

---

# **django-auth-exchange Documentation**

*Release 0.1.5*

**Gregory N. Schmit**

**Apr 23, 2019**



---

## Contents:

---

<b>1</b>	<b>Backends</b>	<b>3</b>
<b>2</b>	<b>How to Use</b>	<b>5</b>
2.1	Settings . . . . .	5
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>



Version: 0.1.5



**class** `django_auth_exchange.backends.ExchangeAuthBackend`

This authentication backend uses `exchangelib` to authenticate against the Exchange or Office365 server.

By default, the `authenticate` method creates `User` objects for usernames that don't already exist in the database. Subclasses can disable this behavior by setting the `create_unknown_user` attribute to `False`.

**authenticate** (*request, username, password*)

Check for the format of the username (`dom\user` vs `user@dom`), then authenticate, and if successful, get or create the user object and return it.

Documentation: <https://django-auth-exchange.readthedocs.io>

Source: <https://github.com/gregschmit/django-auth-exchange>

PyPI: <https://pypi.org/project/django-auth-exchange/>

Auth Exchange is a reusable Django app that allows you to authenticate users against an Exchange/Office365 server (using `exchangelib`).

**The Problem:** I don't want users of my app to remember another password.

**The Solution:** This app allows those users to authenticate using their Exchange email credentials.





## CHAPTER 2

---

### How to Use

---

```
$ pip install django-auth-exchange
```

Include `django_auth_exchange` in your `INSTALLED_APPS`.

Add `django_auth_exchange.backends.ExchangeAuthBackend` to your `AUTHENTICATION_BACKENDS`, e.g.:

```
AUTHENTICATION_BACKENDS = [  
    'django_auth_exchange.backends.ExchangeAuthBackend',  
    'django.contrib.auth.backends.ModelBackend',  
]
```

Configure at least one domain:

```
AUTH_EXCHANGE_DOMAIN_SERVERS = {  
    'example.org': 'autodiscover',  
}
```

### 2.1 Settings

`AUTH_EXCHANGE_CREATE_UNKNOWN_USER` (default: `True`) - Determines if users should be created if they are not found in the local database.

`AUTH_EXCHANGE_DEFAULT_DOMAIN` (default: `'example.com'`) - If only a username is provided, this is the default domain that will be associated.

`AUTH_EXCHANGE_ALLOWED_FORMATS` (default: `['email', 'netbios', 'username']`) - This specifies which formats are allowed as the username (email means `someuser@example.com`, netbios means `EXAMPLE\someuser`, and username means `someuser`).

`AUTH_EXCHANGE_DOMAIN_SERVERS` (default: `{}`) - This specifies the domains which are allowed to authenticate and the server that should be used for authentication (or `'autodiscover'`). Hint: Office365 uses the server `outlook.office365.com`.

`AUTH_EXCHANGE_DOMAIN_USER_PROPERTIES` (default: `{}`) - This specifies the settings we should apply to a user when they are added to the local database for each domain (e.g., to make all `example.com` users superusers, `do: {'example.com': {'is_staff': True, 'is_superuser': True}}`).

`AUTH_EXCHANGE_NETBIOS_TO_DOMAIN_MAP` (default: `{}`) - This specifies a mapping from NETBIOS names to domain names.

## CHAPTER 3

---

### Contributing

---

Email [gschmi4@uic.edu](mailto:gschmi4@uic.edu) if you want to contribute. You must only contribute code that you have authored or otherwise hold the copyright to, and you must make any contributions to this project available under the MIT license.

To collaborators: don't push using the `--force` option.



## CHAPTER 4

---

### Indices and tables

---

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



**d**

`django_auth_exchange.backends`, 3





## A

`authenticate()` (*django\_auth\_exchange.backends.ExchangeAuthBackend* method), 3

## D

`django_auth_exchange.backends` (*module*), 3

## E

`ExchangeAuthBackend` (*class* in *django\_auth\_exchange.backends*), 3