648	Mike Smith (Lab A Lab)	test user	Claim!

Showing 1 to 4 of 4 entries



Beutenbergstraße 11
D-07745 Jena • Germany

Phone: +49 3641 65-6000
Fax: +49 3641 65-6351

E-mail: info@leibniz-fli.de
www.leibniz-fli.de
[data privacy](#)



If a user is interested in an animal, they should click on the button “Claim” which will bring up another page (see below) in which they can review their claim before finally submitting. When they click on “Yes, I want to claim this!”, then they will be entered as *new owner* of this animal and an email will be sent to them as well as the responsible/contact person informing them about this transaction. Further steps might need to be necessary such as transferring the animal in the LIMS (eg. PyRat).

Note: If more than one animal is available, the user can adjust the number they want to claim. The remaining animals will still be available for claim.

AniShare Animals All Animals Organs Search for animals Search

RSS Feed Logged in as: hdinkel Logout Add Animal

Dear **Holger Dinkel**, do you really want to claim the following animal for yourself?

ID	#	Type	Sex	Entry Date	Date of Birth	Age (w)	Available From	Available To	Line	Mutations	Location	License#	ext. Info	Resp. Person	Added By	Comments
5	1	mouse	♂	31.05.2018	01.01.2018	22	01.06.2018	01.07.2018	ko/ko		Outside	#007	ext. ID 123 LAB123	Roger Roger (Lab B Lab)	test user	None

By clicking on the button, an email will be sent from
Holger.Dinkel@leibniz-fli.de to
Roger Roger (roger.roger@nowhere.com),
informing that you take responsibility for 1 animal from entry 5.
They will make the necessary adjustments in the database.

2.1.2 Organs

There exists an individual page for animal organ share. It is very similar to the animal page, however only individual organs are for offer. Also there is no availability period, but a day at which the animal gets sacrificed. The person responsible for killing the animal will be informed via email if anybody claims some of the available organs. The entry will remain available to others (as they might want to claim other organs).

Organ index view:

AniShare

Animals

All Animals

Organs

Search for animals

Search

RSS Feed

Logged in as: admin

Logout

Add Animal

AniShare Organs

ID ↴	# ↴	Type ↴	Sex ↴	Date of Birth ↴	Date of Death ↴	Killing Method ↴	Killing Person ↴	Age (w) ↴	Line ↴	Mutations ↴	Location ↴	License# ↴	ext. Info ↴	Resp. Person ↴	Added By ↴	New Owner ↴
1	1	mouse	♀	03.05.2018	07.06.2018	cervicale dislocation	holger.dinkel@leibniz-fli.de	5	no line		Tierhaus4	007	167842 125946	Holger Dinkel (False Lab)	Admin	Claim!

Showing 1 to 1 of 1 entries

Organ claim view:

AniShare																													
AniShare	Animals	All Animals	Organs	Search for animals										Search															
RSS																													
Feed																													
Logged in as: hdinkel																													
Logout																													
Add Animal																													

Dear **Holger Dinkel**, do you really want to claim an **organ** of the following animal for yourself?

ID	#	Type	Organ	Sex	Date of Birth	Date of Death	Method of Killing	Killing Person	Age (w)	Line	Mutations	Location	License#	ext. Info	Added By
3	2	fish	brain	?	10.02.2018	09.06.2018	CO2	hannibal.lecter@nowhere.com	17	no line		Animal House 2	007	321659 59135	admin Admin

By clicking on the button, an email will be sent from
 Holger.Dinkel@leibniz-fli.de to
 hannibal.lecter@nowhere.com
 informing that you are interested in the following organs/parts
 enter organs you want from the entry above.
 They will get in touch with you.

2.1.3 RSS Feed

An RSS feed containing the latest ten animals and organs is automatically generated and can be found at */animals/feed*. Users can subscribe (Most email clients allow the subscription to RSS feeds) to this feed to stay up-to-date with the animal catalogue. By clicking on a link in the feed, they are directed to the claim page of the individual animal/organ.

Anishare animal feed

Updates on animals to share.

[10 unknown fish, some line id:3 \[2018-05-06\]](#)

id:248962, lab_id:LAB648, available:2018-06-01-2018-06-30, location:Fish Facility, mutations:

[2 female mouse, ko/ko id:4 \[2018-03-01\]](#)

id:695968, lab_id:LAB3584, available:2018-05-01-2018-05-16, location:Animal House 2, mutations:

[2 male mouse, ko/ko id:1 \[2018-01-01\]](#)

id:274628, lab_id:LAB648, available:2018-05-29-2018-06-30, location:Animal House 1, mutations:

[1 female mouse, kj/ki id:2 \[2018-03-01\]](#)

id:159758, lab_id:LAB3584, available:2018-05-29-2018-06-29, location:Animal House 2, mutations:

2.2 Main animal manager tasks

An *animal manager* can add animals and organs to the database.

Welcome to AniShare

ANIMALS		
Animals	+ Add	Change
Organs	+ Add	Change
Persons		Change

2.2.1 Animals

Click on Animals -> Add to add an animal.

Add animal

Amount:	<input type="text" value="1"/>	<small>How many animals? (eg. fish in tank)</small>
Animal type:	<input type="text" value="mouse"/>	
Organ type:	<input type="text" value="spleen"/>	
Day of birth:	<input type="text" value="01.03.18"/>	Today
Available from:	<input type="text" value="30.05.18"/>	Today
Available to:	<input type="text" value="30.06.18"/>	Today
Sex:	<input type="radio"/> male <input checked="" type="radio"/> female <input type="radio"/> unknown <small>Select "unknown" if multiple animals.</small>	
External id:	<input type="text" value="946283"/>	
External lab id:	<input type="text" value="LAB3418"/>	
Line:	<input type="text" value="ko-/ko-"/>	
Location:	<input type="text" value="Animal House 1"/>	Change +
	<small>Where is the animal housed?</small>	
Responsible person:	<input type="text" value="Roger Roger (Lab B Lab)"/>	Change +
	<small>Person who is responsible in the lab for dealing with the animals</small>	
Licence number:	<input type="text" value="#182938"/>	

All fields in bold **need** to be filled in, the others are optional.

After adding several animals, the main (index) view should look like this:

AniShare admin interface
WELCOME, ADMIN. VIEW SITE / DOCUMENTATION / CHANGE PASSWORD / LOG OUT

Home › Animals › Animals

Select animal to change
ADD ANIMAL +

Action: 0 of 4 selected

AMOUNT	ENTRY DATE	DAY OF BIRTH	AGE (W)	AVAILABLE FROM	AVAILABLE TO	LINE	SEX	LOCATION	LICENCE NUMBER	RESPONSIBLE PERSON	ADDED BY
2	30.05.2018	01.03.2018	10	01.05.2018	16.05.2018	ko/ko	female	Animal House 2	#154253	Mike Smith (Lab A Lab)	admin
10	30.05.2018	06.05.2018	3	01.06.2018	30.06.2018	some line	unknown	Fish Facility	#154634	Fishy McFishface (Fish Lab)	admin
1	29.05.2018	01.03.2018	12	29.05.2018	29.06.2018	ki/ki	female	Animal House 2	#900238	Roger Roger (Lab B Lab)	admin
2	29.05.2018	01.01.2018	21	29.05.2018	30.06.2018	ko/ko	male	Animal House 1	#900238	Mike Smith (Lab A Lab)	admin

4 animals

FILTER

By amount

All

1

2

10

By sex

All

male

female

unknown

By responsible for lab

All

Lab A Lab

Lab B Lab

Fish Lab

By location

All

Animal House 1

Outdoors

Animal House 2

Fish Facility

2.2.2 Organs

Click on Organs -> Add to add an organ.

Add organ

Amount:
How many organs?

Animal type:

Sex: ☐ male ☐ female ☐ unknown
Select "unknown" if multiple animals.

Organ type:

Day of birth: Today

Day of death: Today

Method of killing:

Killing person:
Email address of the person who is responsible for killing the animal

Database id:
ID of animal in eg. PYRAT

Lab id:
ID of lab in eg. PYRAT

Line:
genetic trait of animal

Location:
Where is the animal housed?

All fields in bold **need** to be filled in, the others are optional.

2.3 Main admin tasks

The admin interface allows to edit the following types of entries:

Welcome to AniShare

ANIMALS		
Animals	+ Add	Change
Labs	+ Add	Change
Locations	+ Add	Change
Organs	+ Add	Change
Persons	+ Add	Change

2.3.1 Animals

The main category to administer are animals to share. Here, several filters (such as “sex”, “location”, etc.) are available to search for any set of animals.

Select animal to change

Q

Search

Action:

 Go 0 of 5 selected

<input type="checkbox"/>	AMOUNT	ENTRY DATE	DAY OF BIRTH	AGE (W)	AVAILABLE FROM	AVAILABLE TO	LINE	SEX	LOCATION	LICENCE NUMBER	RESPONSIBLE PERSON	ADC
<input type="checkbox"/>	1	31.05.2018	01.01.2018	21	01.06.2018	01.07.2018	ko/ko	male	Outside	#007	Roger Roger (Lab B Lab)	test
<input type="checkbox"/>	2	30.05.2018	01.03.2018	10	01.05.2018	16.05.2018	ko/ko	female	Animal House 2	#154253	Mike Smith (Lab A Lab)	test
<input type="checkbox"/>	10	30.05.2018	06.05.2018	3	01.06.2018	30.06.2018	some line	unknown	Fish Facility	#154634	Fishy McFishface (Fish Lab)	test
<input type="checkbox"/>	1	29.05.2018	01.03.2018	13	29.05.2018	29.06.2018	ki/ki	female	Animal House 2	#900238	Roger Roger (Lab B Lab)	test
<input type="checkbox"/>	2	29.05.2018	01.01.2018	21	29.05.2018	30.06.2018	ko/ko	male	Animal House 1	#900238	Mike Smith (Lab A Lab)	test

5 animals

FILTER

By amount

All

1

2

10

By sex

All

male

female

unknown

By responsible for lab

All

Lab A Lab

Lab B Lab

Fish Lab

By location

All

Animal House 1

Outdoors

Animal House 2

Fish Facility

Outside

Note: in order to remove a claim (thus making the animal available again), either click on an animal and remove the email address from the field “new owner”, or select one or multiple animals and select the “clear claim” *Action* and click “Go”.

Note: Once created, an animal cannot be deleted, except by the administrator.

2.3.2 Labs

Labs are research labs/research groups and need to have at least one responsible/contact person each

Select lab to change

Action: 0 of 3 selected

<input type="checkbox"/>	NAME	RESPONSIBLE PERSON
<input type="checkbox"/>	Fish	Fishy McFishface
<input type="checkbox"/>	Lab B	Roger Roger
<input type="checkbox"/>	Lab A	Mike Smith

3 labs

Note: Only *administrators* are allowed to see and change Labs

2.3.3 Locations

Locations are where animals are stored. Usually something like room numbers or “animal house” or “fish facility”.

Select location to change

Action: 0 of 5 selected

<input type="checkbox"/>	NAME
<input type="checkbox"/>	Outside
<input type="checkbox"/>	Fish Facility
<input type="checkbox"/>	Animal House 2
<input type="checkbox"/>	Outdoors
<input type="checkbox"/>	Animal House 1

5 locations

Note: Only *administrators* are allowed to see and change Locations

2.3.4 Persons

Persons responsible for the animals. Could be a vet or similar. Every animal needs to have a responsible person associated to them. This person then gets an email when the animal is being claimed.

Select person to change

Action: ----- 0 of 3 selected

<input type="checkbox"/>	NAME	EMAIL	RESPONSIBLE FOR LAB
<input type="checkbox"/>	Fishy McFishface	fishymcfishface@nowhere.com	Fish Lab
<input type="checkbox"/>	Mike Smith	mike.smith@nowhere.com	Lab A Lab
<input type="checkbox"/>	Roger Roger	roger.roger@nowhere.com	Lab B Lab

3 persons

Note: Only *administrators* are allowed to see and change Persons

2.3.5 Make a user an animal manager

In order to be able to add/edit animals, a user has to be in the group *animal manager* and have *staff status* in django admin. For this, an *administrator* has to go to the [user management](#) in the admin interface by clicking “Home” -> “Authentication and Authorization” -> “Users”. Here, they can make a *user* an *animal manager*, by setting these values (*staff* and group *animal manager*):

Permissions

☒ **Active**
 Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☒ **Staff status**
 Designates whether the user can log into this admin site.

☐ **Superuser status**
 Designates that this user has all permissions without explicitly assigning them.

Groups:

Available groups

Chosen groups

animal manager

Choose all

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

Chapter 3

Installation

3.1 Requirements

We use the latest version of [django](#), which requires [python3](#). Install django and other dependencies (see file `requirements.txt`). We recommend using a virtual environment for this):

```
virtualenv -p python3 .
source bin/activate
pip install -r requirements.txt
```

3.2 First time setup

First, in the folder `anishare`, copy the file `local_settings.py.template` to `local_settings.py` and fill it in. If you want to use LDAP, comment in the respective lines. Most importantly, you should configure the following lines:

```
EMAIL_HOST = ''
SECRET_KEY = ''
ALLOWED_HOSTS = ['127.0.0.1', ]
```

Then, you can run migrations:

```
python manage.py migrate
```

Note: This will create the sqlite database `db.sqlite3` containing all the models (eg. tables) as defined in `animals.models`.

Now create a superuser:

```
python manage.py createsuperuser
```

You are now able to login to the admin interface, but first run the dev server:

```
python manage.py runserver
```

This will listen on `http://localhost:8000`, so browse to the admin page `http://localhost:8000/admin` and you should see this after login:

AniShare admin interface

Welcome to AniShare

ANIMALS

Animals	+ Add	Change
Labs	+ Add	Change
Locations	+ Add	Change
Organs	+ Add	Change
Persons	+ Add	Change

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

Recent actions

My actions

- 1 female mouse, no line id:1 [2018-03-14]
Organ
- + 1 female mouse, no line id:1 [2018-03-14]
Organ
- testuser
User
- admin
User

You can also import a dummy set of data using the `loaddata` command:

```
python manage.py loaddata initial_data.json
```

After loading the data, the main admin interface should look like this:

AniShare admin interface

WELCOME, ADMIN. VIEW SITE / DOCUMENTATION / CHANGE PASSWORD / LOG OUT

Home > Animals > Animals

Select animal to change

Action: ----- 0 of 4 selected

AMOUNT	ENTRY DATE	DAY OF BIRTH	AGE (W)	AVAILABLE FROM	AVAILABLE TO	LINE	SEX	LOCATION	LICENCE NUMBER	RESPONSIBLE PERSON	ADDED BY
2	30.05.2018	01.03.2018	10	01.05.2018	16.05.2018	ko/ko	female	Animal House 2	#154253	Mike Smith (Lab A Lab)	admin
10	30.05.2018	06.05.2018	3	01.06.2018	30.06.2018	some line	unknown	Fish Facility	#154634	Fishy McFishface (Fish Lab)	admin
1	29.05.2018	01.03.2018	12	29.05.2018	29.06.2018	ki/ki	female	Animal House 2	#900238	Roger Roger (Lab B Lab)	admin
2	29.05.2018	01.01.2018	21	29.05.2018	30.06.2018	ko/ko	male	Animal House 1	#900238	Mike Smith (Lab A Lab)	admin

4 animals

ADD ANIMAL +

FILTER

By amount

All

1

2

10

By sex

All

male

female

unknown

By responsible for lab

All

Lab A Lab

Lab B Lab

Fish Lab

By location

All

Animal House 1

Outdoors

Animal House 2

Fish Facility

3.3 Importing existing data

For import of existing data in tabular (excel) format, a management command is available at `animals.models.animals.management.commands.import_animals`

```
python manage.py import_animals
```

Note: See the file `example_import.xls` for an example...

Information exported from PYRAT:										Information to add manually			
Nr.	Entry date	ID	Lab ID	DOB	Age (w)	Line / Strain	Sex	Building	Licence num	Responsible	Available from	Available until	New Owner
1	01.05.18	ext. ID 123	LAB123	01.01.18	10	ko/ko	m	Outside	#007	Roger Roger	01.06.18	01.07.18	

3.4 Running Tests

Tests reside in `animals/tests.py`. You can invoke the django tests like so:

```
python manage.py test
```

```

1  """
2  Tests for Anishare website
3  """
4  from django.test import TestCase, Client
5  from django.contrib.auth.models import User
6  from django.core.management import call_command
7
8  class GetAnimalsTest(TestCase):
9      """
10     Test module to GET Animal pages
11     """
12
13     def setUp(self):
14         """ Creating a user first and loading fixtures """
15         call_command('loaddata', 'initial_data.json', verbosity=0) # Load fixtures
16         self.user = User.objects.create_user(pk=1, username='testuser', password=
17         ↪ '12345')
18         self.client = Client()
19
20     def test_get_all_animals(self):
21         """ try to retrieve all animals """
22         response = self.client.get('/animals/')
23         self.assertEqual(response.status_code, 302)
24         self.client.login(username='testuser', password='12345')
25         response = self.client.get('/animals/')
26         self.assertEqual(response.status_code, 200)
27
28     def test_get_one_animal(self):
29         """ try to retrieve individual animals """
30         response = self.client.get('/animals/1')
31         self.assertEqual(response.status_code, 302)
32         self.client.login(username='testuser', password='12345')
33         response = self.client.get('/animals/1')

```

(continues on next page)

(continued from previous page)

```

33     self.assertEqual(response.status_code, 200)
34
35     def test_claim_one_animal(self):
36         """ try to claim individual animals """
37         response = self.client.get('/animals/claim/1')
38         self.assertEqual(response.status_code, 302)
39         self.client.login(username='testuser', password='12345')
40         response = self.client.get('/animals/claim/1')
41         self.assertEqual(response.status_code, 200)

```

3.4.1 Upgrading django

To upgrade django or any other python library for anishare, go into the anishare directory, and activate its virtualenv:

```

cd anishare
source bin/activate

```

Next, install/upgrade whatever library (here: django to the latest version):

```

pip install --upgrade django

```

Note: It's best to test the latest version in a local/development environment first!

3.4.2 Upgrading python

When upgrading the python version of the host operating system, it might be necessary to also upgrade the python in the virtualenv. Otherwise an error like the following might occur:

```

python: error while loading shared libraries: libpython3.4m.so.1.0:
cannot open shared object file: No such file or directory

```

In that case, go into the anishare directory, and delete the following directories:

- bin
- include
- lib
- lib64

Afterwards, create a new virtualenv and install the required libraries like so:

```

virtualenv -p python3 .
source bin/activate
pip install -r requirements.txt

```

API documentation

4.1 Admin

Admin module

class `animals.admin.AnimalAdmin(model, admin_site)`
ModelAdmin for Animal model

age (*obj*)
Show the age in the admin as 'Age (w)' instead of 'age'

form
alias of `AnimalForm`

save_model (*request, obj, form, change*)
Given a model instance save it to the database.

class `animals.admin.AnimalForm(data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, instance=None, use_required_attribute=None)`

Form for animal editing in admin

clean ()
Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

class `animals.admin.LabAdmin(model, admin_site)`
ModelAdmin for Lab model

class `animals.admin.LocationAdmin(model, admin_site)`
ModelAdmin for Location model

class `animals.admin.OrganAdmin(model, admin_site)`
ModelAdmin for Organ model

age (*obj*)
Show the age in the admin as 'Age (w)' instead of 'age'

save_model (*request, obj, form, change*)
Given a model instance save it to the database.


```
class animals.admin.PersonAdmin(model, admin_site)
```

```
    ModelAdmin for Person model
```

```
animals.admin.clear_claim(modeladmin, request, queryset)
```

```
    Convenience Function to delete a claim from several selected animals
```

```
1  """
2  Admin module
3  """
4  from datetime import datetime, timedelta
5  from django.contrib import admin
6  from django import forms
7  from django.conf import settings
8  from .models import Animal, Person, Lab, Location, Organ
9  from rangefilter.filter import DateRangeFilter, DateTimeRangeFilter
10
11
12  admin.site.site_header = 'AniShare admin interface'
13  admin.site.site_title = 'AniShare'
14  admin.site.index_title = 'Welcome to AniShare'
15
16  class AnimalForm(forms.ModelForm):
17      """
18      Form for animal editing in admin
19      """
20      class Meta:
21          model = Animal
22          fields = ('amount', 'animal_type', 'day_of_birth',
23                  'available_from', 'available_to', 'sex', 'database_id',
24                  'lab_id', 'line', 'location', 'responsible_person',
25                  'licence_number', 'mutations', 'comment', 'new_owner', )
26
27      def clean(self):
28          available_from = self.cleaned_data.get('available_from')
29          available_to = self.cleaned_data.get('available_to')
30          day_of_birth = self.cleaned_data.get('day_of_birth')
31          # self.author = request.user
32          if available_from > available_to:
33              raise forms.ValidationError("Dates are incorrect")
34          if day_of_birth and (
35              (datetime.now().date() - day_of_birth) <=
36              timedelta(days=settings.MIN_SHARE_DURATION_PUPS)):
37              if available_to - available_from <= timedelta(days=settings.MIN_SHARE_
38  ↳DURATION_PUPS):
39                  raise forms.ValidationError(
40                      "Minimum share duration for pups must be {} days!".format(
41                          settings.MIN_SHARE_DURATION_PUPS))
42              elif available_to - available_from <= timedelta(days=settings.MIN_SHARE_
43  ↳DURATION):
44                  raise forms.ValidationError(
45                      "Minimum share duration must be {} days!".format(settings.MIN_SHARE_
46  ↳DURATION))
47              return self.cleaned_data
48
49  @admin.register(Person)
50  class PersonAdmin(admin.ModelAdmin):
51      """
52      ModelAdmin for Person model
53      """
```

(continues on next page)

(continued from previous page)

```

50     """
51     list_display = ('name', 'email', 'responsible_for_lab')
52     search_fields = ('name', 'email', 'responsible_for_lab__name')
53     ordering = ('name', )
54
55 @admin.register(Location)
56 class LocationAdmin(admin.ModelAdmin):
57     """
58     ModelAdmin for Location model
59     """
60     list_display = ('name',)
61     search_fields = ('name',)
62
63 def clear_claim(modeladmin, request, queryset):
64     """
65     Convenience Function to delete a claim from several selected animals
66     """
67     queryset.update(new_owner='')
68     clear_claim.short_description = "Clear 'new_owner' from selected animals"
69
70
71 @admin.register(Animal)
72 class AnimalAdmin(admin.ModelAdmin):
73     """
74     ModelAdmin for Animal model
75     """
76     list_display = ('amount', 'entry_date', 'day_of_birth', 'age', 'available_from',
77                    'available_to', 'line', 'sex', 'location', 'licence_number',
78                    'responsible_person', 'added_by', 'new_owner')
79     list_display_links = ('amount', 'entry_date', 'day_of_birth', 'age',
80                           'available_from', 'available_to', 'line', 'sex',
81                           'location', 'licence_number', 'responsible_person',
82                           'added_by', 'new_owner')
83     search_fields = ('amount', 'database_id', 'lab_id', 'day_of_birth',
84                    'line', 'sex', 'location__name', 'new_owner', 'licence_number',
85                    'mutations', 'available_from', 'available_to', 'responsible_
86     ↪ person__name',
87                    'responsible_person__email', 'added_by_email')
88     autocomplete_fields = ['responsible_person']
89     list_filter = ('amount', 'sex', 'responsible_person__responsible_for_lab',
90                  ('day_of_birth', DateRangeFilter),
91                  'location', 'licence_number', 'new_owner', 'added_by')
92     radio_fields = {'sex': admin.HORIZONTAL}
93     readonly_fields = ('creation_date', 'modification_date')
94     form = AnimalForm
95     actions = [clear_claim,]
96     def age(self, obj):
97         """Show the age in the admin as 'Age (w)' instead of 'age'"""
98         return obj.age()
99     age.short_description = 'Age (w)'
100 #     def formfield_for_foreignkey(self, db_field, request, **kwargs):
101 #         if db_field.name == 'author':
102 #             kwargs['queryset'] = get_user_model().objects.filter(username=request.
103 ↪ user.username)
104 #         return super(AnimalAdmin, self).formfield_for_foreignkey(db_field, request,
105 ↪ **kwargs)
106     def save_model(self, request, obj, form, change):

```

(continues on next page)

(continued from previous page)

```

104         if not obj.pk:
105             # Only set added_by during the first save.
106             obj.added_by = request.user
107         super().save_model(request, obj, form, change)
108
109 @admin.register(Organ)
110 class OrganAdmin(admin.ModelAdmin):
111     """
112     ModelAdmin for Organ model
113     """
114     list_display = ('amount', 'animal_type', 'organ_type', 'entry_date', 'day_of_birth
115 ↪',
116 ↪          'day_of_death', 'age', 'method_of_killing', 'killing_person',
117 ↪ 'line',
118 ↪          'sex', 'location', 'licence_number', 'responsible_person', 'added_
119 ↪by')
120     list_display_links = ('amount', 'animal_type', 'organ_type', 'entry_date', 'day_
121 ↪of_birth',
122 ↪          'day_of_death', 'age', 'method_of_killing', 'killing_person
123 ↪', 'line',
124 ↪          'sex', 'location', 'licence_number', 'responsible_person',
125 ↪ 'added_by')
126     search_fields = ('amount', 'animal_type', 'organ_type', 'entry_date', 'day_of_
127 ↪birth',
128 ↪          'day_of_death', 'age', 'method_of_killing', 'killing_person',
129 ↪ 'line',
130 ↪          'sex', 'location', 'licence_number', 'responsible_person',
131 ↪ 'added_by')
132     autocomplete_fields = ['responsible_person']
133     list_filter = ('amount', 'sex', 'responsible_person__responsible_for_lab',
134 ↪('day_of_birth', DateRangeFilter), ('day_of_death',
135 ↪DateRangeFilter),
136 ↪          'location', 'licence_number', 'added_by')
137     radio_fields = {'sex': admin.HORIZONTAL}
138     readonly_fields = ('added_by', 'creation_date', 'modification_date')
139     # form = OrganForm
140     actions = [clear_claim,]
141
142     def age(self, obj):
143         """Show the age in the admin as 'Age (w)' instead of 'age'"""
144         return obj.age()
145     age.short_description = 'Age (w)'
146
147     def save_model(self, request, obj, form, change):
148         if not obj.pk:
149             # Only set added_by during the first save.
150             obj.added_by = request.user
151         super().save_model(request, obj, form, change)
152
153 @admin.register(Lab)
154 class LabAdmin(admin.ModelAdmin):
155     """
156     ModelAdmin for Lab model
157     """
158     list_display = ('name', 'responsible_person')
159     search_fields = ('name',)

```

4.2 Models

This file describes all the models in the database.

```
class animals.models.Animal (*args, **kwargs)
    Main model containing the animals.

    exception DoesNotExist

    exception MultipleObjectsReturned

    age ()
        Return the age of the animal, calculated by the difference to either the current date or the available_to date

    available ()
        Returns True if the animal is still available

    description ()
        Return description of this model

    get_absolute_url ()
        Get absolute url for this model. Important to link from the admin.

class animals.models.Lab (*args, **kwargs)
    Labs are only defined by a name and are referenced by Person(s) which are responsible (contact) person for this
    lab

    exception DoesNotExist

    exception MultipleObjectsReturned

    responsible_person ()
        Retrieve only the person(s) which are responsible for this lab.

class animals.models.Location (*args, **kwargs)
    Location of animals. Eg. animal house, fish facilities etc.

    exception DoesNotExist

    exception MultipleObjectsReturned

class animals.models.Organ (*args, **kwargs)
    Model containing the organs

    exception DoesNotExist

    exception MultipleObjectsReturned

    age ()
        Return the age of the animal, at the time of death

    available ()
        Returns True if the animal is still available

    description ()
        Return description of this model

    get_absolute_url ()
        Get absolute url for this model. Important to link from the admin.

class animals.models.Person (*args, **kwargs)
    The responsible (contact) person for each lab. This person gets an email when an animal is being claimed.

    exception DoesNotExist
```

exception MultipleObjectsReturned

```

1  """
2  This file describes all the models in the database.
3  """
4  from datetime import datetime
5  from django.urls import reverse
6  from django.db import models
7  from django.contrib.auth.models import User
8
9  class Lab(models.Model):
10     """
11     Labs are only defined by a name and are referenced by
12     Person(s) which are responsible (contact) person for this lab
13     """
14     name = models.CharField(max_length=200)
15     def __str__(self):
16         return self.name + ' Lab'
17     def responsible_person(self):
18         """
19         Retrieve only the person(s) which are responsible for this lab.
20         """
21         persons = Person.objects.filter(responsible_for_lab=self)
22         return ', '.join(i.name for i in persons)
23
24  class Person(models.Model):
25     """
26     The responsible (contact) person for each lab.
27     This person gets an email when an animal is being claimed.
28     """
29     name = models.CharField(max_length=200)
30     email = models.EmailField()
31     responsible_for_lab = models.ForeignKey(Lab, on_delete=models.CASCADE, default=0)
32     def __str__(self):
33         return self.name + ' (' + str(self.responsible_for_lab) + ')'
34
35  class Location(models.Model):
36     """
37     Location of animals. Eg. animal house, fish facilities etc.
38     """
39     name = models.CharField(max_length=200)
40     def __str__(self):
41         return self.name
42
43  class Animal(models.Model):
44     """
45     Main model containing the animals.
46     """
47     amount = models.PositiveIntegerField(default=1,
48                                         help_text="How many animals? (eg. fish in_
49 ↪ tank)")
50     animal_type = models.CharField(max_length=100, choices=(
51         ('fish', 'fish'),
52         ('mouse', 'mouse'),
53     ),
54                                     default='mouse')
55     database_id = models.CharField(max_length=200, help_text="ID of animal in eg.
56 ↪ PYRAT")

```

(continues on next page)

(continued from previous page)

```

55     lab_id = models.CharField(max_length=200, help_text="ID of lab in eg. PYRAT")
56     creation_date = models.DateTimeField(null=False, auto_now_add=True)
57     modification_date = models.DateTimeField(null=False, auto_now=True)
58     entry_date = models.DateField(null=False, auto_now_add=True)
59     day_of_birth = models.DateField()
60     line = models.CharField(max_length=200, help_text="genetic trait of animal")
61     sex = models.CharField(max_length=2, choices= (('m', 'male'), ('f', 'female'), ('u
↳ ', 'unknown'))),
62                                     help_text='Select "unknown" if multiple animals.')
63     location = models.ForeignKey(Location, on_delete=models.CASCADE,
64                                   help_text='Where is the animal housed?')
65     mutations = models.TextField(blank=True, null=True,
66                                   help_text="Describe the mutations of this line in as
↳ much detail as possible")
67     licence_number = models.CharField(max_length=200)
68     responsible_person = models.ForeignKey(Person, on_delete=models.CASCADE,
↳ default=0,
69                                     help_text='Person who is responsible in
↳ the lab for dealing with the animals')
70     available_from = models.DateField()
71     available_to = models.DateField() # default=datetime.today() + timedelta(days=15))
72     comment = models.TextField(blank=True, null=True,
73                                   help_text='Comments, such as individual organs to be
↳ offered')
74     new_owner = models.CharField(max_length=200, blank=True,
75                                   help_text='Person claiming this animal for themselves
↳ ') # turn into foreignkey to auth_users?
76     added_by = models.ForeignKey(User, unique=False, on_delete=models.CASCADE,
↳ default=1)
77
78     def age(self):
79         """
80         Return the age of the animal, calculated by the difference to either
81         the current date or the available_to date
82         """
83         # return int((self.entry_date - self.day_of_birth).days / 7)
84         now = datetime.today().date()
85         if now < self.available_to:
86             return int((now - self.day_of_birth).days / 7)
87         return int((self.available_to - self.day_of_birth).days / 7)
88
89     def available(self):
90         """
91         Returns True if the animal is still available
92         """
93         today = datetime.now().date()
94         return (self.available_from <= today) and (today <= self.available_to)
95
96     def get_absolute_url(self):
97         """
98         Get absolute url for this model. Important to link from the admin.
99         """
100         return reverse('animals:claim', kwargs={'primary_key': self.pk})
101
102     def __str__(self):
103         return "{} {} {}, {} id: {} [{}]" .format(
104             self.amount, self.get_sex_display(), self.animal_type, self.line, self.pk,
↳ self.day_of_birth)

```

(continues on next page)

(continued from previous page)

```

105
106     def description(self):
107         """
108         Return description of this model
109         """
110         return "id:{}, lab_id:{}, available:{}-{}, location:{}, mutations:{}".format(
111             self.database_id, self.lab_id, self.available_from,
112             self.available_to, self.location, "".join(self.mutations))
113
114
115
116 class Organ(models.Model):
117     """
118     Model containing the organs
119     """
120     amount = models.PositiveIntegerField(default=1,
121                                         help_text="How many organs?")
122     animal_type = models.CharField(max_length=100, choices=(
123         ('fish', 'fish'),
124         ('mouse', 'mouse'),
125         ('unknown', 'unknown'),
126     ),
127                                     default='mouse')
128     sex = models.CharField(max_length=2, choices= (('m', 'male'), ('f', 'female'), ('u
129 ↪', 'unknown'))),
130                                     help_text='Select "unknown" if multiple animals.')
131     organ_type = models.CharField(max_length=100, choices=(
132         ('bladder', 'bladder'),
133         ('bone marrow', 'bone marrow'),
134         ('brain', 'brain'),
135         ('genitals', 'genitals'),
136         ('heart', 'heart'),
137         ('intestine', 'intestine'),
138         ('kidney', 'kidney'),
139         ('liver', 'liver'),
140         ('lungs', 'lungs'),
141         ('spleen', 'spleen'),
142         ('stomach', 'stomach'),
143         ('other', 'other'),
144     ),
145                                     )
146     day_of_birth = models.DateField()
147     day_of_death = models.DateField()
148     method_of_killing = models.CharField(max_length=100, choices=(
149         ('CO2', 'CO2'),
150         ('cervicale dislocation', 'cervicale dislocation'),
151         ('decapitation', 'decapitation'),
152         ('blood withdrawl', 'blood withdrawl'),
153         ('finale heart punction', 'finale heart punction'),
154         ('overdose anaesthetics', 'overdose anaesthetic'),
155         ('other', 'other'),
156     ),)
157     killing_person = models.EmailField(help_text='Email address of the person who is
158 ↪responsible for killing the animal')
159     database_id = models.CharField(max_length=200, help_text="ID of animal in eg.
160 ↪PYRAT")
161     lab_id = models.CharField(max_length=200, help_text="ID of lab in eg. PYRAT")

```

(continues on next page)

(continued from previous page)

```

159     entry_date = models.DateField(null=False, auto_now_add=True)
160     line = models.CharField(max_length=200, help_text="genetic trait of animal")
161     location = models.ForeignKey(Location, on_delete=models.CASCADE, help_text='Where
↳ is the animal housed?')
162     licence_number = models.CharField(max_length=200)
163     responsible_person = models.ForeignKey(Person, on_delete=models.CASCADE,
↳ default=0,
164                                           help_text='Person who is responsible in
↳ the lab for dealing with the animals')
165     mutations = models.TextField(blank=True, null=True, help_text="Describe the
↳ mutations of this line in as much detail as possible")
166     comment = models.TextField(blank=True, null=True,
167                               help_text='Comments, such as individual organs to be
↳ offered')
168     # new_owner = models.CharField(max_length=200, blank=True,
169     #                               help_text='Person claiming this animal for
↳ themselves') # turn into foreignkey to auth_users?
170     creation_date = models.DateTimeField(null=False, auto_now_add=True)
171     modification_date = models.DateTimeField(null=False, auto_now=True)
172     added_by = models.ForeignKey(User, unique=False, on_delete=models.CASCADE,
↳ default=1)
173
174     def age(self):
175         """
176         Return the age of the animal, at the time of death
177         """
178         return int((self.day_of_death - self.day_of_birth).days / 7)
179
180     def available(self):
181         """
182         Returns True if the animal is still available
183         """
184         today = datetime.now().date()
185         return self.day_of_death >= today
186
187     def get_absolute_url(self):
188         """
189         Get absolute url for this model. Important to link from the admin.
190         """
191         return reverse('claim_organ', kwargs={'primary_key': self.pk})
192
193     def __str__(self):
194         return "{} {} {}, {} id:{{}} [{}]" .format(
195             self.amount, self.get_sex_display(), self.animal_type, self.line, self.pk,
↳ self.day_of_birth)
196
197     def description(self):
198         """
199         Return description of this model
200         """
201         return "id:{{}}, lab_id:{{}}, available:{{}}-{{}}, location:{{}}, mutations:{{}}".format(
202             self.database_id, self.lab_id, self.day_of_birth,
203             self.day_of_death, self.location, "".join(self.mutations))
204

```


4.3 Views

Django Views contains all the functions for rendering objects (HTML display). It also contains an RSS Feed generator class to create an RSS feed from newly created animals

Important: When adding new functions, use the `login_required` decorator When adding new classes, use the `LoginRequiredMixin`

```
class animals.views.AnimalDetailView (**kwargs)
```

Detail view for an animal. This is rarely used, rather use the claim page.

model

alias of `animals.models.Animal`

```
class animals.views.AnimalIndexView (**kwargs)
```

Index / List view for all available animals. Generic ListView using the `LoginRequiredMixin`

Parameters

- **q** – query / search term to filter the results
- **show** – limit the results to ‘current’, ‘archive’, or all animals

```
get_queryset ()
```

Return the latest additions to the Animals table

model

alias of `animals.models.Animal`

```
class animals.views.LatestAnimalsFeed
```

RSS Feed for new animals/organs.

```
item_description (item)
```

What to print as item description (use default description from model).

```
item_title (item)
```

What to print as item title (use default `__str__` of model).

```
items ()
```

Get latest animals as items.

```
class animals.views.OrganIndexView (**kwargs)
```

Index / List view for all available Organs. Generic ListView using the `LoginRequiredMixin`

Parameters

- **q** – query / search term to filter the results
- **show** – limit the results to ‘current’, ‘archive’, or all Organs

```
get_queryset ()
```

Return the latest additions to the Organs table

model

alias of `animals.models.Organ`

```
animals.views.claim (request, primary_key)
```

View to claim an animal.

Parameters **primary_key** – the id/pk of the animal to retrieve

Returns rendered page with the claim form or 404 if animal not found

```
animals.views.claim_organ (request, primary_key)
```

View to claim an organ.

Parameters `primary_key` – the id/pk of the organ to retrieve

Returns rendered page with the claim form or 404 if organ not found

`animals.views.send_email_animal(request)`

Function to send an email about an animal being claimed.

Needs these variables in the POST request: email, pk, count

Parameters

- **email** – email address of the request user / new owner
- **pk** – primary_key of the animal(s) to be claimed
- **count** – how many animals are being claimed

`animals.views.send_email_organ(request)`

Function to send an email about an animal being claimed.

Needs these variables in the POST request: email, pk, count

Parameters

- **email** – email address of the request user / new owner
- **pk** – primary_key of the animal(s) to be claimed
- **organs_wanted** – organs wanted from the given animal

```

1  """
2  Django Views contains all the functions for rendering objects (HTML display).
3  It also contains an RSS Feed generator class to create an RSS feed from newly created_
  ↪ animals
4
5  **Important**:
6      When adding new functions, use the login_required decorator
7      When adding new classes, use the LoginRequiredMixin
8  """
9  import operator
10 from functools import reduce
11 from datetime import datetime
12 from django import forms
13 from django.contrib import messages
14 from django.contrib.auth.decorators import login_required
15 from django.contrib.auth.mixins import LoginRequiredMixin
16 from django.contrib.syndication.views import Feed
17 from django.db.models import Q
18 from django.core.mail import EmailMessage
19 from django.http import HttpResponse
20 from django.http import HttpResponseRedirect
21 from django.shortcuts import get_object_or_404, render
22 from django.template.loader import render_to_string
23 #from django.urls import reverse
24 from django.views import generic
25
26 from .filters import AnimalFilter
27 from .models import Animal, Organ
28
29 @login_required
30 def claim(request, primary_key):
31     """
32     View to claim an animal.

```

(continues on next page)

(continued from previous page)

```

33
34     :param primary_key: the id/pk of the animal to retrieve
35
36     :returns: rendered page with the claim form
37               or 404 if animal not found
38     """
39     animal = get_object_or_404(Animal, pk=primary_key)
40     return render(request, 'animals/animal-claim.html', {'object': animal})
41
42 @login_required
43 def claim_organ(request, primary_key):
44     """
45     View to claim an organ.
46
47     :param primary_key: the id/pk of the organ to retrieve
48
49     :returns: rendered page with the claim form
50               or 404 if organ not found
51     """
52     organ = get_object_or_404(Organ, pk=primary_key)
53     return render(request, 'animals/organ-claim.html', {'object': organ})
54
55 class AnimalDetailView(LoginRequiredMixin, generic.DetailView):
56     """
57     Detail view for an animal.
58     This is rarely used, rather use the claim page.
59     """
60     model = Animal
61     template_name = 'animals/animal-detail.html'
62
63 class AnimalIndexView(LoginRequiredMixin, generic.ListView):
64     """
65     Index / List view for all available animals.
66     Generic ListView using the LoginRequiredMixin
67
68     :param q: query / search term to filter the results
69     :param show: limit the results to 'current', 'archive', or all animals
70     """
71     model = Animal
72     template_name = 'animals/index.html'
73     context_object_name = 'all_animals'
74     paginate_by = 100
75     def get_queryset(self):
76         """Return the latest additions to the Animals table"""
77         result = super(AnimalIndexView, self).get_queryset()
78         query = self.request.GET.get('q')
79         if query:
80             query_list = query.split()
81             result = result.filter(
82                 reduce(operator.and_, (Q(comment__icontains=q) for q in query_list)) |
83                 reduce(operator.and_, (Q(mutations__icontains=q) for q in query_
84 ↪list)) |
85                 reduce(operator.and_, (Q(database_id__icontains=q) for q in query_
86 ↪list)) |
87                 reduce(operator.and_, (Q(line__icontains=q) for q in query_list)) |
88                 reduce(operator.and_, (Q(lab_id__icontains=q) for q in query_list)) |
89                 reduce(operator.and_, (Q(location_name__icontains=q) for q in query_
90 ↪list)) |

```

(continues on next page)

(continued from previous page)

```

88         reduce(operator.and_, (Q(new_owner__icontains=q) for q in query_
↪list)) |
89         reduce(operator.and_, (Q(responsible_person__name__icontains=q) for q_
↪in query_list)) |
90         reduce(operator.and_, (Q(licence_number__icontains=q) for q in query_
↪list))
91     )
92     return result
93     try:
94         show = self.kwargs['show']
95     except KeyError:
96         show = 'current'
97     if show == 'archive':
98         return Animal.objects.filter(available_to__lte=datetime.now().date()).
↪order_by('-entry_date')
99     elif show == 'current':
100        return Animal.objects.filter(available_to__gte=datetime.now().date()).
↪order_by('-entry_date')
101        return Animal.objects.order_by('-entry_date')
102
103 class OrganIndexView(LoginRequiredMixin, generic.ListView):
104     """
105     Index / List view for all available Organs.
106     Generic ListView using the LoginRequiredMixin
107
108     :param q: query / search term to filter the results
109     :param show: limit the results to 'current', 'archive', or all Organs
110     """
111     model = Organ
112     template_name = 'animals/organs.html'
113     context_object_name = 'all_organs'
114     paginate_by = 100
115     def get_queryset(self):
116         """Return the latest additions to the Organs table"""
117         result = super(OrganIndexView, self).get_queryset()
118         query = self.request.GET.get('q')
119         if query:
120             query_list = query.split()
121             result = result.filter(
122                 reduce(operator.and_, (Q(comment__icontains=q) for q in query_list)) |
123                 reduce(operator.and_, (Q(mutations__icontains=q) for q in query_
↪list)) |
124                 reduce(operator.and_, (Q(database_id__icontains=q) for q in query_
↪list)) |
125                 reduce(operator.and_, (Q(line__icontains=q) for q in query_list)) |
126                 reduce(operator.and_, (Q(lab_id__icontains=q) for q in query_list)) |
127                 reduce(operator.and_, (Q(location__name__icontains=q) for q in query_
↪list)) |
128                 reduce(operator.and_, (Q(killing_person__icontains=q) for q in query_
↪list)) |
129                 reduce(operator.and_, (Q(animal_type__icontains=q) for q in query_
↪list)) |
130                 reduce(operator.and_, (Q(method_of_killing__icontains=q) for q in_
↪query_list)) |
131                 reduce(operator.and_, (Q(licence_number__icontains=q) for q in query_
↪list))
132             )

```

(continues on next page)

(continued from previous page)

```

133         return result
134     return Organ.objects.order_by('-entry_date')
135
136 def send_email_animal(request):
137     """
138     Function to send an email about an animal being claimed.
139
140     Needs these variables in the POST request: email, pk, count
141
142     :param email: email address of the request user / new owner
143     :param pk: primary_key of the animal(s) to be claimed
144     :param count: how many animals are being claimed
145     """
146     email = request.POST['email']
147     primary_key = request.POST['pk']
148     count = request.POST['count']
149
150     animal = Animal.objects.get(pk=primary_key)
151     animal.new_owner = email
152     amount_difference = int(animal.amount) - int(count)
153     if amount_difference < 0: # Save remainder of animals as new object
154         messages.add_message(request, messages.ERROR, 'You cannot claim more animals_
155         then are available!')
156         raise forms.ValidationError("You cannot claim more animals then are available!
157         ")
158         animal.amount = count
159         animal.save() # Save the animal with the new owner
160         messages.add_message(request, messages.SUCCESS,
161         'The entry {} has been claimed by {}'.format(animal.pk,
162         animal.new_owner))
163         subject = "User {} claimed animal {} in AniShare".format(email, primary_key)
164         message = render_to_string('email.html', {'email': email, 'object': animal, 'now
165         ': datetime.now()})
166         msg = EmailMessage(subject, message, email, [animal.responsible_person.email,
167         email])
168         msg.content_subtype = "html"
169         msg.send()
170         if amount_difference > 0: # If there were multiple animals, save the remainder_
171         of animals as a new object
172             animal.pk = None
173             animal.amount = amount_difference
174             animal.new_owner = ''
175             animal.save()
176             messages.add_message(request, messages.SUCCESS, 'The amount of available_
177             animals in this entry has been reduced to {}'.format(animal.amount))
178             messages.add_message(request, messages.SUCCESS, 'An Email has been sent to <{}>.'.
179             format(animal.responsible_person.email))
180
181     return HttpResponseRedirect('/')
182
183 def send_email_organ(request):
184     """
185     Function to send an email about an animal being claimed.
186
187     Needs these variables in the POST request: email, pk, count
188

```

(continues on next page)

(continued from previous page)

```

182     :param email: email address of the request user / new owner
183     :param pk: primary_key of the animal(s) to be claimed
184     :param organs_wanted: organs wanted from the given animal
185     """
186     email = request.POST['email']
187     primary_key = request.POST['pk']
188     organs_wanted = request.POST['organs_wanted']
189
190     organ = Organ.objects.get(pk=primary_key)
191     subject = "AniShare User {} claimed organ(s) {}".format(email, organs_wanted)
192     message = render_to_string('email.html', {'email': email, 'organs_wanted':organs_
193     ↪wanted, 'object': organ, 'now': datetime.now()})
194
195     msg = EmailMessage(subject, message, email, [organ.responsible_person.email,
196     ↪email])
197     msg.content_subtype = "html"
198     msg.send()
199     messages.add_message(request, messages.SUCCESS, 'An Email has been sent to <{}>.'.
200     ↪format(organ.responsible_person.email))
201
202     return HttpResponseRedirect('/organs/')
203
204 class LatestAnimalsFeed(Feed):
205     """
206     RSS Feed for new animals/organs.
207     """
208     title = 'Anishare animal/organ feed'
209     link = '/animals/feed'
210     description = 'Updates on animals/organs to share.'
211
212     def __call__(self, request, *args, **kwargs):
213         if not request.user.is_authenticated:
214             return HttpResponseRedirect(status=401)
215         return super().__call__(request, *args, **kwargs)
216
217     def items(self):
218         """
219         Get latest animals as items.
220         """
221         from itertools import chain
222         animals = Animal.objects.order_by('-entry_date')[:10]
223         organs = Organ.objects.order_by('-entry_date')[:10]
224         return chain(animals, organs)
225
226     def item_title(self, item):
227         """
228         What to print as item title (use default __str__ of model).
229         """
230         return item
231
232     def item_description(self, item):
233         """
234         What to print as item description (use default description from model).
235         """
236         return item.description()

```

(continues on next page)

(continued from previous page)

```

236 @login_required
237 def animal_list(request):
238     f = AnimalFilter(request.GET, queryset=Animal.objects.order_by('-entry_date'))
239     return render(request, 'animals/animal-index.html', {'filter': f})

```

4.4 URLs

animals URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see: <https://docs.djangoproject.com/en/2.0/topics/http/urls/>

Examples:

Function views:

1. Add an import: from `my_app` import `views`
2. Add a URL to `urlpatterns`: `path('', views.home, name='home')`

Class-based views:

1. Add an import: from `other_app.views` import `Home`
2. Add a URL to `urlpatterns`: `path('', Home.as_view(), name='home')`

Including another `URLconf`:

1. Import the `include()` function: from `django.urls` import `include`, `path`
2. Add a URL to `urlpatterns`: `path('blog/', include('blog.urls'))`

```

1  """animals URL Configuration
2
3  The `urlpatterns` list routes URLs to views. For more information please see:
4      https://docs.djangoproject.com/en/2.0/topics/http/urls/
5
6  Examples:
7
8  Function views:
9      1. Add an import:  from my_app import views
10     2. Add a URL to urlpatterns:  path('', views.home, name='home')
11
12 Class-based views:
13     1. Add an import:  from other_app.views import Home
14     2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
15
16 Including another URLconf:
17     1. Import the include() function: from django.urls import include, path
18     2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
19
20 """
21
22 from django.urls import path
23 from . import views
24
25 app_name = 'animals'
26 urlpatterns = [

```

(continues on next page)

(continued from previous page)

```
27 # path('', views.AnimalIndexView.as_view(), {'show':'current'}, name='animal-list
    ↳'),
28     path('', views.animal_list, name='animal-list'),
29 # path('archive', views.AnimalIndexView.as_view(), {'show':'archive'}, name=
    ↳'archive'),
30 # path('all', views.AnimalIndexView.as_view(), {'show':'all'}, name='all'),
31     path('claim/<int:primary_key>', views.claim, name='claim'),
32     path('send_email_animal', views.send_email_animal, name='send_email_animal'),
33     path('send_email_organ', views.send_email_organ, name='send_email_organ'),
34     path('<int:pk>', views.AnimalDetailView.as_view(), name='animal-detail'),
35     path('feed', views.LatestAnimalsFeed(), name='feed')
36 ]
```


Index

A

`age()` (animals.admin.AnimalAdmin method), 14
`age()` (animals.admin.OrganAdmin method), 14
`age()` (animals.models.Animal method), 18
`age()` (animals.models.Organ method), 18
`Animal` (class in animals.models), 18
`Animal.DoesNotExist`, 18
`Animal.MultipleObjectsReturned`, 18
`AnimalAdmin` (class in animals.admin), 14
`AnimalDetailView` (class in animals.views), 23
`AnimalForm` (class in animals.admin), 14
`AnimalIndexView` (class in animals.views), 23
`animals.admin` (module), 14
`animals.models` (module), 18
`animals.urls` (module), 29
`animals.views` (module), 23
`available()` (animals.models.Animal method), 18
`available()` (animals.models.Organ method), 18

C

`claim()` (in module animals.views), 23
`claim_organ()` (in module animals.views), 23
`clean()` (animals.admin.AnimalForm method), 14
`clear_claim()` (in module animals.admin), 15

D

`description()` (animals.models.Animal method), 18
`description()` (animals.models.Organ method), 18

F

`form` (animals.admin.AnimalAdmin attribute), 14

G

`get_absolute_url()` (animals.models.Animal method), 18
`get_absolute_url()` (animals.models.Organ method), 18
`get_queryset()` (animals.views.AnimalIndexView method), 23
`get_queryset()` (animals.views.OrganIndexView method), 23

I

`item_description()` (animals.views.LatestAnimalsFeed method), 23
`item_title()` (animals.views.LatestAnimalsFeed method), 23
`items()` (animals.views.LatestAnimalsFeed method), 23

L

`Lab` (class in animals.models), 18
`Lab.DoesNotExist`, 18
`Lab.MultipleObjectsReturned`, 18
`LabAdmin` (class in animals.admin), 14
`LatestAnimalsFeed` (class in animals.views), 23
`Location` (class in animals.models), 18
`Location.DoesNotExist`, 18
`Location.MultipleObjectsReturned`, 18
`LocationAdmin` (class in animals.admin), 14

M

`model` (animals.views.AnimalDetailView attribute), 23
`model` (animals.views.AnimalIndexView attribute), 23
`model` (animals.views.OrganIndexView attribute), 23

O

`Organ` (class in animals.models), 18
`Organ.DoesNotExist`, 18
`Organ.MultipleObjectsReturned`, 18
`OrganAdmin` (class in animals.admin), 14
`OrganIndexView` (class in animals.views), 23

P

`Person` (class in animals.models), 18
`Person.DoesNotExist`, 18
`Person.MultipleObjectsReturned`, 18
`PersonAdmin` (class in animals.admin), 14

R

`responsible_person()` (animals.models.Lab method), 18

S

`save_model()` (animals.admin.AnimalAdmin method), 14

`save_model()` (`animals.admin.OrganAdmin` method), [14](#)
`send_email_animal()` (in module `animals.views`), [24](#)
`send_email_organ()` (in module `animals.views`), [24](#)