Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт фундаментального образования

Кафедра ИИТ

**Доклад по дисциплине “Web - программирование”**

**Тема: “Безопасность”**

Выполнил(а): Клименко В.В. ,студент группы ФО-350005

Проверил(а): Свинцов Д.В.

Екатеринбург 2017

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc501780949)

[ВВЕДЕНИЕ 3](#_Toc501780950)

[АУНТЕФИКАЦИЯ И АВТОРИЗАЦИЯ 4](#_Toc501780951)

[ОПЕРАТОР ИМПОРТА И НАСТРАИВАЕМЫЙ КОРНЕВОЙ ЗАВОД 6](#_Toc501780952)

[АКСЕСС ЛИСТЫ (ACL) 8](#_Toc501780953)

# ВВЕДЕНИЕ

Pyramid предоставляет дополнительную, декларативную систему безопасности. В пирамиде система безопасности поделена на 2 части. Первая, это аутентификация, которая производит идентификацию пользователя, его проверку (например, что он есть в БД и он не заблокирован) и определяет какими правами он наделен. Второе, это авторизация, система, которая проверяет, имеет ли этот пользователь доступ к запрошенному ресурсу.

# АУНТЕФИКАЦИЯ И АВТОРИЗАЦИЯ

Простыми словами, [Аутентификация](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) — это проверка подлинности предъявленного пользователем идентификатора (например, проверки соответствия введённого пользователем пароля к учётной записи паролю в базе данных). Аутентификация требуется при доступе к таким интернет - сервисам как:

* [электронная почта](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%87%D1%82%D0%B0)
* [веб-форум](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%84%D0%BE%D1%80%D1%83%D0%BC)
* [социальные сети](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C_(%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82))
* [интернет-банкинг](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%B1%D0%B0%D0%BD%D0%BA%D0%B8%D0%BD%D0%B3)
* [платежные системы](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%91%D0%B6%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)
* [корпоративные сайты](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BE%D1%80%D0%BF%D0%BE%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D1%81%D0%B0%D0%B9%D1%82&action=edit&redlink=1)
* [интернет - магазины](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BC%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD).

Авторизация  — это предоставление определённому лицу или группе лиц прав на выполнение определённых действий, а также процесс подтверждения данных прав при попытке выполнения этих действий. Часто можно услышать выражение, что какой-то человек «авторизован» для выполнения данной операции — это значит, что он имеет на неё право. Иными словами, она производит контроль доступа легальных пользователей к ресурсам системы после успешного прохождения ими аутентификации. Зачастую процедуры аутентификации и авторизации совмещаются.

Pyramid имеет встроенные возможности аутентификации и авторизации, которые упрощают ограничение действий обработчика. Ниже приведен обзор шагов, которые вам, как правило, необходимо предпринять:

1. Создайте корневую фабрику в своей модели, которая связывает директивы allow / deny с группами и разрешениями
2. Создание пользователей и групп в вашей модели
3. Создайте функцию обратного вызова, чтобы получить список групп, на которые пользователь подписался, на основе их идентификатора пользователя
4. Сделайте «запрещенный вид», который будет вызываться, когда возникает исключение Forbidden.
5. Создайте действие для входа, которое будет проверять имя пользователя / пароль и помнить пользователя в случае успеха
6. Ограничьте доступ к действиям обработчика, передав аргумент permission = 'somepermission' @view\_config.
7. Соедините все это в своей конфигурации.

# ОПЕРАТОР ИМПОРТА И НАСТРАИВАЕМЫЙ КОРНЕВОЙ ЗАВОД

Пользовательский корневой завод генерирует объекты, которые будут использоваться в контексте запросов, отправленных в ваше веб-приложение. Первым атрибутом корневой фабрики является ACL или список управления доступом. Это список кортежей, которые содержат директиву для обработки запроса (например, «Разрешить» или «Запретить»), группе, которой предоставлен или запрещен доступ к ресурсу, а также разрешено (или необязательно кортеж разрешений) для этой группы.

Пример вышеприведенного списка контроля доступа указывает, что мы разрешим всем просматривать страницы с разрешением «всех», членами основной группы для просмотра страниц, ограниченных разрешением «запись», а также членов защищенной группы для просмотра страниц, ограниченных либо разрешения 'entry' или 'topsecret'. Специальный «Everyone» - это встроенная функция, которая позволяет любому человеку, посещающему ваш сайт (известный как основной), получить доступ к данному ресурсу.

Чтобы пользователь мог войти в систему, вы можете создать обработчик, который проверяет логин и пароль (или любые дополнительные критерии), представленные через форму. Обычно вам нужно добавить следующие импорты:

После того, как вы подтвердите логин и пароль пользователя в отношении модели, вы можете настроить заголовки «запомнить» идентификатор пользователя, а затем вы можете перенаправить пользователя на главную страницу или URL-адрес, к которому они пытались получить доступ:

Также, следует обратить внимание, что в вызове функции запоминания мы передаем идентификатор пользователя, который мы извлекли из базы данных и сохранили в переменной 'userid'.

Чтобы зарегистрировать пользователя, вы «забудете» их и используйте HTTPFound для перенаправления на другой URL:

Прежде чем ограничивать действие обработчика с помощью разрешения, вам понадобится функция обратного вызова, чтобы вернуть список групп, к которым принадлежит идентификатор пользователя. Вот один из способов реализовать его в вашей модели, в этом случае предполагая, что у вас есть объект Group с атрибутом groupname и объект Users с отношением mygroups к группам:

Например, теперь вы можете импортировать и использовать декоратор @action для ограничения по разрешению и authenticated\_userid для извлечения идентификатора пользователя из запроса, что дает нам очень простой способ ограничить действия обработчика, а также получить информацию о пользователе.

Последний шаг - настроить \_\_init\_\_.py для использования вашей политики аутентификации. Обязательно добавьте эти импорты:

В своей основной функции вы хотите определить свои политики аутентификации, чтобы вы могли включить их в вызов Configurator:

Возможности аутентификации и авторизации в Pyramid очень легко начать с использования Pylons и repoze.what. Преимуществом является упрощение поддержки кода и встроенных методов для решения общих задач, таких как запоминание или забывание пользователей, настройка разрешений и легкое изменение обратного вызова groupfinder для работы с вашей моделью.

# АКСЕСС ЛИСТЫ (ACL)

Access Control List или [ACL](http://lectures.uralbash.ru/6.www.sync/3.framework/pyramid/glossary.html#term-acl) — список контроля доступа, который определяет, кто или что может получать доступ к конкретному объекту, и какие именно операции разрешено или запрещено этому субъекту проводить над объектом.

В пирамиде аксесс лист это список содержащий записи, определяющие права индивидуального пользователя или группы на ресурсы проекта. Элементы ACL также еще называют Access Control Entry или [ACE](http://lectures.uralbash.ru/6.www.sync/3.framework/pyramid/glossary.html#term-ace).