
Django ZXCVCBN Password

Release 2.0.1

May 02, 2017

Contents

1	Django ZXCVCBN Password	1
1.1	License	1
1.2	Installation	1
1.3	Usage	1
1.4	Screen-shot	2
1.5	Development	2
2	Contributing	3
2.1	Bug reports	3
2.2	Documentation improvements	3
2.3	Feature requests and feedback	3
2.4	Development	4
3	Authors	5
4	Changelog	7
4.1	2.0.1 (2017-02-17)	7
4.2	2.0.0 (2017-02-17)	7
4.3	1.1.0 (2016-10-18)	7
4.4	1.0.5 (2015-03-31)	7
4.5	1.0.3 (2015-03-12)	8
4.6	1.0.2 (2015-03-12)	8
4.7	1.0.0 (2015-02-21)	8
4.8	0.1.0 (2015-02-01)	8
5	Indices and tables	9

Django Zxcvbn Password

Back-end and Front-end password validation with Zxcvbn.

A combination of [pirandig's django-zxcvbn](#) and [aj-may's django-password-strength](#) Django apps. It combines back-end and front-end validation with strength meter display.

License

Software licensed under [ISC](#) license.

Installation

```
pip install django-zxcvbn-password
```

Usage

```
# settings.py

AUTH_PASSWORD_VALIDATORS = [{
    'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator
    ↪',
}, {
    'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
}, {
    'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
}, {
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
}, {
    'NAME': 'zxcvbn_password.ZxcvbnValidator',
```

```
'OPTIONS': {
    'min_score': 3,
    'user_attributes': ('username', 'email', 'first_name', 'last_name')
}
}]
```

```
# forms.py

from django import forms
from zxcvbn_password.fields import PasswordField, PasswordConfirmationField

class RegisterForm(forms.Form):
    password1 = PasswordField()
    password2 = PasswordConfirmationField(confirm_with='password1')
```

Note: Remember to include `{% form.media %}` in your template. Please refer to the documentation of the two upstream repositories for more information.

By default, other inputs won't be used to compute the score, but you can enforce it like this:

```
# forms.py

from zxcvbn_password import zxcvbn

# in your form class
def clean():
    password = self.cleaned_data.get('password')
    other_field1 = ...
    other_field2 = ...

    if password:
        score = zxcvbn(password, [other_field1, other_field2])['score']
        # raise forms.ValidationError if needed

    return self.cleaned_data
```

Screen-shot

Development

To run all the tests: `tox`

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Documentation improvements

Django ZXCVCBN Password could always use more documentation, whether as part of the official Django ZXCVCBN Password docs, in docstrings, or even on the web in blog posts, articles, and such.

Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/Pawamoy/django-zxcvbn-password/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

Development

To set up *django-zxcvbn-password* for local development:

1. Fork *django-zxcvbn-password* (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/django-zxcvbn-password.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you’re done making changes, run all the tests with one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .  
git commit -m "Your detailed description of your changes."  
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)¹.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

¹ If you don’t have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.

It will be slower though ...

CHAPTER 3

Authors

- Timothee Mazzucotelli - <http://pawamoy.github.io/>

2.0.1 (2017-02-17)

- Fix call to super in PasswordStrengthInput.

2.0.0 (2017-02-17)

- Drop Django 1.8 support in favor of AUTH_PASSWORD_VALIDATORS setting introduced in Django 1.9.
- Update zxcvbn to more recent version (dwolfhub/zxcvbn-python on GitHub).
- Update JavaScript code to latest version.
- Remove all settings (they now go in AUTH_PASSWORD_VALIDATOR options).
- Change license to ISC.

Thanks to Nick Stefan and Daniel Wolf.

1.1.0 (2016-10-18)

- Cookiecutterize the project.

1.0.5 (2015-03-31)

- I don't remember.

1.0.3 (2015-03-12)

- Switch README to rst.
- Fix manifest rules.

1.0.2 (2015-03-12)

- Change package name from `django_zxcvbn_password` to `zxcvbn_password`.

1.0.0 (2015-02-21)

- Beta release on PyPI.

0.1.0 (2015-02-01)

- Alpha release on PyPI.

CHAPTER 5

Indices and tables

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