<https://ilyachch.gitbook.io/django-rest-framework-russian-documentation/overview/navigaciya-po-api/views>

Классы APIView имеют следующие отличиия от обычных классов View:

* Запросы, переданные обработчику будут экземплярами Request REST framework, а не джанговских HttpRequest.
* Обрабатыващие методы могут возвращать Response REST framework вместо джанговских HttpResponse. Представление будет осущестлвять согласование содержимого и устанавливать нужный рендерер для ответа.
* Любые исключения APIException будут выявлены и связаны с соответствующими ответами.
* Входящие запросы будут подтверждены и соответствующее разрешение и/или проверки будут произведены перед тем как передать запрос на обработку.

from rest\_framework.views import APIView

from rest\_framework.response import Response

from rest\_framework import authentication, permissions

from django.contrib.auth.models import User

class ListUsers(APIView):

"""

View to list all users in the system.

\* Requires token authentication.

\* Only admin users are able to access this view.

"""

authentication\_classes = (authentication.TokenAuthentication,)

permission\_classes = (permissions.IsAdminUser,)

def get(self, request, format=None):

"""

Return a list of all users.

"""

usernames = [user.username for user in User.objects.all()]

return Response(usernames)

**Атрибуты API**

Следующие атрибуты отвечают за подключаемые аспекты представлений API.

* .renderer\_classes
* .parser\_classes
* .authentication\_classes
* .throttle\_classes
* .permission\_classes
* .content\_negotiation\_class

**Методы реализации API**

Следующие методы используются в REST framework для реализации различных подключаемых компонентов API. Как правило эти методы не заменяются.

* .get\_renderers(self)
* .get\_parsers(self)
* .get\_authenticators(self)
* .get\_throttles(self)
* .get\_permissions(self)
* .get\_content\_negotiator(self)
* .get\_exception\_handler(self)

**Методы применения компонентов API**

Следующие методы вызываются до обрабатывающих методов.

* .check\_permissions(self, request)
* .check\_throttles(self, request)
* .perform\_content\_negotiation(self, request, force=False)

@api\_view(http\_method\_names=['GET'])

Каждое общее представление строится путем комбинации GenericAPIView с одним из классов-миксинов.

<https://ilyachch.gitbook.io/django-rest-framework-russian-documentation/overview/navigaciya-po-api/generic-views#miksiny>

ListModelMixin

CreateModelMixin

RetrieveModelMixin

UpdateModelMixin

DestroyModelMixin