

---

# **NewAuth Documentation**

*Release 0.2.1*

**adrien-f**

April 18, 2014



<b>1</b>	<b>Instructions</b>	<b>3</b>
<b>2</b>	<b>Dependencies</b>	<b>5</b>
<b>3</b>	<b>LDAP</b>	<b>7</b>
<b>4</b>	<b>Preview</b>	<b>9</b>
<b>5</b>	<b>Indices and tables</b>	<b>11</b>
5.1	OpenLDAP . . . . .	11
5.2	Settings . . . . .	12



**This is a work in progress, do not expect it to work on the first try or even do what you wish!**

NewAuth is an authentication platform and management for your Eve Online alliance. It uses a mix of LDAP (for access to various services) and MongoDB (to store additional informations).



---

## Instructions

---

Clone the repository and create a new virtual environment:

```
git clone https://github.com/J4LP/newauth
virtualenv .
pip install -r requirements.txt
```

Edit the settings:

```
cp auth/settings_dist.py auth/settings.py
```

Get the assets:

```
bower install
```

Build the assets:

```
python manage.py assets build
```

Launch:

```
python run.py
```

More detailed instructions for installing OpenLDAP can be found in *OpenLDAP*.





---

### Dependencies

---

This project requires a LDAP server to store user data for access to the forums, mumble, jabber, etc... It also requires a MongoDB server for groups storage, and additional user's informations storage. And of course, a Redis instance for caching purposes.



---

**LDAP**

---

To stay compatible with the old auth, it still uses the PIZZA LDAP schema. Installation instructions for this schema are located [here](#).



---

**Preview**

---



---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*

### 5.1 OpenLDAP

Installing NewAuth is a matter of inserting a new schema in OpenLDAP and configuring it.

You will need to have a running and configured OpenLDAP server on your network. On Ubuntu, it is a matter of installing `slapd` and `ldap-utils`, more instructions can be found on [Ubuntu's help](#).

NewAuth uses a LDAP schema used previously on Pizza-Auth (another Eve Online authentication portal) to stay backwards compatible. Please download it [here](#) and copy it in `/etc/ldap/schema`.

Create somewhere the file **schema\_convert.conf**:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/pizza.schema
```

Create a temporary ldif folder to store the generated files and run the conversion:

```
mkdir /tmp/ldif_output
slaptest -f schema_convert.conf -F /tmp/ldif_output
```

This will create a bunch of files in `/tmp/ldif_output`. Edit the file `/tmp/ldif_output/cn=config/cn=schema/{xx}pizza.ldif` and edit it so that the dn and cn lines look like this (j4lp can be anything else, like your alliance ticker):

```
dn: cn=j4lp,cn=schema,cn=config
cn: j4lp
```

And also remove the extra lines at the end of the file that look like this:

```
structuralObjectClass: olcSchemaConfig
entryUUID: 65f628a4-aa72-1032-9bfb-3d59b251971c
creatorsName: cn=config
createTimestamp: 20130905122822Z
entryCSN: 20130905122822.411617Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20130905122822Z
```

And finally, insert the new schema:

```
# You might need to sudo this command
ldapadd -Q -Y EXTERNAL -H ldapi:/// -f /tmp/ldif_output/cn=config/cn=schema/cn=\{xx\}pizza.ldif
```

Fantastic ! Your OpenLDAP server is now nearly ready for NewAuth, you just need to create a member directory. Create a file called `new_dn.ldif` and edit it with:

```
# Replace dc=nodomain by your OpenLDAP domain, something like dc=j4lp,dc=com for example
dn: ou=People,dc=nodomain
objectClass: organizationalUnit
ou: People

# Don't forget an extra line
```

Now that this is done, head on to

## 5.2 Settings

### `class auth.settings_dist.Config`

Config class for Newauth, this can be subclassed for different environments

Set the `NEWAUTH_ENV` environment variable to either `prod`, `dev`, or `test` for different configurations.

#### **SECRET\_KEY**

This is the flask secret key, you can easily generate one with `openssl rand -base64 64`

#### **EVE**

Dictionary to hold your alliance information.

Key	Description
<code>internal_mask</code>	The mask to use for verifying api keys for alliance members.
<code>alliance_id</code>	The alliance ID, it can easily be found on <a href="#">Dotlan</a>
<code>alliance_name</code>	The alliance name, if it's too long you can use the ticker

#### **LDAP**

OpenLDAP configuration.

Key	Description	Example	Environment
<code>server</code>	The server address	<code>localhost, 127.0.0.1</code>	<code>LDAP_HOST</code>
<code>port</code>	The server port	<code>389</code>	<code>LDAP_PORT</code>
<code>admin</code>	The admin user	<code>cn=admin,dc=nodomain</code>	<code>LDAP_ADMIN</code>
<code>password</code>	The admin's password		<code>LDAP_PASSWORD</code>
<code>basedn</code>	The base domain	<code>dc=nodomain,</code> <code>dc=j4lp,dc=com</code>	<code>LDAP_BASE</code>
<code>mem-berdn</code>	The organisation where the users live	<code>ou=People,dc=nodomain</code>	<code>LDAP_MEMBER</code>



**REDIS\_URL**

URL to the redis server, e.g., `redis://:password@localhost:6379/0`

**MONGODB\_SETTINGS**

MongoDB settings for MongoEngine

Key	Description	Default
DB	The mongo database name	auth
HOST	The mongo server address	127.0.0.1
PORT	The mongo server port	
USERNAME	The mongo server user if needed	
PASSWORD	The mongo server password if needed	