**Тема занятия № 52: Модуль 30. Разработка Web-служб REST. REST framework. Django REST Framework**

1. Разграничение доступа в Django REST framework

**Третий принцип REST:**

Данные клиента хранятся на стороне клиента в сайтах, построенных по традиционной архитектуре, данные клиента, включая признак, выполнил ли пользователь вход на сайт, хранятся на стороне сервера. Благодаря этому сервер всегда может проверить, выполнил ли текущий пользователь вход на сайт или он является гостем.

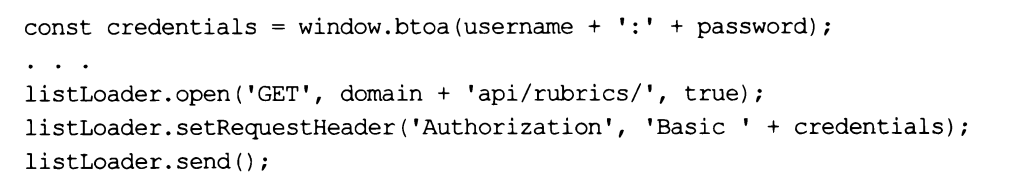
Но в веб-службах REST данные клиента должны храниться на стороне клиента.

Фронтенд должен сохранить введенные пользователем имя и пароль: в локальном хранилище DOM, cookie или в обычных переменных.

Чтобы бэкенд смог проверить, имеет ли текущий пользователь права на доступ к данным, фронтенд должен посылать ему сохраненные имя и пароль в каждом запросе. Они записываются в заголовке Authorization запроса в виде строки формата Basic <имя и пароль>, где имя и пароль представляются в формате <имя>; <пароль> и кодируются в кодировке base64. Для кодирования можно использовать метод btoa (), поддерживаемый классом window.

Вот пример кода фронтенда, отправляющего имя и пароль пользователя бэкенду:





Таким образом реализуется основная аутентификация (basic authentication), при которой в каждом клиентском запросе бэкенду пересылаются непосредственно имя и пароль пользователя.

**Классы разграничения доступа**

Чтобы в коде бэкенда указать, какие пользователи имеют доступ к определенному контроллеру, следует задать у этого контроллера набор необходимых классов разграничения доступа.

Классы разграничения доступа объявлены в модуле rest framework.permissions. наиболее часто используемые из них приведены далее:

О aiiowany— разрешает доступ к данным всем— и зарегистрированным пользователям, и гостям. Установлен по умолчанию;

□ isauthenticated— разрешает доступ к данным только зарегистрированным пользователям;

□ isauthenticatedorreadoniy— предоставляет полный доступ к данным только зарегистрированным пользователям, гости получат доступ лишь на чтение;

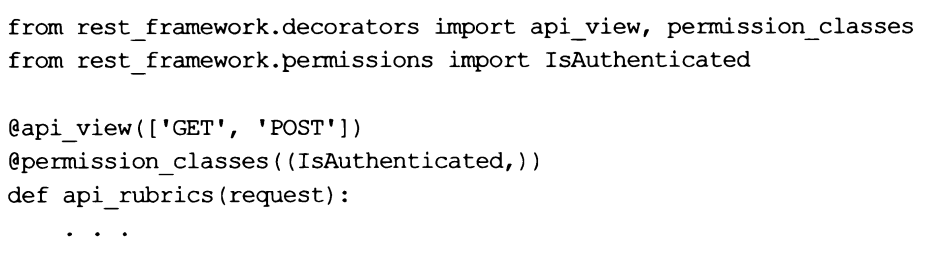
□ isadminuser — разрешает доступ к данным только зарегистрированным пользователям со статусом персонала;

□ djangomodei Permissions— разрешает доступ к данным только зарегистрированным пользователям, имеющим необходимые права на работу с этими данными (о правах пользователей рассказывалось ранее);

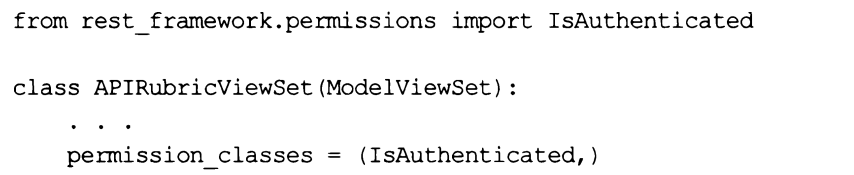
□ djangomodelpermissionsoranonreadonly— предоставляет ПОЛНЫЙ доступ к данным только зарегистрированным пользователям, имеющим необходимые права на работу с ними. Все прочие пользователи, включая гостей, получат доступ только на чтение.

Перечень классов разграничения доступа записывается в виде кортежа. Указать его можно:

□ В контроллере-функции — С ПОМОЩЬЮ декоратора permission\_classes (<перечень классов>) ИЗ модуля rest\_framework. Decorators:

****

Перечень классов разграничения доступа, используемых по умолчанию, указывается в настройках проекта (в модуле settings.py пакета конфигурации) следующим образом:

****

Чтобы пользователь получил доступ к данным, он должен ’’пройти” через все классы разграничения доступа, указанные в перечне. Так, если указать классы Isadminuser И djangomodelpermissions, ТО доступ К данным получат ТОЛЬКО пользователи со статусом персонала, имеющие права на работу с этими данными.