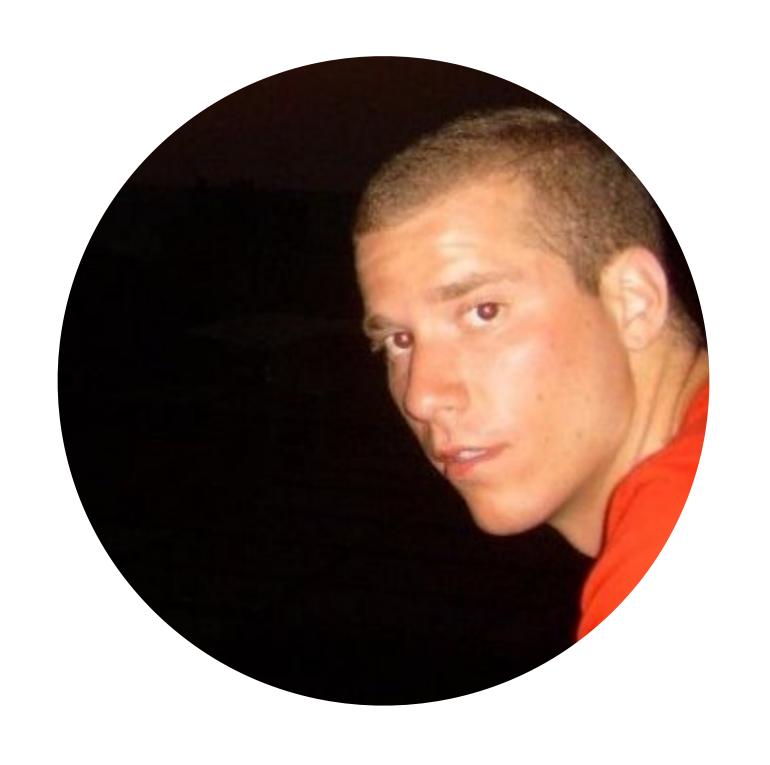


Django Rest Framework - Tips & Tricks

#### About Me



- Backend developer @ Cloud Academy
- Working in Python since 2008
- Using Django since version 1.3
- In love with Cloud Computing

@luca\_zack



## Agenda

- Introduction
  - What is DRF?
  - Can I trust it?
- DRF main advantages
- DRF main concepts
- Demo time
- · Q&A

#### Introduction

## What is Django Rest Framework?

Django REST framework is a powerful and flexible toolkit for building Web APIs.

#### Introduction

#### Can I trust it?

Excellent documentation, active community and widely adopted.

- Django Rest Framework: 7,670 stars and 2,600 forks on GitHub
- **Django Tastypie**: 3,160 stars and 1,100 forks on GiHub

## Django rest Framework - Main Advantages

- Easy to install and integrate
- Browsable interface
- Excellent documentation
- Write your favorite view types: supports both FBVs and CBVs
- Easy Serialization for both ORM and non-ORM data sources.
- Authentication and authorization
- Pagination and filtering
- Fully customizable

### Request / Response

- request.POST —> request.data
  - Handles PATCH and PUT requests
  - Includes both POST and FILES data
  - Automatic parsing of incoming data
- request.GET —> request.query\_params
- DRF Response automatically renders the response into the requested content type.

#### Parsers / Renderers

- Parsers are used to accept and decode requests in various content types
  - Parser is determined by checking the Content-Type header
  - JSON, Form, Multipart, FileUpload and custom
  - YAML, XML, JSONP and many others are available using third party packages
- Renderers are used to return responses in various content types
  - Renderer is determined on the basis of Accept request header
  - JSON, static/rendered HTML and custom
  - YAML, XML, JSONP and many others are available using third party packages

#### Serializers

- Used to translate objects and query sets into JSON/XML or any other supported content type
- Can be declared from scratch or automatically generated from models
- Validates incoming data before creation/update
- ModelSerializer supports model fields and custom ones
- Allows nested serializations

#### **Views**

- Function based views using DRF @api\_view decorator
- Class based views using DRF APIView
- Ready to use mixins and generic class based views
- Viewsets, Generic Viewsets and Model Viewsets

## Authentication, Permissions and Throttling

- Authentication identifies who is making a request
- Permissions define which resources the identified user can access
- Throttling defines the amount of requests than clients can make

## **Pagination & Filtering**

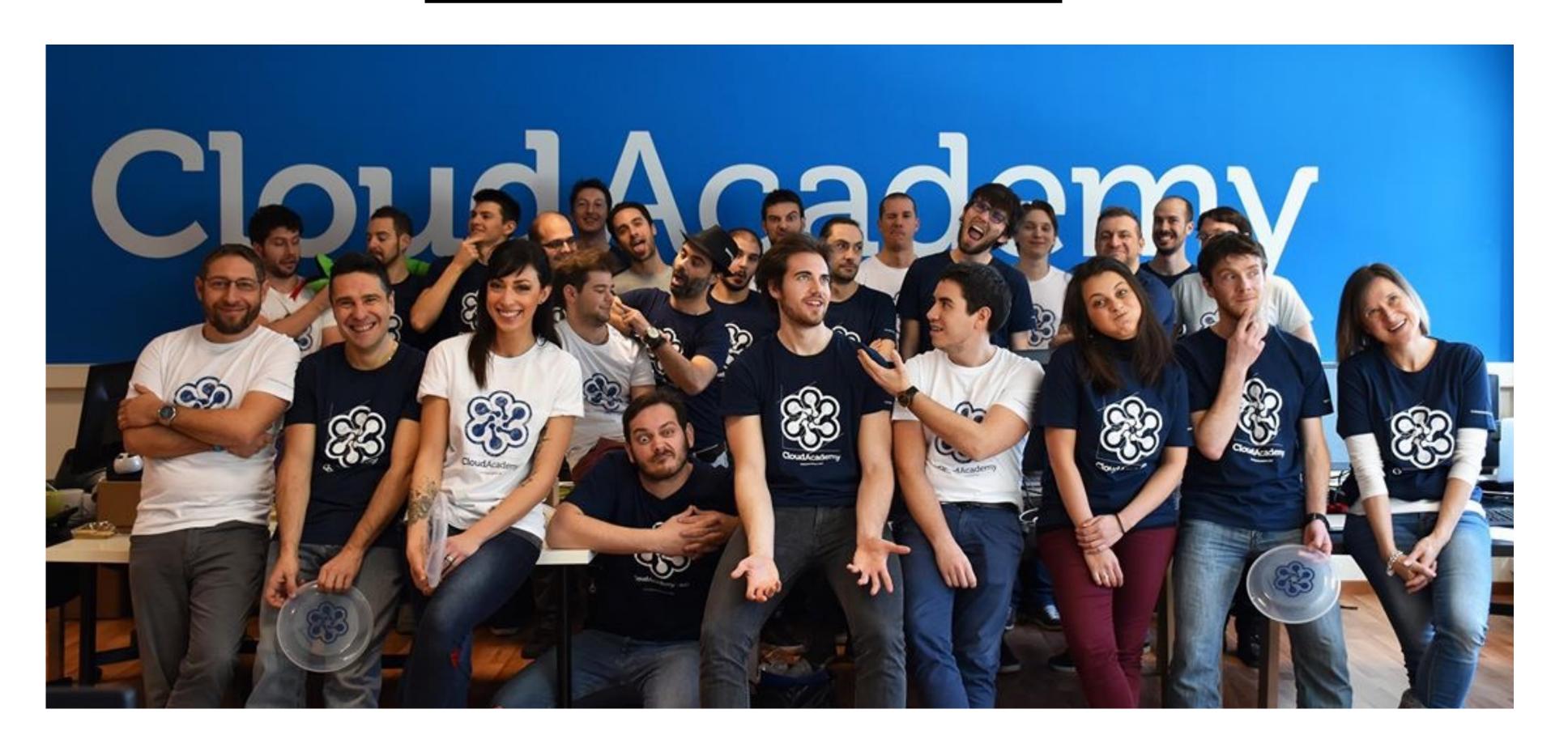
- Pagination is in charge to split requested resources into individual pages of data
- Filtering limits the resources based on one or more criteria.

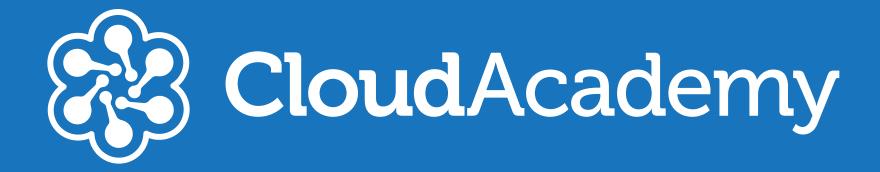


## Demo time

## We are Hiring

## clda.co/fullstack-dev





# Thank you

Q & A