

```
manish@manish:~/redmine/superset/app$ cat superset_config.py
import os
from flask import Flask, redirect, url_for, session, jsonify
from flask_appbuilder import AppBuilder, expose, has_access
from flask_appbuilder.security.manager import AUTH_OID
from keycloak_security_manager import OIDCSecurityManager
from keycloak import KeycloakOpenID

# Initialize Flask App
app = Flask(__name__)
app.config.from_object(__name__)

# Get the current working directory
curr = os.path.abspath(os.getcwd())

# Flask AppBuilder configuration
app.config['SECRET_KEY'] =
'rWYzY7dyZARl//h3jl1iCVf978lWPZZNVIXQtBrMOWIUToNGFK4DcRLL'
app.config['AUTH_TYPE'] = AUTH_OID # Use OpenID Connect authentication

# Keycloak OIDC configuration
app.config['OIDC_CLIENT_SECRETS'] = os.path.join(curr, 'pythonpath', 'client_secret.json')
app.config['OIDC_ID_TOKEN_COOKIE_SECURE'] = False
app.config['OIDC_REQUIRE_VERIFIED_EMAIL'] = False
app.config['OIDC_OPENID_REALM'] = 'keenable'
app.config['OIDC_INTROSPECTION_AUTH_METHOD'] = 'client_secret_post'
app.config['CUSTOM_SECURITY_MANAGER'] = OIDCSecurityManager

# User registration settings
app.config['AUTH_USER_REGISTRATION'] = True
app.config['AUTH_USER_REGISTRATION_ROLE'] = 'Public'

# Superset-specific configuration
app.config['SUPERSET_WEBSERVER_DOMAINS'] = ["http://fosteringlinux.com",
"http://keenable.in"]
app.config['ENABLE_PROXY_FIX'] = True
app.config['SUPERSET_WEBSERVER_PROTOCOL'] = "http"
app.config['SUPERSET_WEBSERVER_PORT'] = 8088
app.config['PUBLIC_ROLE_LIKE'] = "Gamma"

# Initialize AppBuilder
appbuilder = AppBuilder(app)

# Keycloak Client Configuration
```

```

keycloak_openid = KeycloakOpenID(server_url="http://127.0.0.1:8080/auth/",
                                  client_id="keycloak-express",
                                  realm_name="keycloak-express",
                                  client_secret_key="long_secret-here")

# Example route to initiate login using Keycloak
@app.route('/login')
def login():
    # Get authorization URL
    auth_url = keycloak_openid.authorization_url(app.config['OIDC_CLIENT_SECRETS'],
    redirect_uri="http://127.0.0.1:3000/auth/callback")
    return redirect(auth_url)

# Example route for callback after Keycloak authentication
@app.route('/auth/callback')
def callback():
    # Handle callback and obtain tokens
    code = request.args.get('code')
    token = keycloak_openid.token(app.config['OIDC_CLIENT_SECRETS'], code,
    redirect_uri="http://127.0.0.1:3000/auth/callback")
    # Store token in session or use it as needed
    session['token'] = token
    return redirect(url_for('index'))

# Example protected route using AppBuilder
@app.route('/protected')
@expose('/protected')
@has_access
def protected():
    return "You are authorized to access this route!"

# Example logout route
@app.route('/logout')
def logout():
    # Perform logout actions, clear session, etc.
    session.clear()
    return redirect(url_for('index'))

# Example index route
@app.route('/')
def index():
    return "Welcome to My App!"

if __name__ == '__main__':

```

```
app.run(debug=True)
```

Different

```
import os
```

```
from flask import Flask
```

```
from flask_sqlalchemy import SQLAlchemy
```

```
from flask_appbuilder import AppBuilder
```

```
from flask_appbuilder.security.manager import AUTH_OID
```

```
from keycloak_security_manager import OIDCSecurityManager
```

```
# Initialize Flask App
```

```
app = Flask(__name__)
```

```
app.config.from_object(__name__)
```

```
# Get the current working directory
```

```
curr = os.path.abspath(os.getcwd())
```

```
# SQLAlchemy Database Configuration
```

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///mydatabase.db' # Example  
SQLite URI
```

```
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
```

```
# Flask AppBuilder configuration
```

```
app.config['SECRET_KEY'] =
```

```
'rWYzY7dyZARl//h3jl1iCVf978IWPZZNVIXQtBrMOWIUToNGFK4DcRLL'
```

```
app.config['AUTH_TYPE'] = AUTH_OID # Use OpenID Connect authentication
```

```
# Keycloak OIDC configuration
```

```
app.config['OIDC_CLIENT_SECRETS'] = os.path.join(curr, 'pythonpath',  
'client_secret.json')
```

```
app.config['OIDC_ID_TOKEN_COOKIE_SECURE'] = False
```

```
app.config['OIDC_REQUIRE_VERIFIED_EMAIL'] = False
```

```
app.config['OIDC_OPENID_REALM'] = 'keenable'
```

```
app.config['OIDC_INTROSPECTION_AUTH_METHOD'] = 'client_secret_post'
```

```
app.config['CUSTOM_SECURITY_MANAGER'] = OIDCSecurityManager
```

```
# Initialize SQLAlchemy
```

```
db = SQLAlchemy(app)
```

```
# Initialize AppBuilder
```

```
appbuilder = AppBuilder(app, db.session)
```

```
# Example route
```

```
@app.route('/')  
def index():  
    return "Hello World"  
  
if __name__ == '__main__':  
    app.run(debug=True)
```