VANDEPOELE Enzo | MICALLEF John | AHOUANDOGBO Amen

NoSQL

Mini projet – Minecraft

**Pour mettre en place le projet via Docker, vous pouvez suivre le guide dans le fichier README.md**

Requêtes MongoDB :

## Trouver l’item qui s’appelle « épée en diamant »

db.itemBlock.find({ name: "diamond\_sword" }).pretty()

Cela retourne tous les items qui s’appelle « épée en diamant »

## Trouve tous les items craftables avec du diamant

db.itemBlock.find({

"recipes.crafting.1": {

$elemMatch: {

$elemMatch: {

$elemMatch: {

$in: ["diamond", "minecraft:diamond"]

}

}

}

}

})

Cela retourne tous les items qui on besoin de diamant dans leur craft.

## Trouver les objets qui ne se stackent pas à 1 ou 64

db.itemBlock.find({

stack\_limit: { $nin: [1, 64] }

}).pretty()

Cela retourne tous les items qui se stack pas en 1 ou 64 donc par exemple 16.

## Trouver les items contenant "wood" et taggés "doors"

db.items.find({

name: /wood/i,

item\_tags: { $in: ["doors"] }

})

Cela retourne tous les items qui sont des portes en bois

## Trouver tous les blocks qui peuvent bouger

db.items.find({

"block\_data.blocks\_movement": true

})

Cela retourne tous les items qui peuvent bouger

## Compter les documents contenant du « iron »

db.items.countDocuments({

name: { $regex: "iron", $options: "i" }

});

Cela retourne le nombre d’item qui on « iron » dans leur nom

## Chercher tous les blocks

db.items.find({ is\_block: true })

Cela retourne tous les items qui sont des blocs

## Trouver les Items non craftables

db.items.find({

$or: [

{ recipes: { $exists: false } },

{ "recipes.crafting": { $exists: false } },

{ "recipes.crafting": { $size: 0 } }

]

})

Cela retourne tous les items non craftables

## Ajouter l’item "frite"

db.items.insertOne({

is\_block: false,

item\_tags: [

"c:foods/vegetables",

"c:foods"

],

name: "frite",

recipes: {

crafting: [

[

[

"frite",

64,

'{count:64,id:"minecraft:frite"}'

],

[

[

[

"iron\_sword",

1,

'{count:1,id:"minecraft:iron\_sword"}'

]

],

[

[

"potato",

8,

'{count:8,id:"minecraft:potato"}'

]

]

],

[

"shapeless"

]

]

]

},

stack\_limit: 64

})

Cela crée l’item « frite »

## Modifier la « hardness » d’un item

db.items.findOne({ name: "acacia\_fence\_gate" })

Cela modifie la solidité de l’item « acacia\_fence\_gate »

Requêtes ElasticSearch :

## Trouver tous les items contenant le tag « boats »

GET items/\_search

{

"query": {

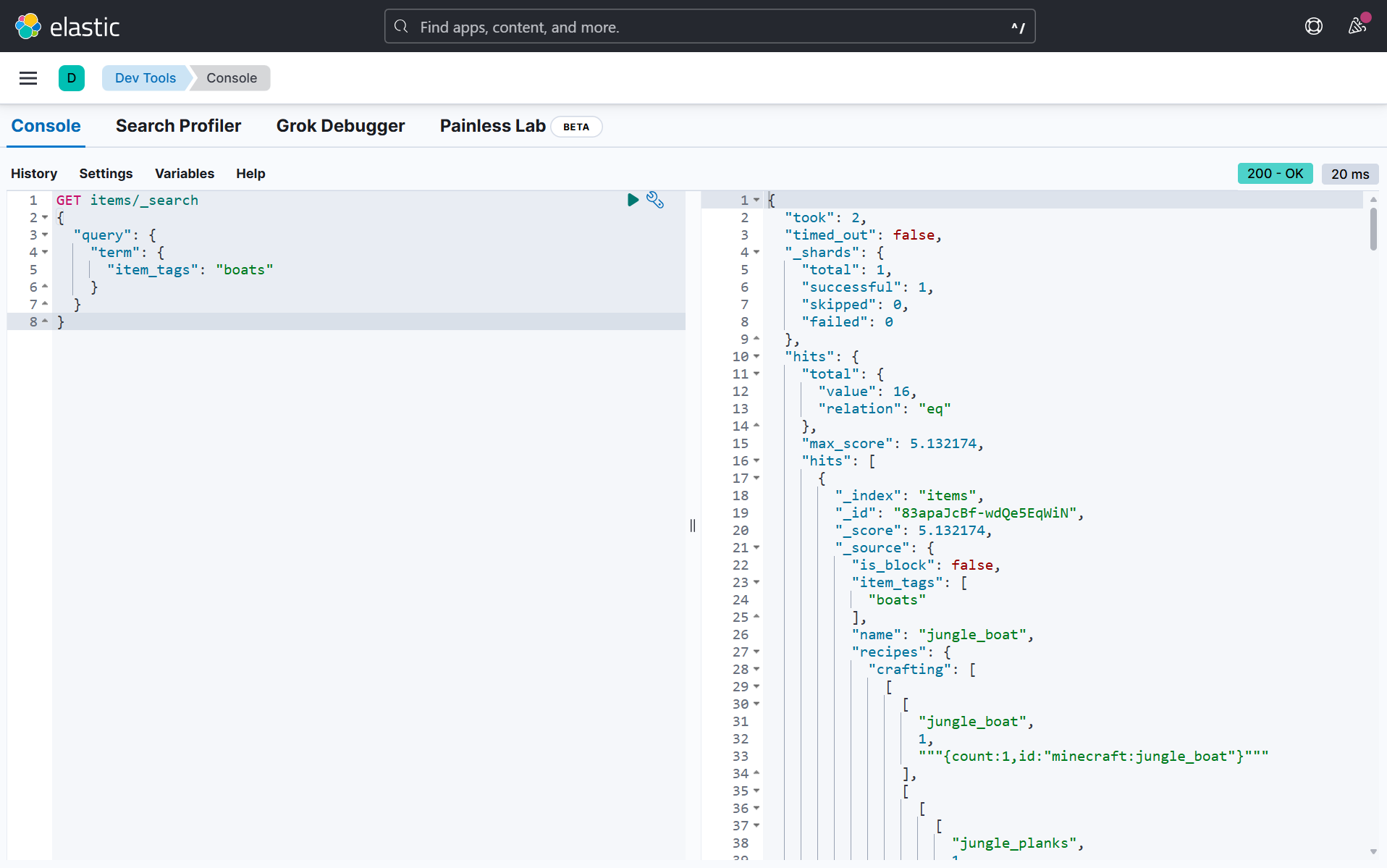
"term": {

"item\_tags": "boats"

}

}

}

Cette requête renvoie tous les documents dont item\_tags contient exactement ‘boats’.

## Trouver tous les items qui ne sont pas des blocs

GET items/\_search

{

"query": {

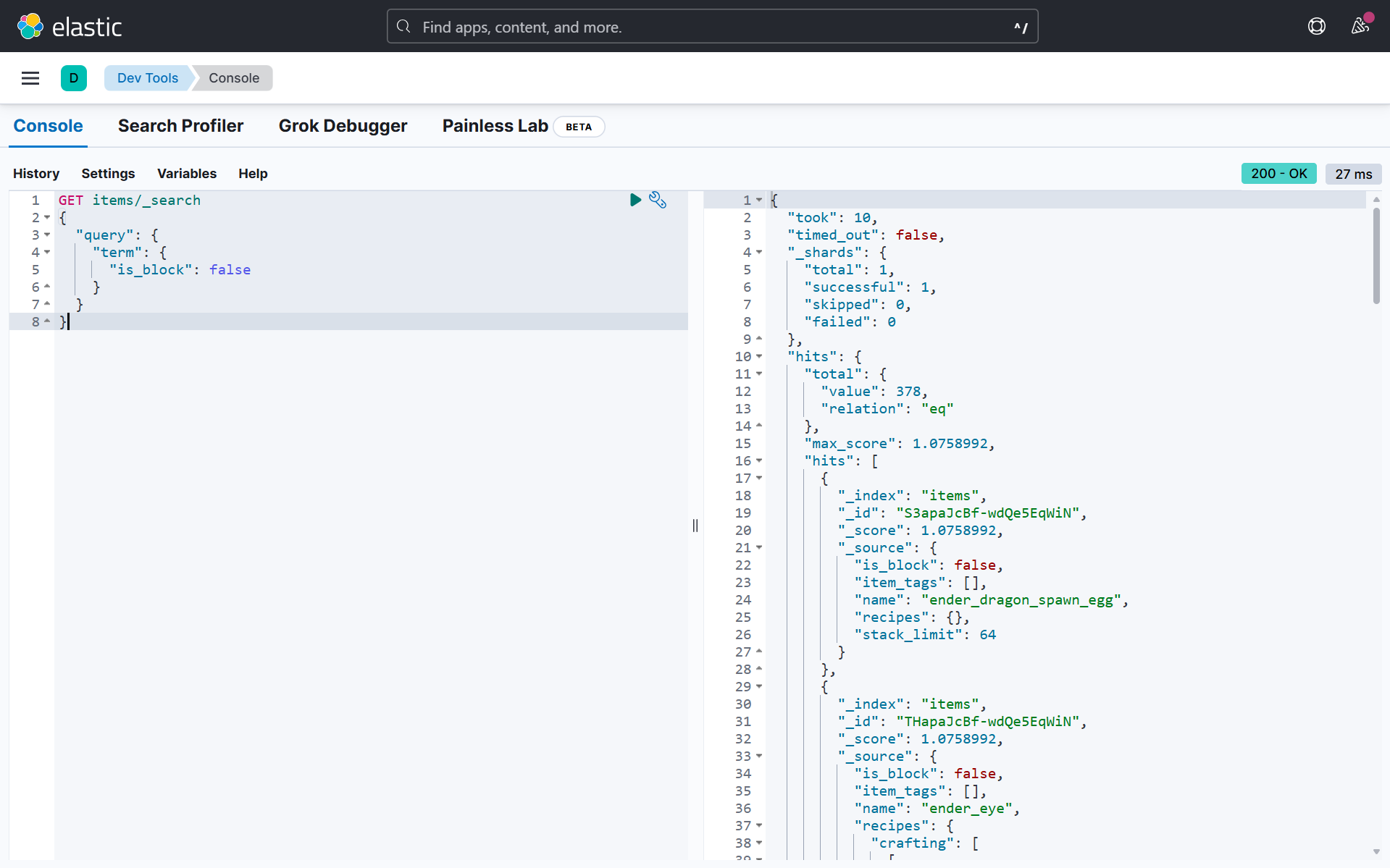
"term": {

"is\_block": false

}

}

}

Cela retourne tous les objets non-blocs comme les bateaux.

## Chercher un item par son nom exact

GET items/\_search

{

"query": {

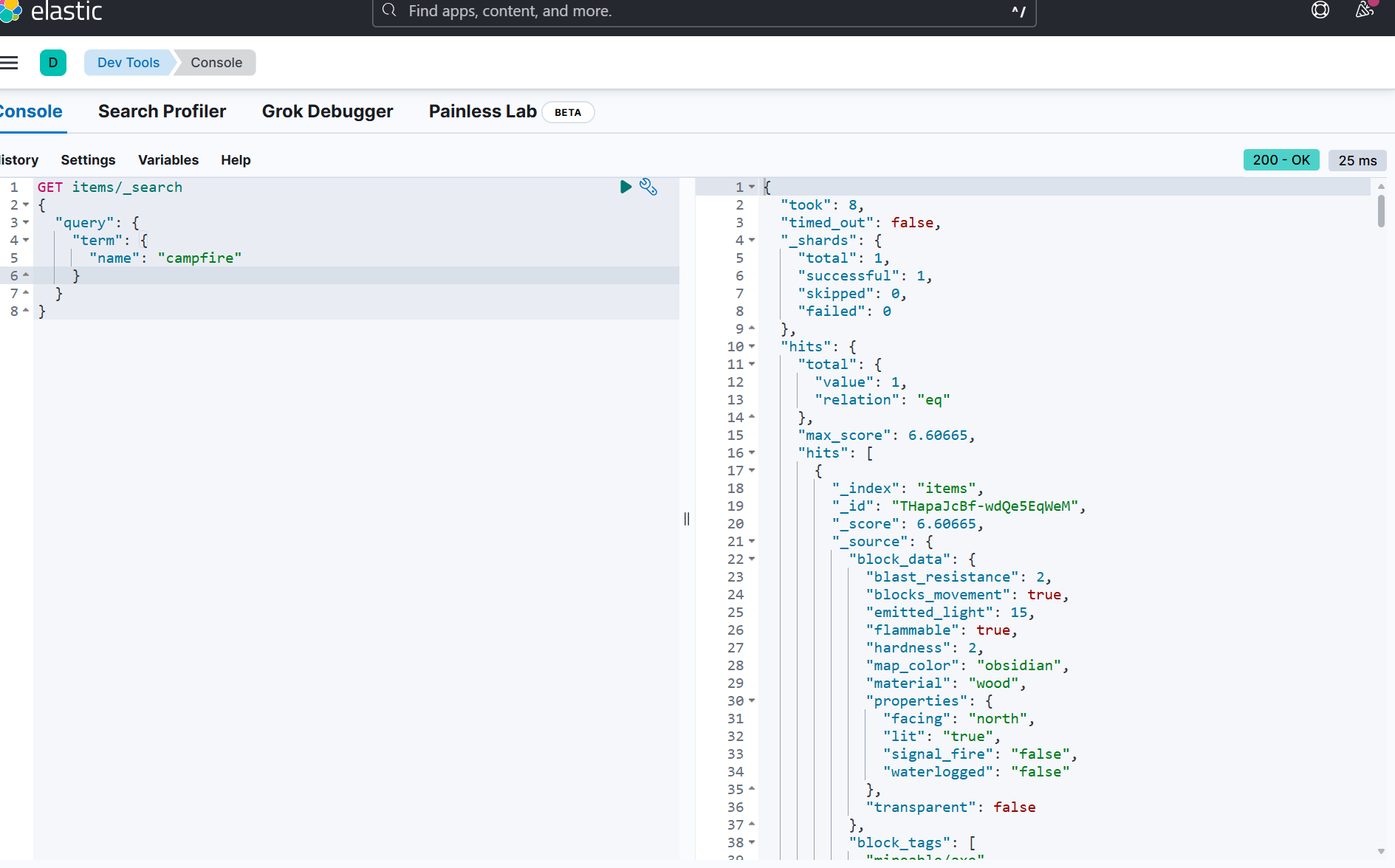
"term": {

"name": "campfire"

}

}

}

Retourne le document correspondant précisément à ‘acacia\_door’.

## Lister tous les blocs faits de « wood »

GET items/\_search

{

"query": {

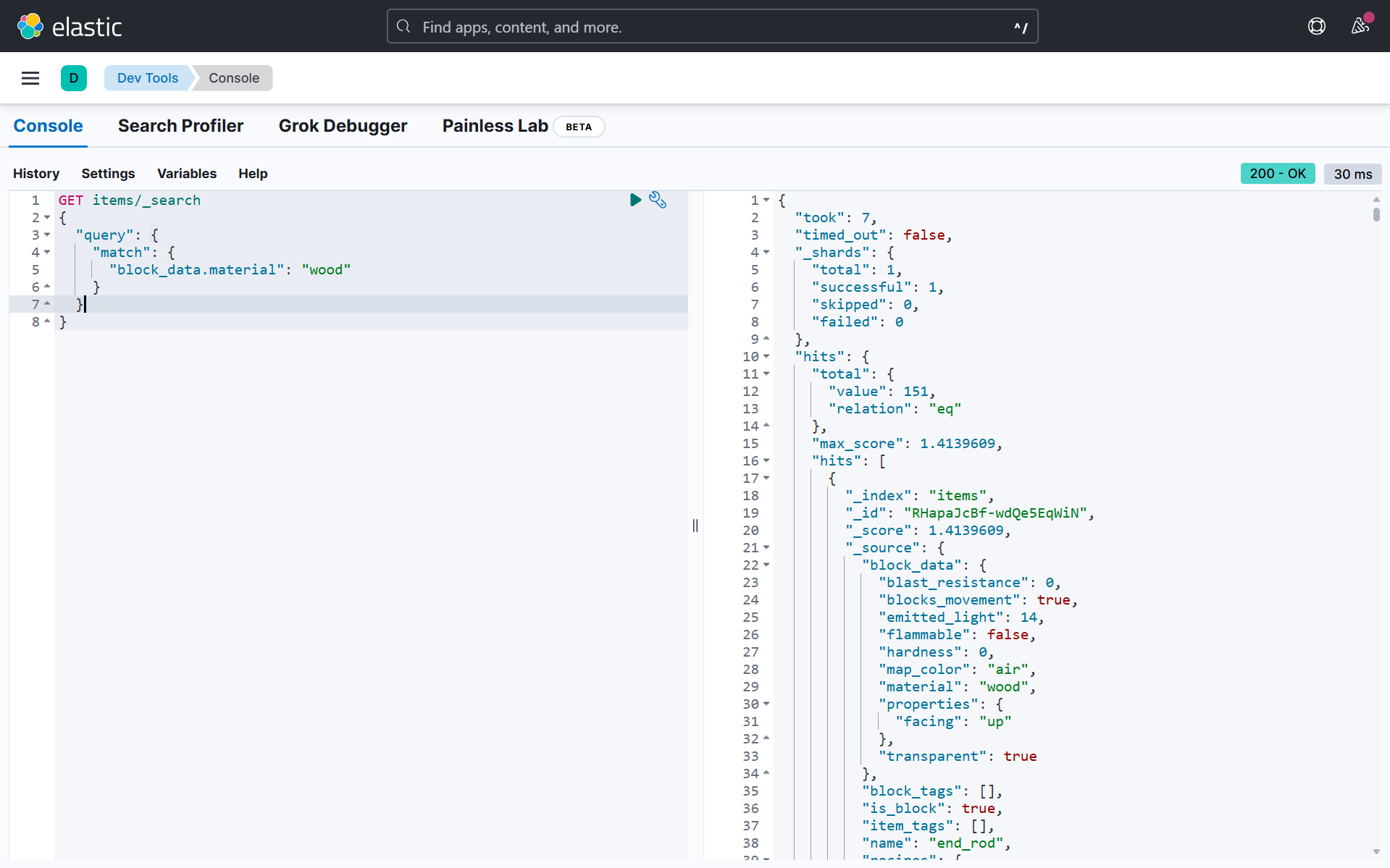
"match": {

"block\_data.material": "wood"

}

}

}

Retourne tous les items blocs (is\_block: true) dont le matériau est ‘wood’.

## Items dont le stack\_limt est égal à 1

GET items/\_search

{

"query": {

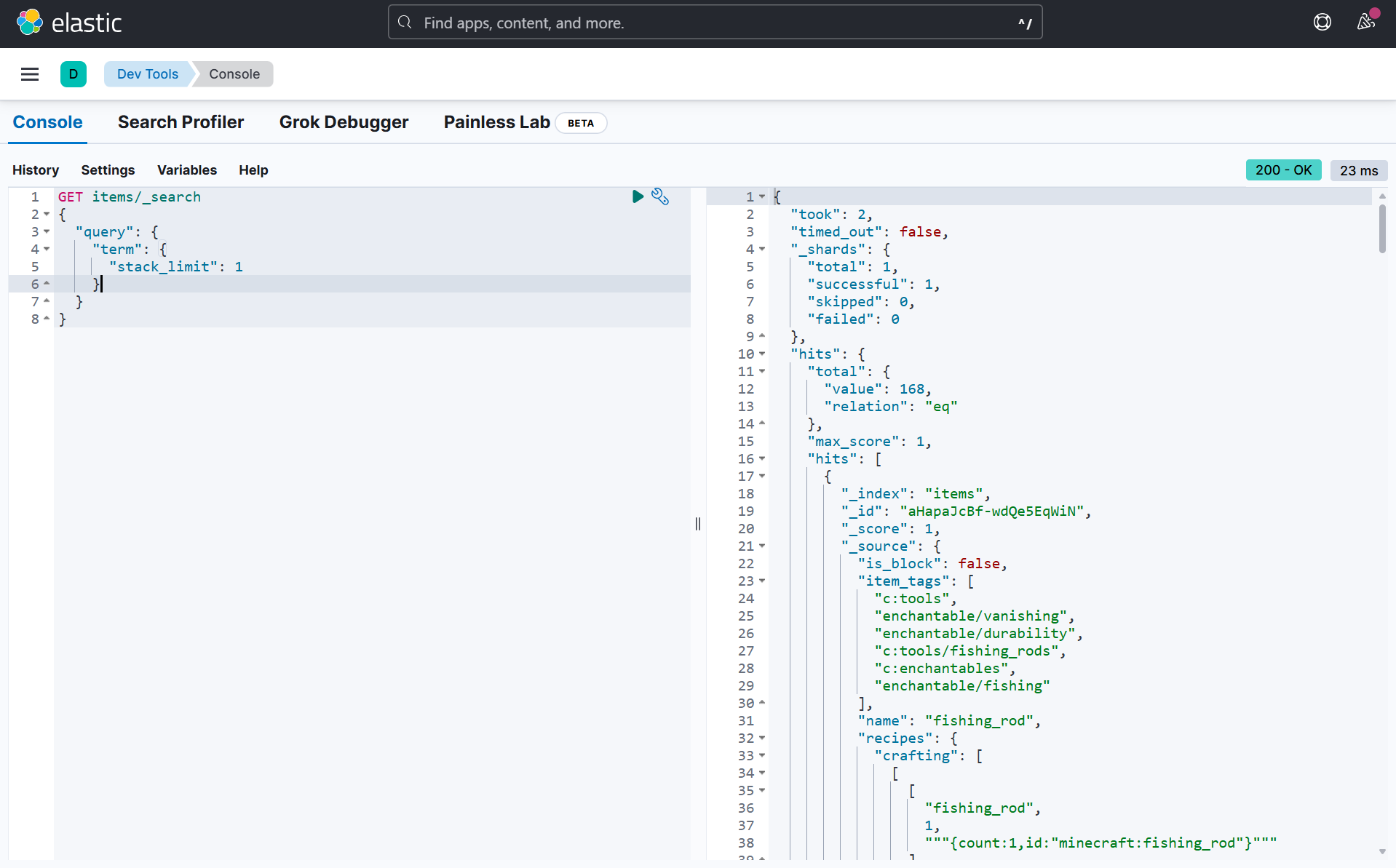
"term": {

"stack\_limit": 1

}

}

}

Retourne les items non-empilables (souvent uniques), comme les bateaux.