Rest verb /Index /Type /Api ====> Get /myindex/mytype/\_search

Rest verb ===> Get ,Post ,Put , Delete

ساخت index جدید :

PUT /product?pretty

فلگه pretty این امکان را به ما میدهد که در postman و curl پاسخ حالت json داشته باشد اما در kibana لازم نیست نوشته شود

اضافه کردن documentها به index :

post /product/default/1

{

"name":"Elastic Search for everybody",

"intructor":{

"firstName":"ahmadali",

"lastNamr":"moshtaghi"

}

}

عدد 1 که جلوی request نوشته شده است به عنوان id برای document در نظر گرفته میشود یعنی با id 1 میتوانیم به این document دسترسی پیدا کنیم . اما اگر گذاشته نشود بصورت پیشفرض یک id به این document داده میشود

Get کردن یک document با id:

GET /product/default/1

برای آپدیت کردن یک document دو روش وجود دارد :

1)

POST /product/default/1/\_update

{

"doc":{

"price":120000,

"tag":["online\_cource"]

}

}

2)

PUT /product/default/1

{

post /product/default/1

{

"name":"Elastic Search for everybody",

"intructor":{

"firstName":"ahmadali",

"lastNamr":"moshtaghi"

}

"price":20000

}

}

آپدیت کردن اسکریپتی:

POST /product/default/1/\_update

{

"script":"ctx.\_source.price += 9000"

}

کوئری upsert:

POST /product/default/1/\_update

{

"script":"ctx.\_source.price += 100",

"upsert":{

"price":5

}

}

این قسمت از کد به این مفهوم است که اگر index product وجود داشت و در این index

Price وجود داشت آن را 100 تا اضافه کن اما اگر وجود نداشت price ساخته شود و مقدارش 5 باشد

پاک کردن document:

1)

DELETE /product/default/1

2)

POST /product/\_delete\_by\_query

{

"query":{

"match":{

"price":105

}

}

}

در روش دوم با استفاده از query داکیومنت را پاک کردیم به این صورت که اگر documentی پیدا شد که field price برابر با 105 بود آن را پاک کن

Delete index:

DELETE /product/

Delete کردن یک عمل برگشت ناپذیر میباشد

\_bulk:

با استفاده از این api که ELK search به ما میدهد میتوانیم چندین کار(عمل یا operation) را با یک request به سمت ELK search بفرستیم مثلا :

POST /product/default/\_bulk

{"update":{"\_id":"100"}}

{"doc":{"price":1100}}

{"delete":{"\_id":"101"}}

دقت شود که بعد از \_bulk حتما باید نوع operationی که قرار است انجام شود نوشته شود مثلا :update , delete , index, …..

Import کردن documentها با دستور curl :

C:\Users\Ahmad Ali\Desktop

curl -H "Content-Type: application/json" -XPOST "http://localhost:9200/product/\_bulk?pretty" --data-binary "@products-bulk.json"

دقت شود که products-bulk.json اسم فایلی است که documentها در آن قرار دارد و فایل دقیقا در مسیر ترمینال است یعنی فایل در Desktop قرار دارد

نشان دادن انواع وضعیت ها :

GET /\_cat/health?v

تعیین وضعیت cluster

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

GET /\_cat/nodes?v

تعیین وضعیت node ها

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

GET /\_cat/indices?v

تعیین وضعیت index ها

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

GET /\_cat/allocation?v

تعیین وضعیت shardها(allocate , unassigned) و disk space

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

GET /\_cat/shards?v

تعیین وضعیت تک تک shard ها

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

GET /\_cluster/allocation/explain

توضیح کامل درمورد allocationها

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

curl -X PUT "localhost:9200/\_cluster/settings?pretty" -H "Content-Type: application/json" -d" {\"transient\" :{\"cluster.routing.allocation.enable\" : \"all\"}}"

با این دستور در ویندوز (به دلیل \”) به الستیک سرچ این اجازه رو میدهیم که replicaی index در همان nodeی که primary index وجود دارد ایجاد شود

Mapping: این مفهوم به این معنی میباشد که ما به elastic search میگوییم که دیتا هایمان از چه نوع data typeی میباشد

هروقت که ما دیتایی را به document الستیک import میکنیم به صورت dynamic mapping اینکار صورت میگرد یعنی خوده الستیک تشخیص میدهد که هر کدام از چه نوع typeی میباشد

Meta fields:

\_index: نام indexی هست که اون document بهش تعلق دارد

\_id: id اون document

\_source: اولین json فایلی هست که ما به این document وارد کردیم (insert)

\_version: بعد از هر تغییری یک واحد اضافه میشود

\_field\_name: هر فیلدی که non-null value داشته باشد

\_meta: یک سری از دیتا هایی که در اپلیکشن وجود دارد بخواهیم در \_meta دخیره کنیم مثلا توضیحی در مورد index یا ذخیره کردن یک user و pass خاص

Field data type:

بر روی text data type آنالیز صورت میگیرد اما بر روی keyword data type آنالیز صورت نمیگیرد و برای aggregation و filtering استفاده میشوند

اضافه کردن mapping:

PUT /product/\_mapping

{

"properties" :{

"discount":{

"type":"double"

}}}

با این دستور به index product ویژگی (properties) جدیدی به نام discount از نوع double ایمپورت میشود

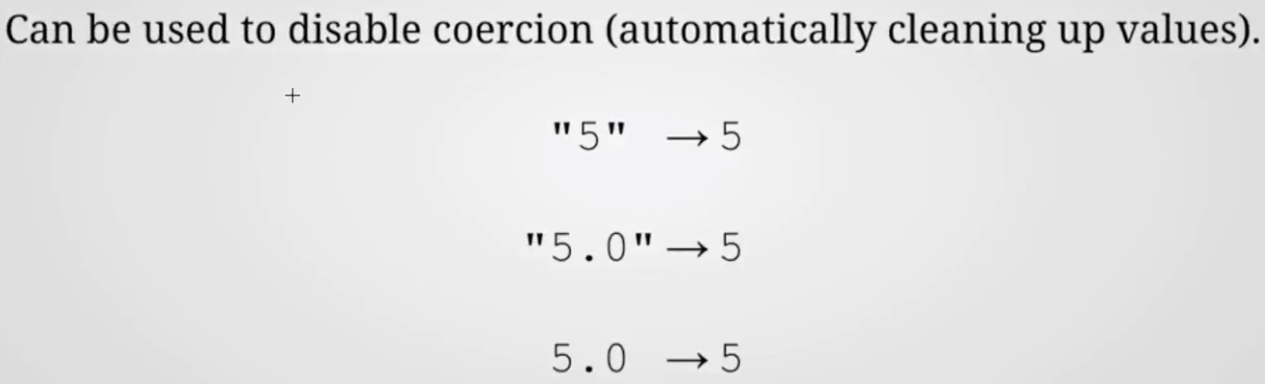
یک نکته ی بسیار مهم اینکه وقتی mappingی به index اضافه شود دیگر به هیچ عنوان نمیتوان mapping را تغییر داد به جز دوحالت :1) اضافه کردن فیلد هایی به

Data type object میباشد 2) additional mapping: data type keyword در واقع additional mapping است و میتواند به فیلدی با type text اضافه شود

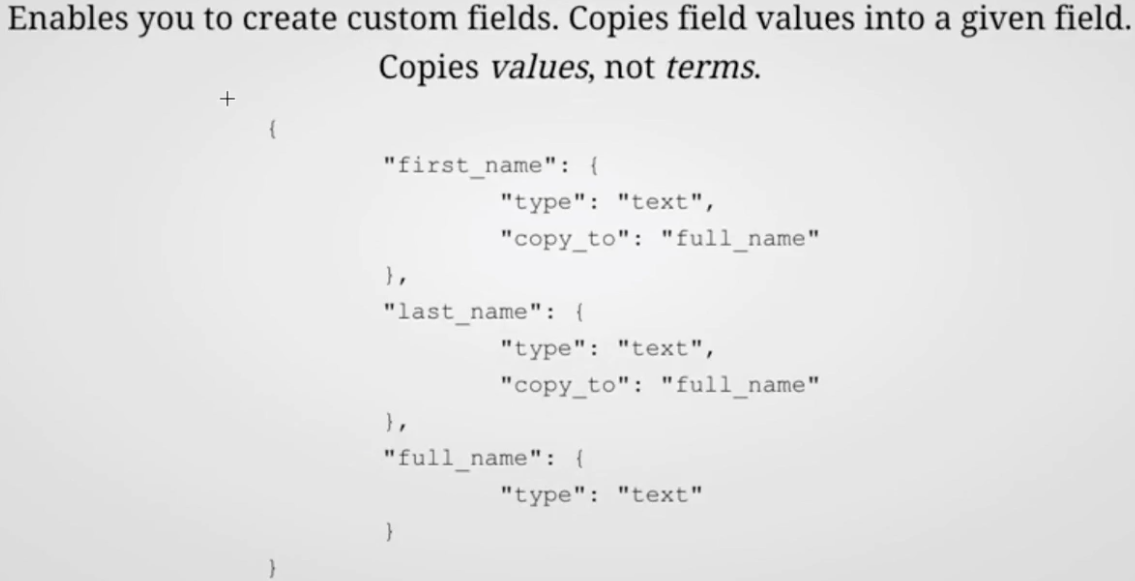
اما اگر از این دو مورد نبود راهکار تغییر دادن این است که index به کلی پاک شود و index جدید با mapping جدید ساخته شود

Mapping parameters:

Coerce :

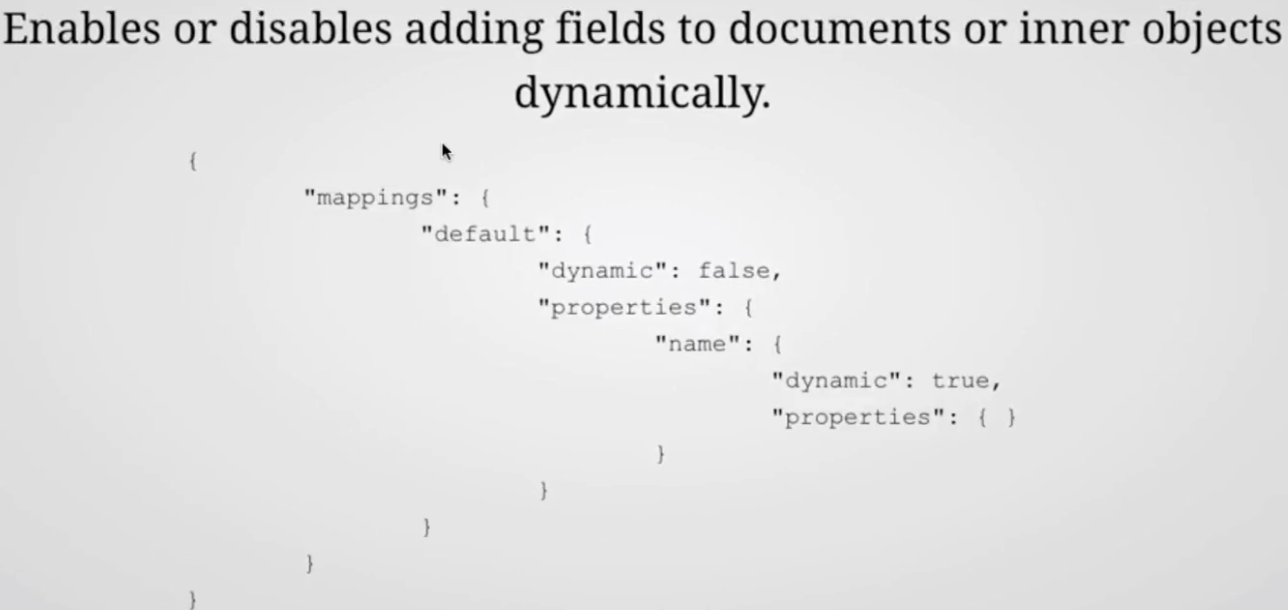


Copy\_to:



نکته مهم اینکه در \_source که فیلدهایمان را نمایش میداد نشان داده نمیشود

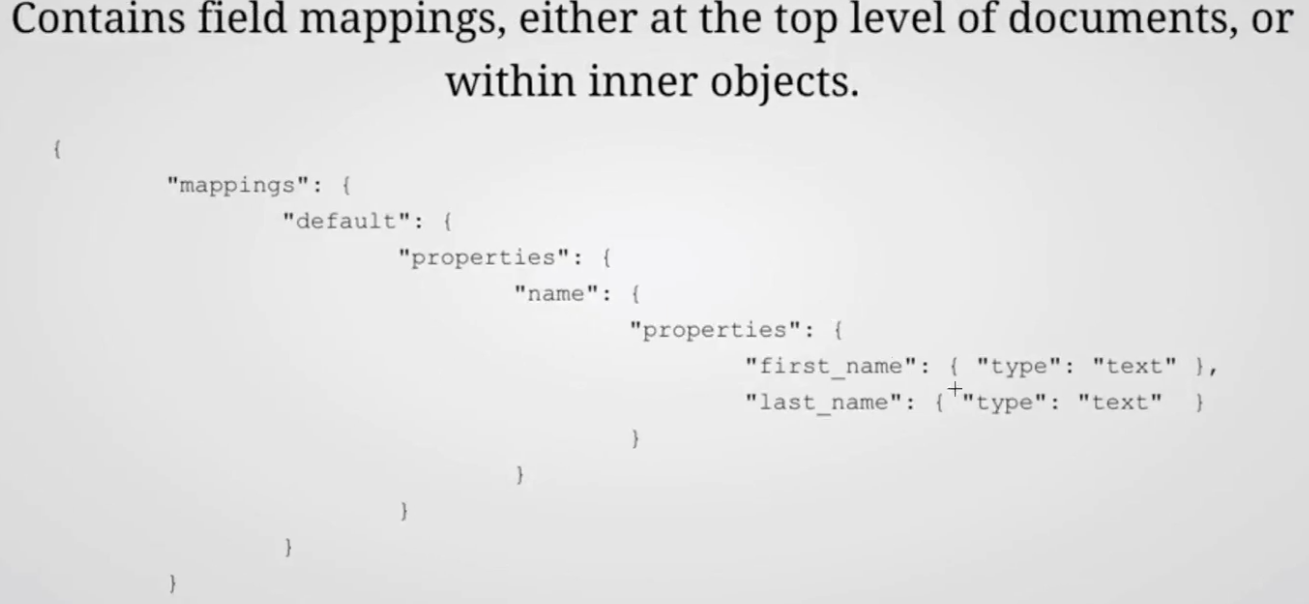
Dynamic :



Dynamic 3 مقدار دریافت میکند false , true , strict

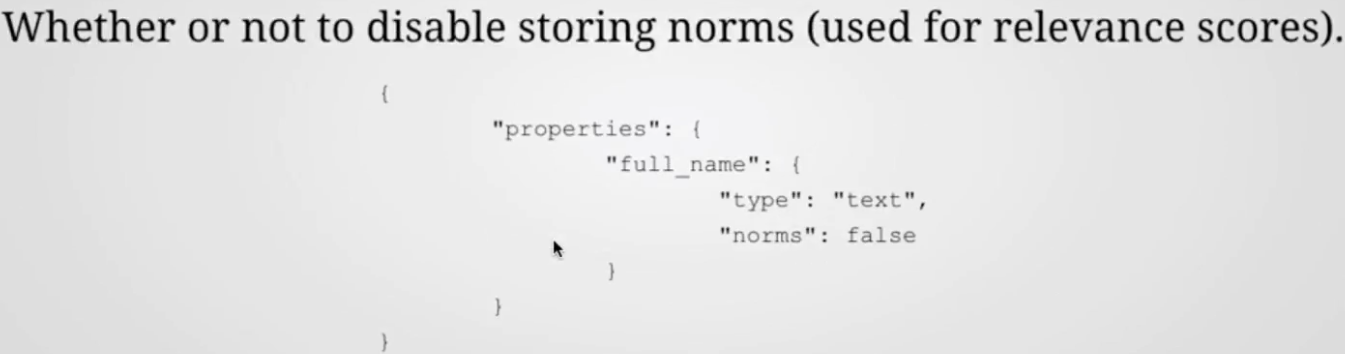
Strict به این معنی است که سخت گیرانه رفتار میکند مثلا حق نداریم به فیلدی که مقدارش integer میباشد مقدار double وارد کنیم یا ......

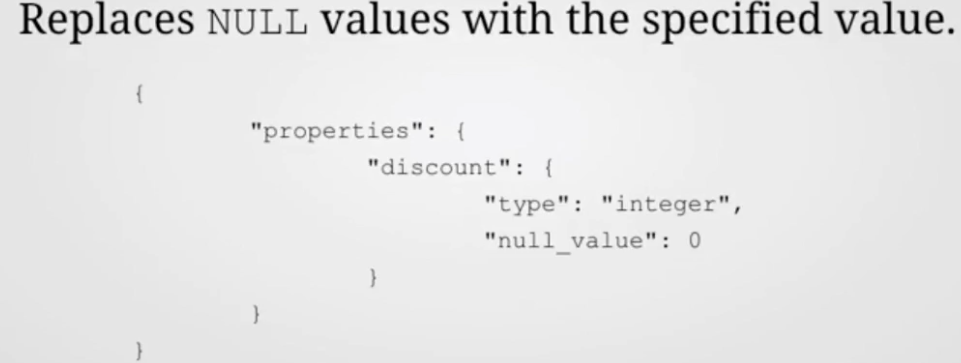
Properties:



نام دیگر fields میباشد

Norms :



Null value: 

Reindex(دوباره درج کردن ):

برای این کار elk search یک api در اختیار ما قرار میدهد که باعث میشود تمام documentهایی که در index وجود دارند reindex شوند:

POST /product/\_update\_by\_query?conflicts=proceed

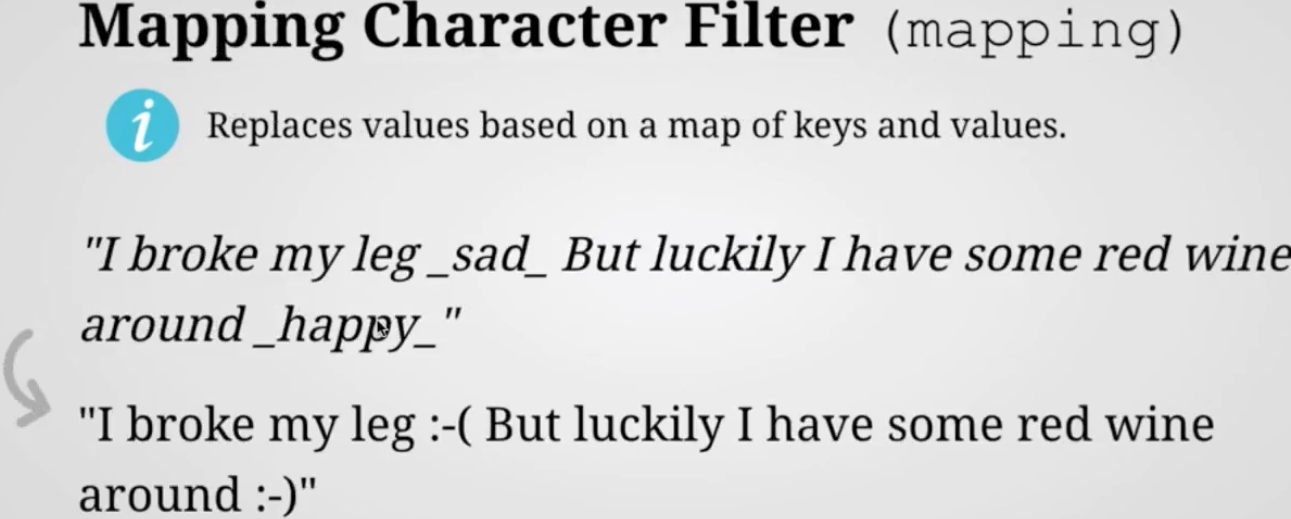
Confilicts=proceed به این معنی میباشد که هنگام reindex کردن documentها به confilict برخورد کردی آن را نادیده بگیر و ادامه ی کار را انجام بده

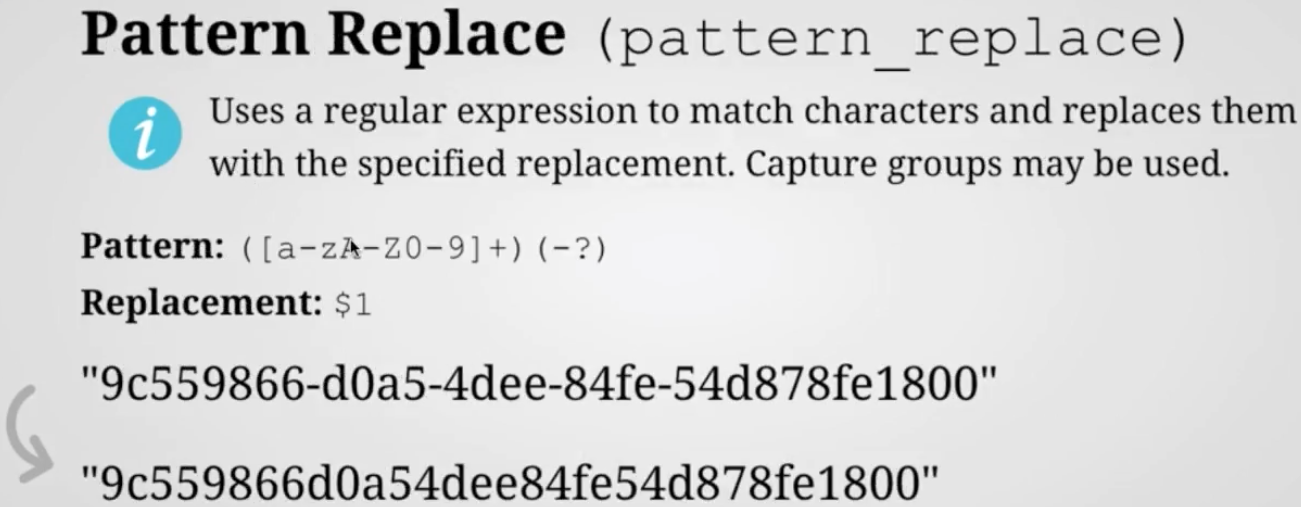
Analysis:

اولین نکته این است که آنالیز بر روی full text انجام میشود و بر روی keyword ها آنالیز صورت نمیگیرد

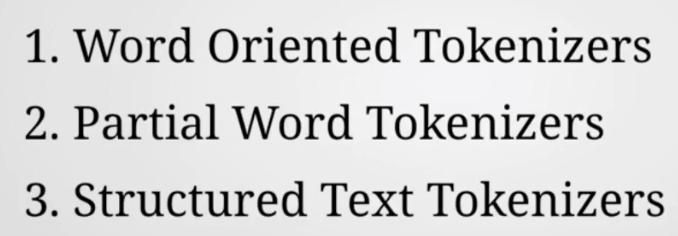
Character filter:



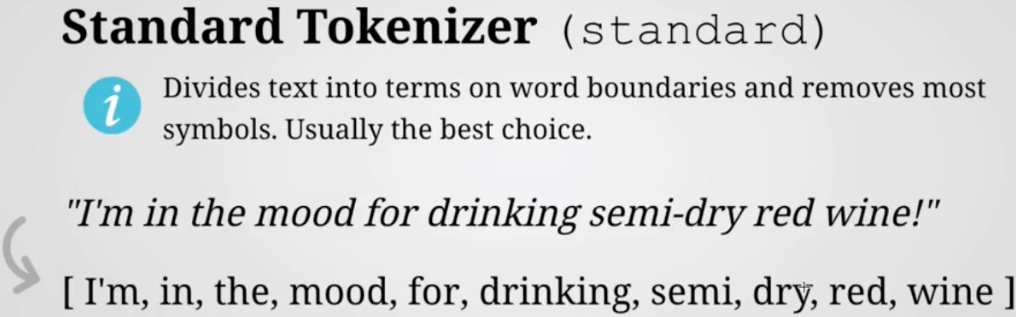


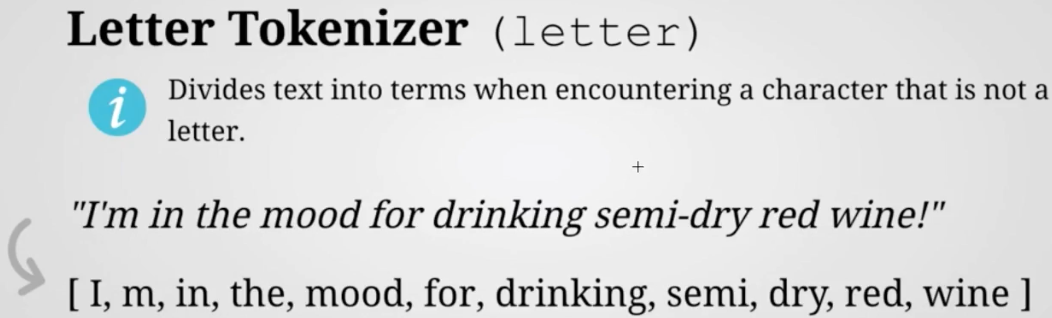


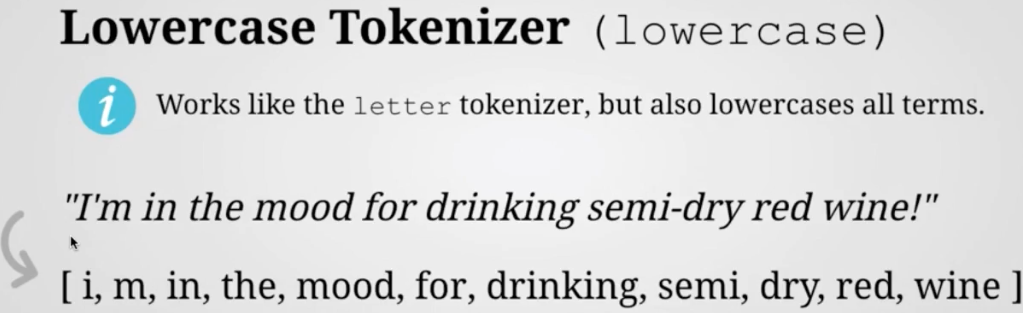
Tokenizer:

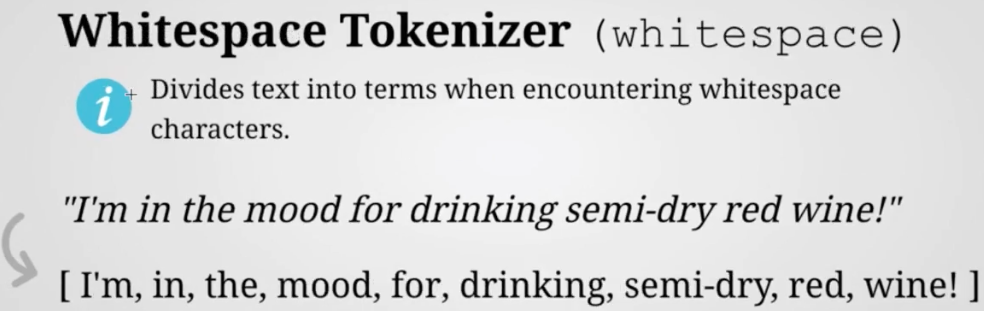


Word oriented tokenizers:





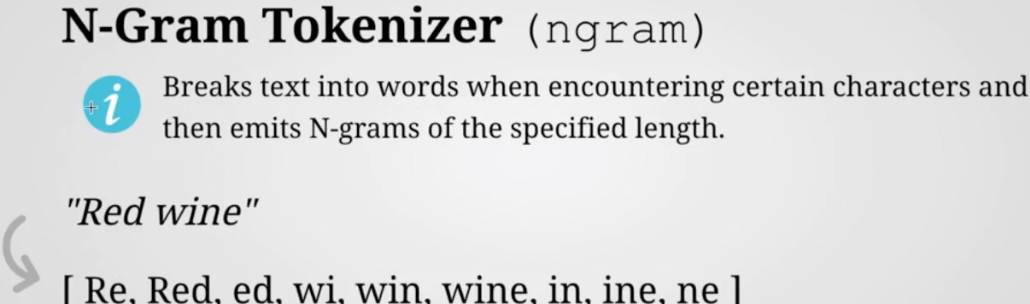


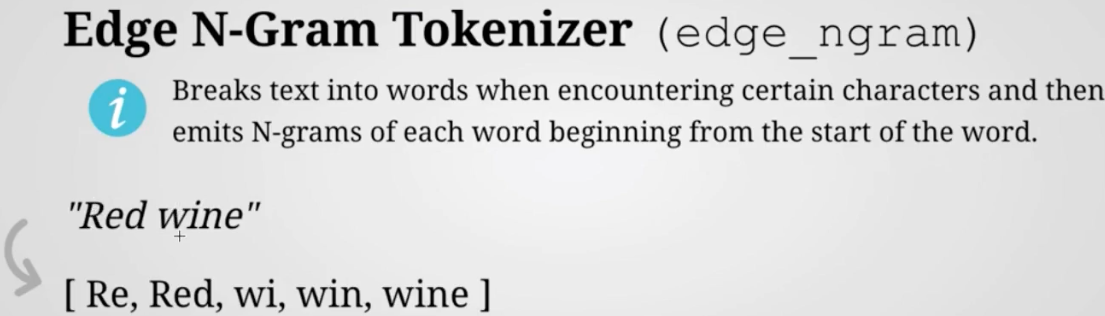




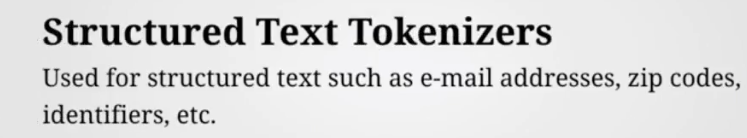
Partial word tokenizers:

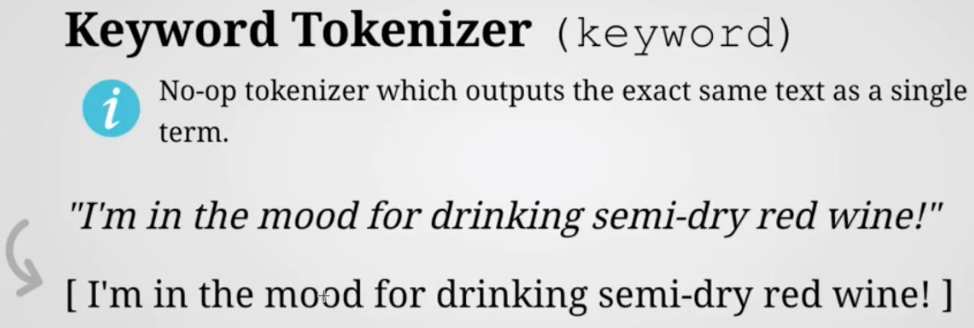




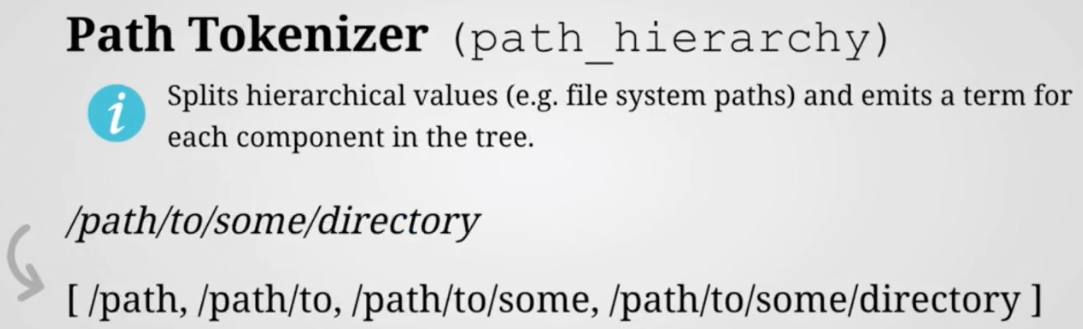


Structured text tokenizers:









Token filter:



