



Пагинация: скроллим, но не сдаемся

Артем Лысенко



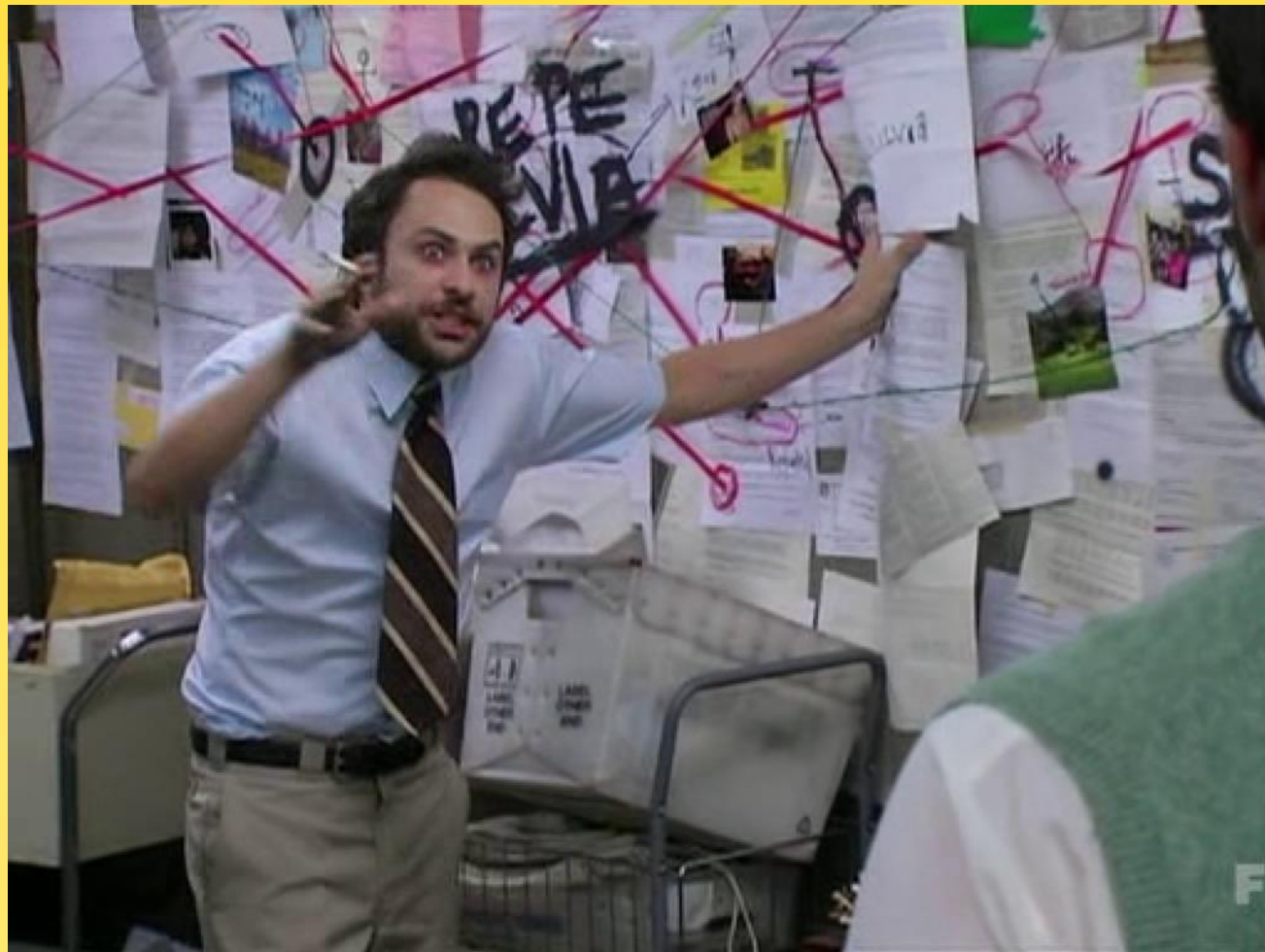
Артём Лысенко

Kotlin разработчик, RiskTech

\$ whoami

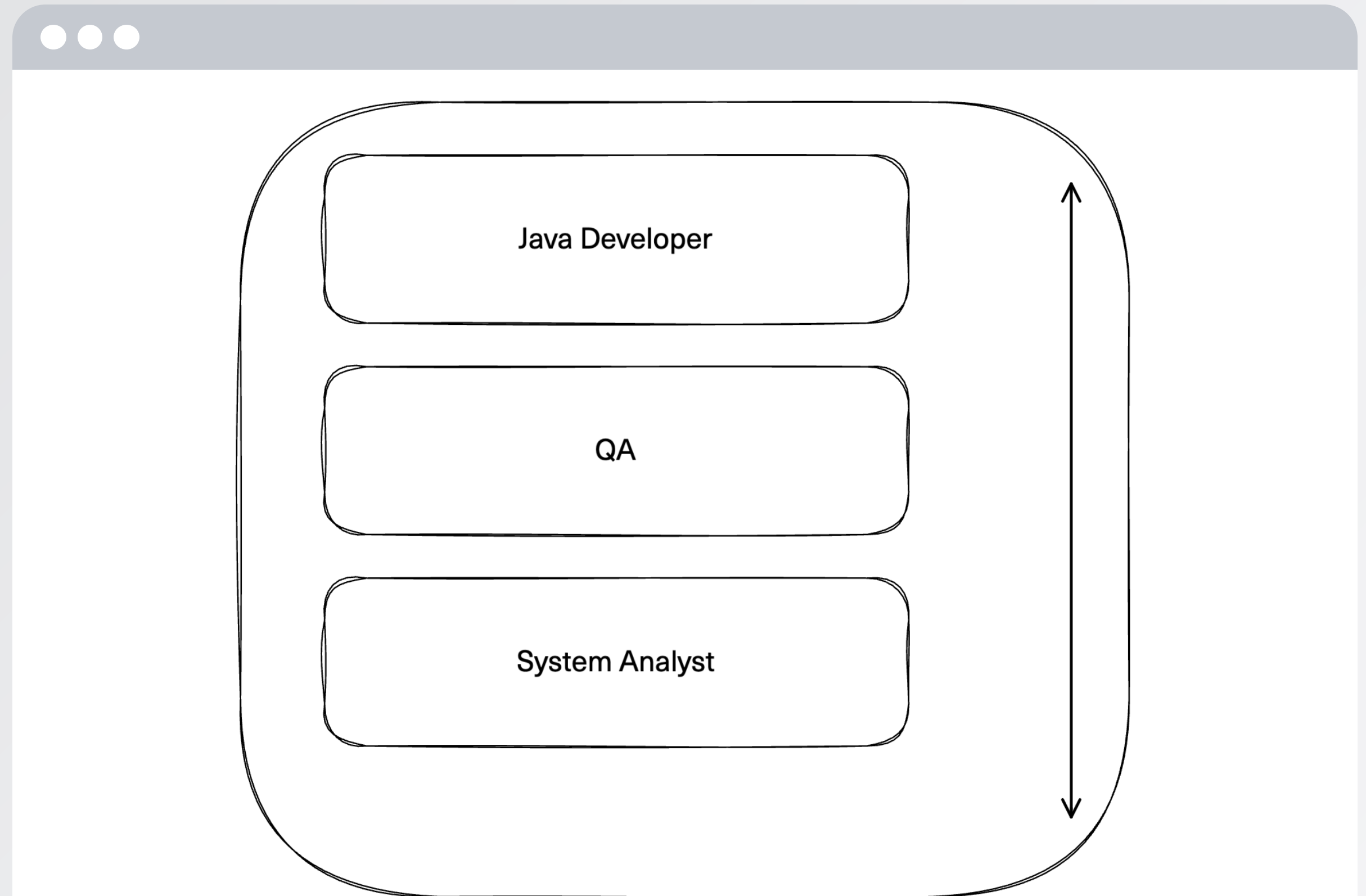
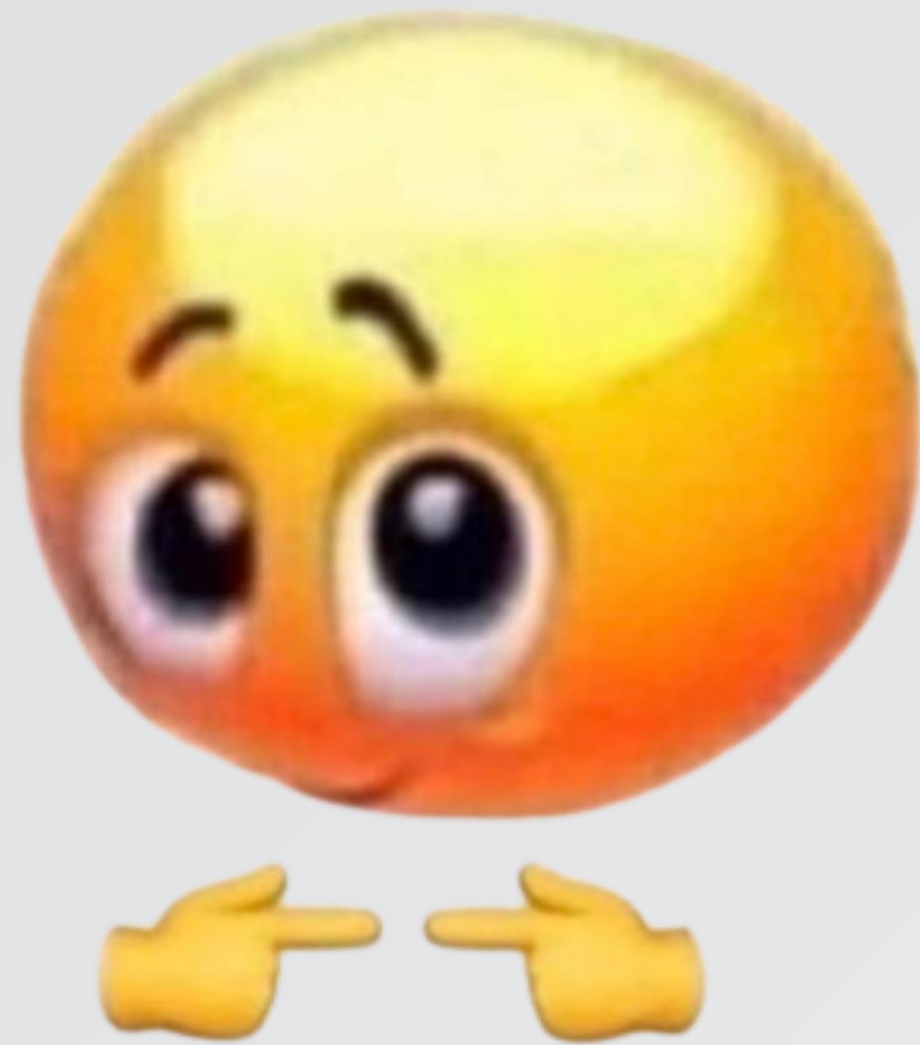
- ✓ 1.5+ года т-банке
- ✓ ТА в академии бекенда
- ✓ 4 курс ФКН ВГУ

План

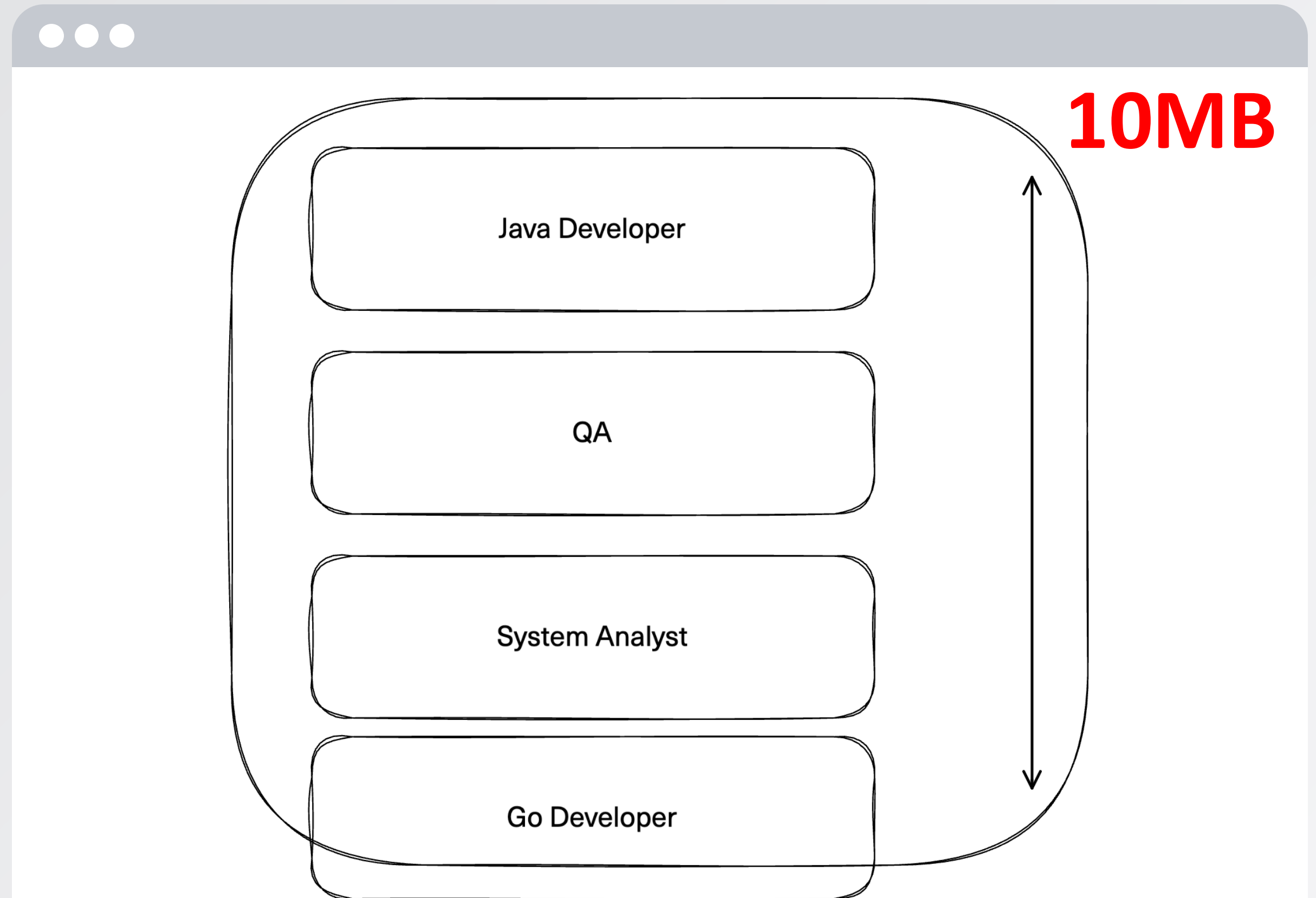


- ✓ Что такое пагинация?
- ✓ Подходы к реализации
- ✓ “Универсальный” API для пагинации

GET /vacancies



GET /vacancies



Что делать?

Ограничить размер ответа

- Не всегда подходит
- Зато просто!

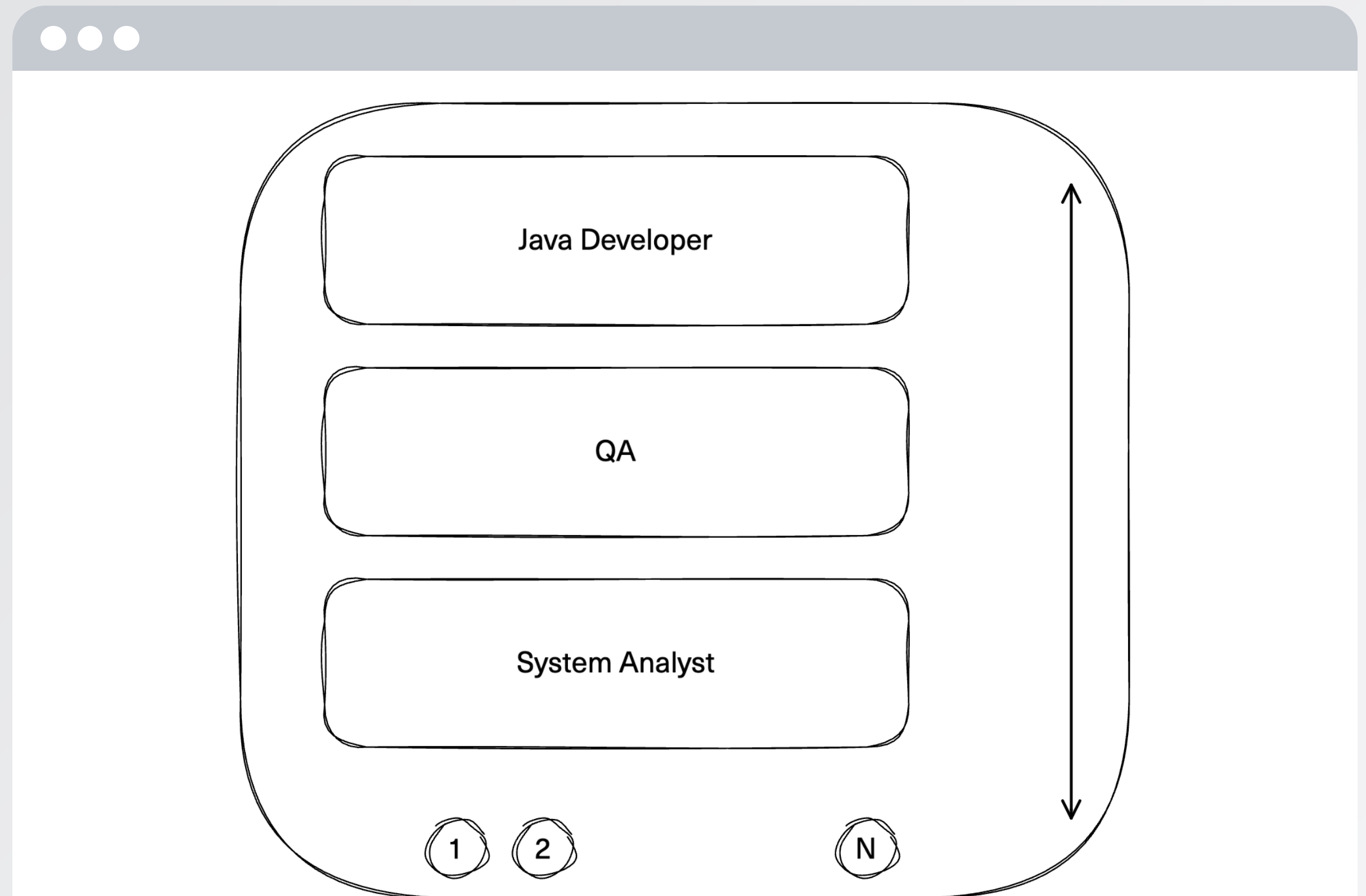
Добавить пагинацию

- Решает проблему
- Сломает клиентов

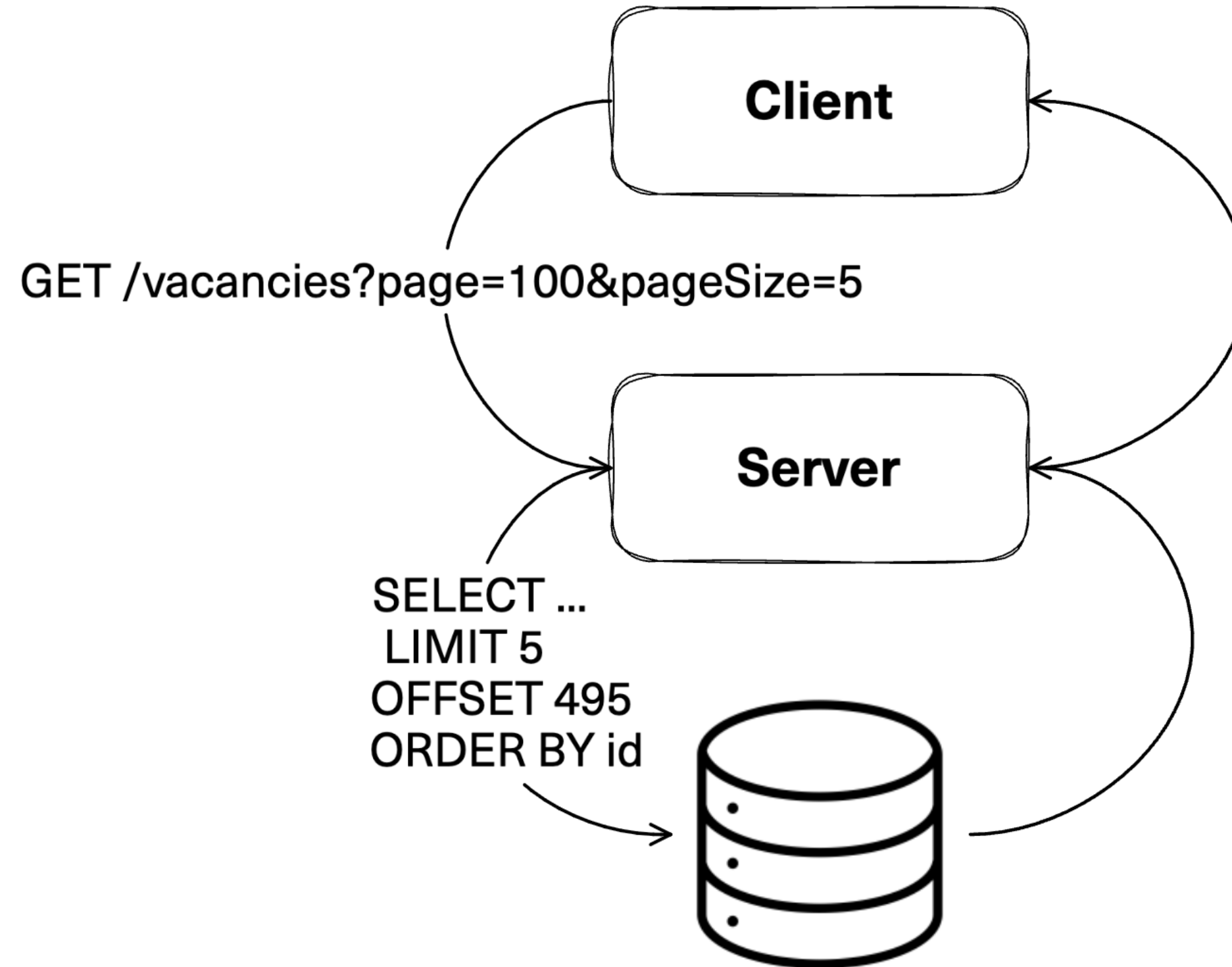


Про пагинацию нужно думать
заранее!

GET /vacancies?page=2&pageSize=5

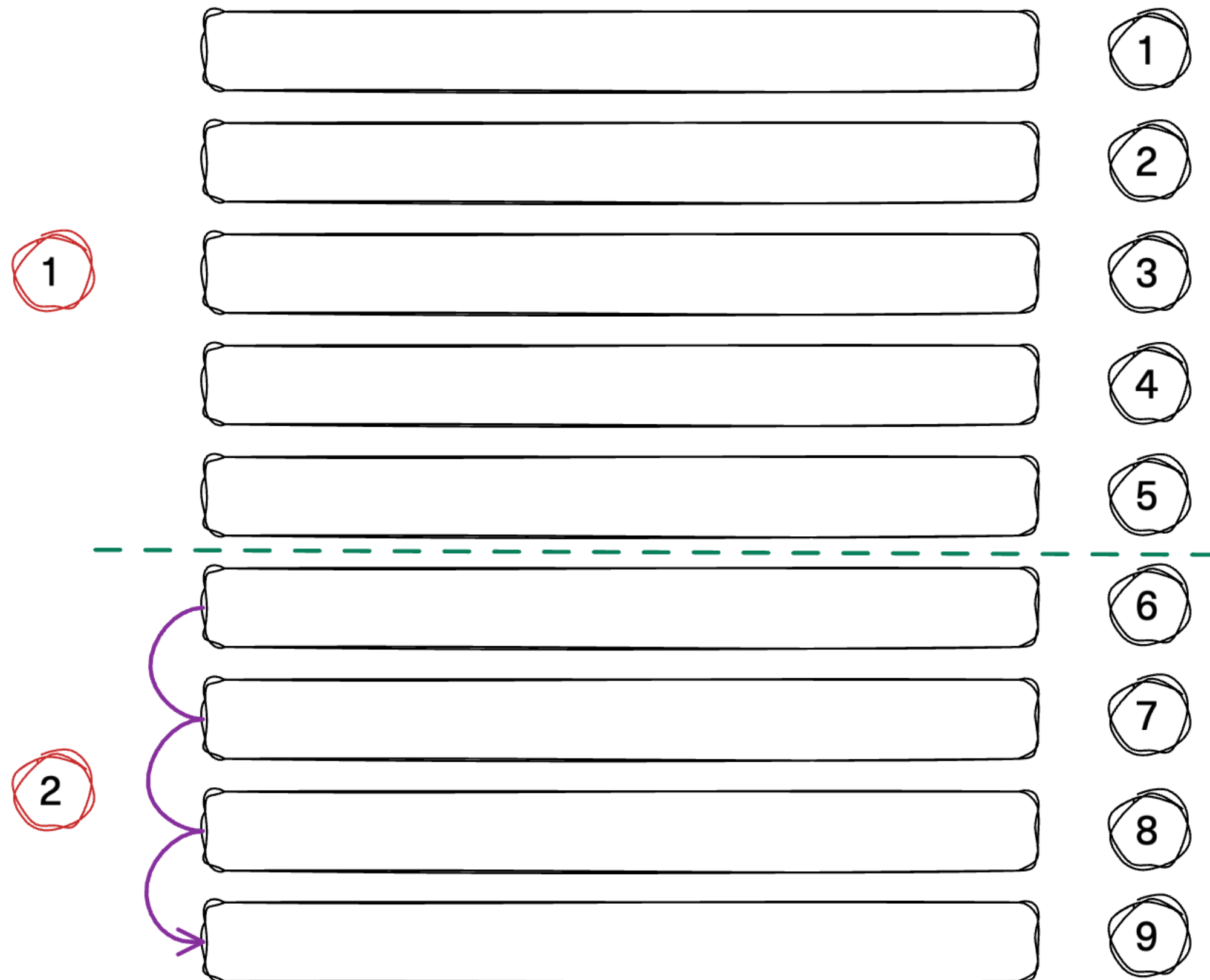


Путь запроса



Как работает offset

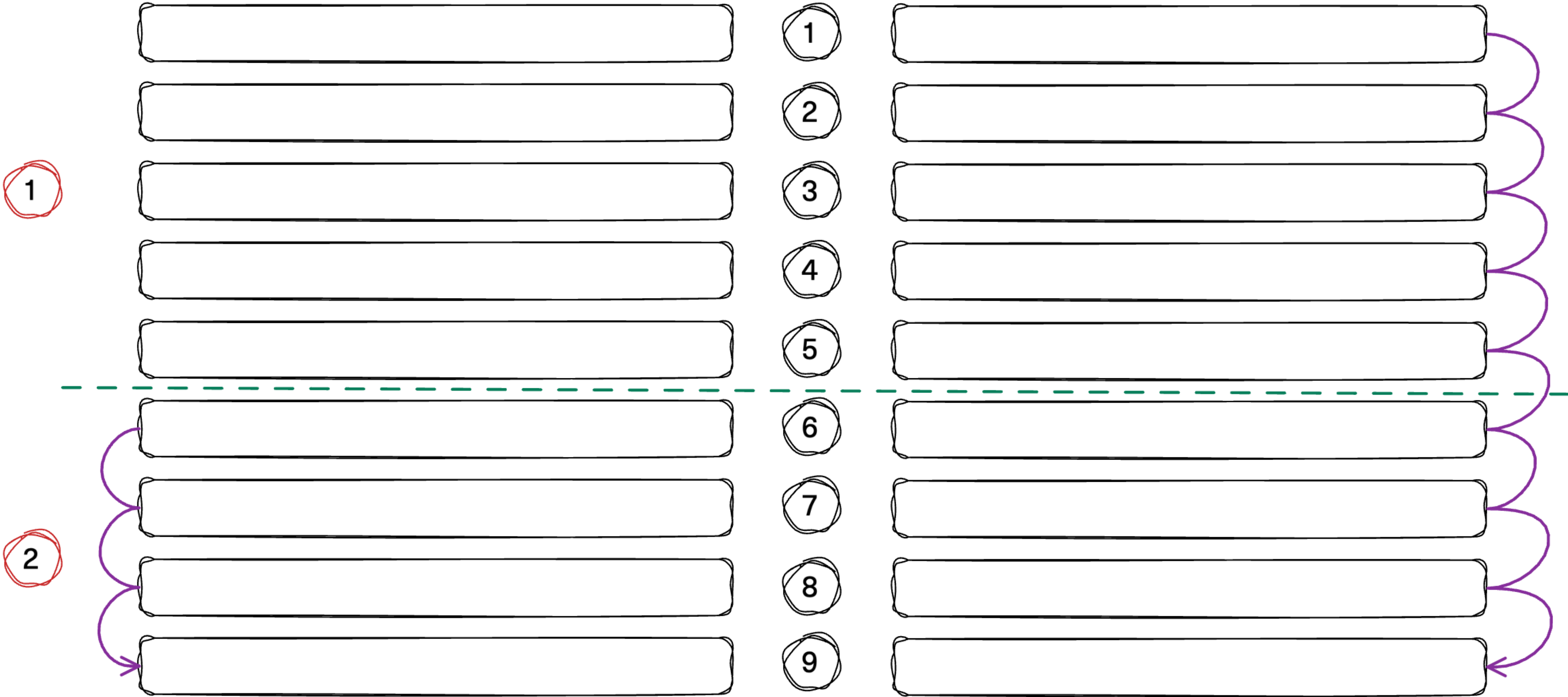
Ожидание



Как работает offset

Ожидание

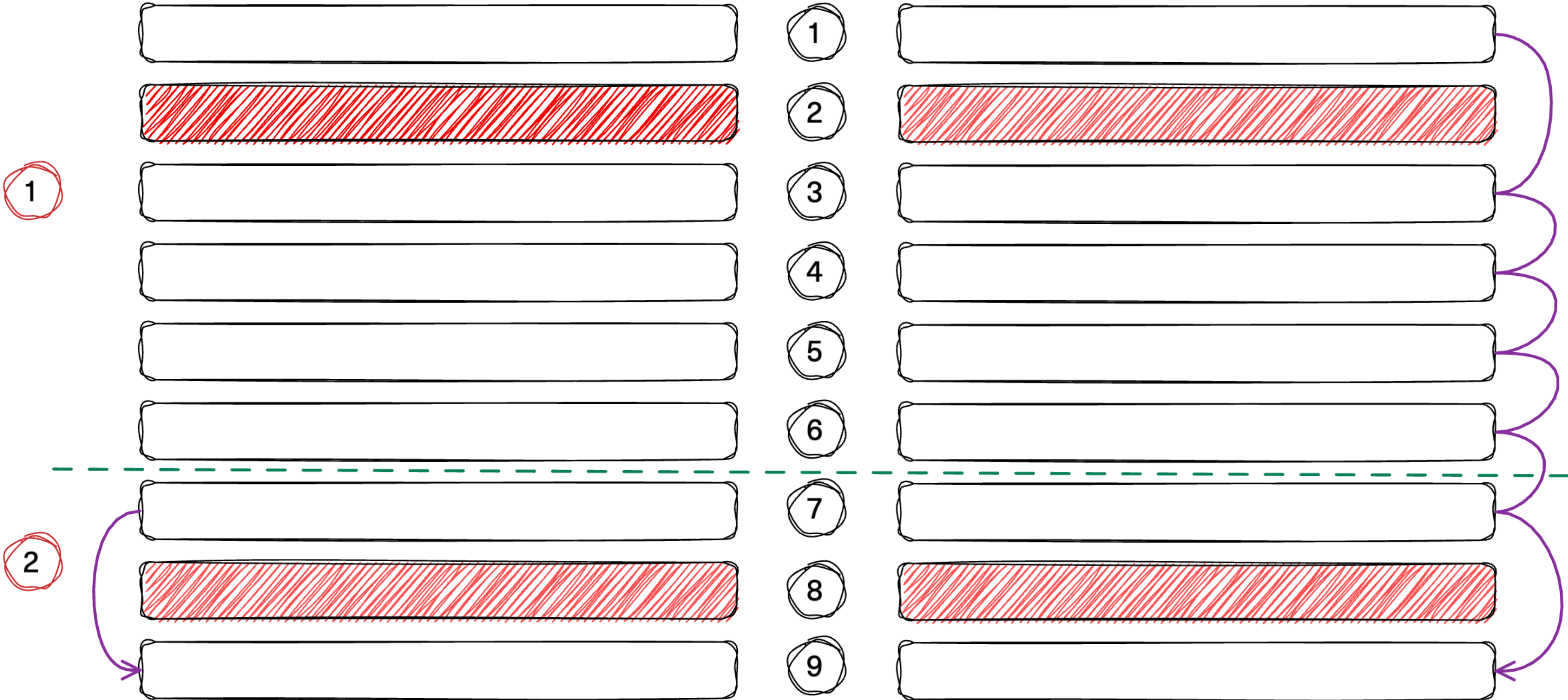
Реальность



Как работает offset

Ожидание

Реальность



План запроса

```
explain (analyze, costs off, buffers)  
select * from internships_int where tag = 'Java'  
order by id limit 10 offset 1999;
```

A-Z QUERY PLAN

Limit (actual time=1.621..1.622 rows=1 loops=1)

Buffers: shared hit=69

-> Sort (actual time=1.514..1.574 rows=2000 loops=1)

Sort Key: id

Sort Method: quicksort Memory: 189kB

Buffers: shared hit=69

-> Seq Scan on internships_int (actual time=0.010..1.198 rows=2000 loops=1)

Filter: ((tag)::text = 'Java'::text)

Rows Removed by Filter: 8000

Buffers: shared hit=66

Planning:

Buffers: shared hit=75

Planning Time: 0.871 ms

Execution Time: 1.692 ms

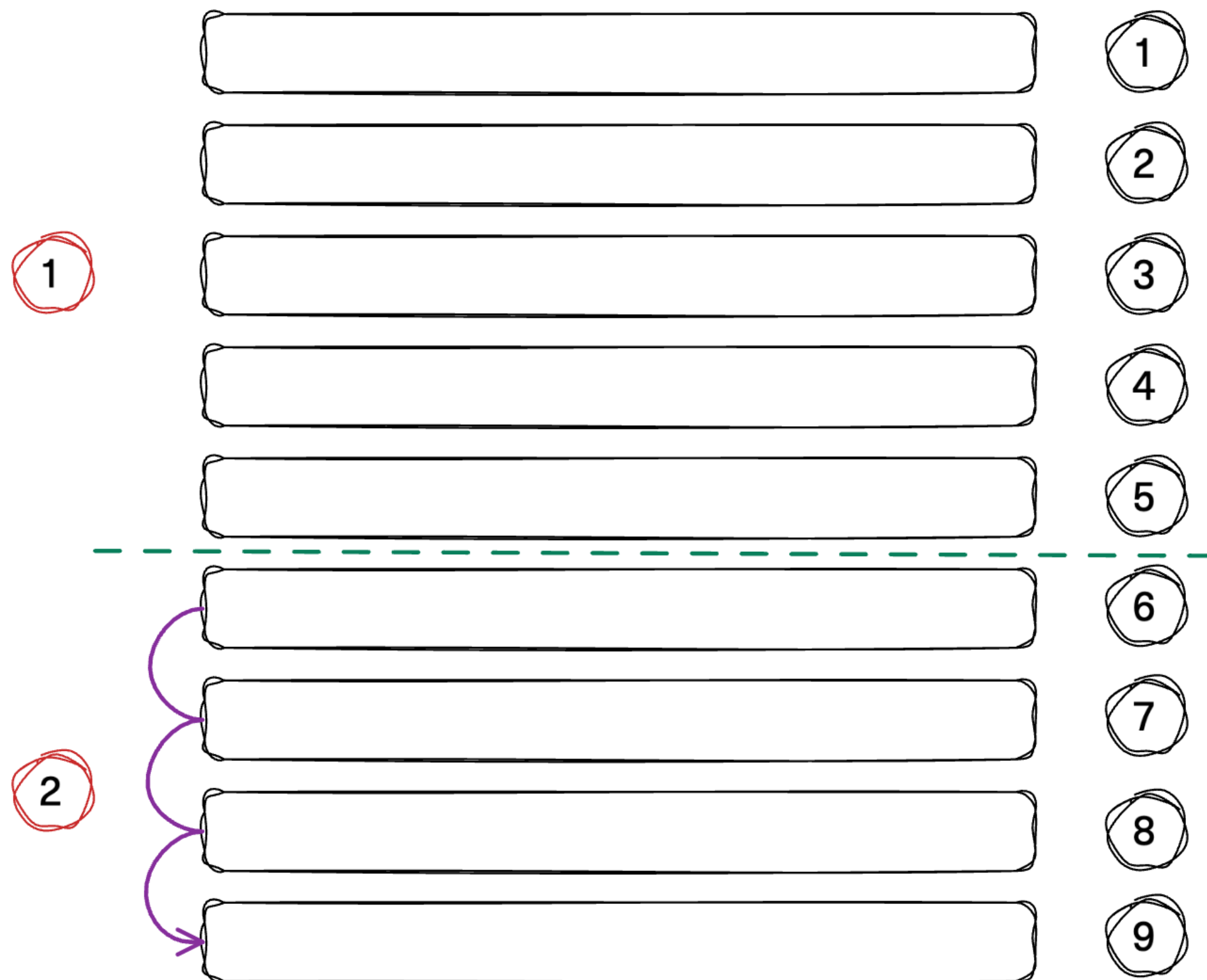


Что делать?



Как работает keyset

Реальность

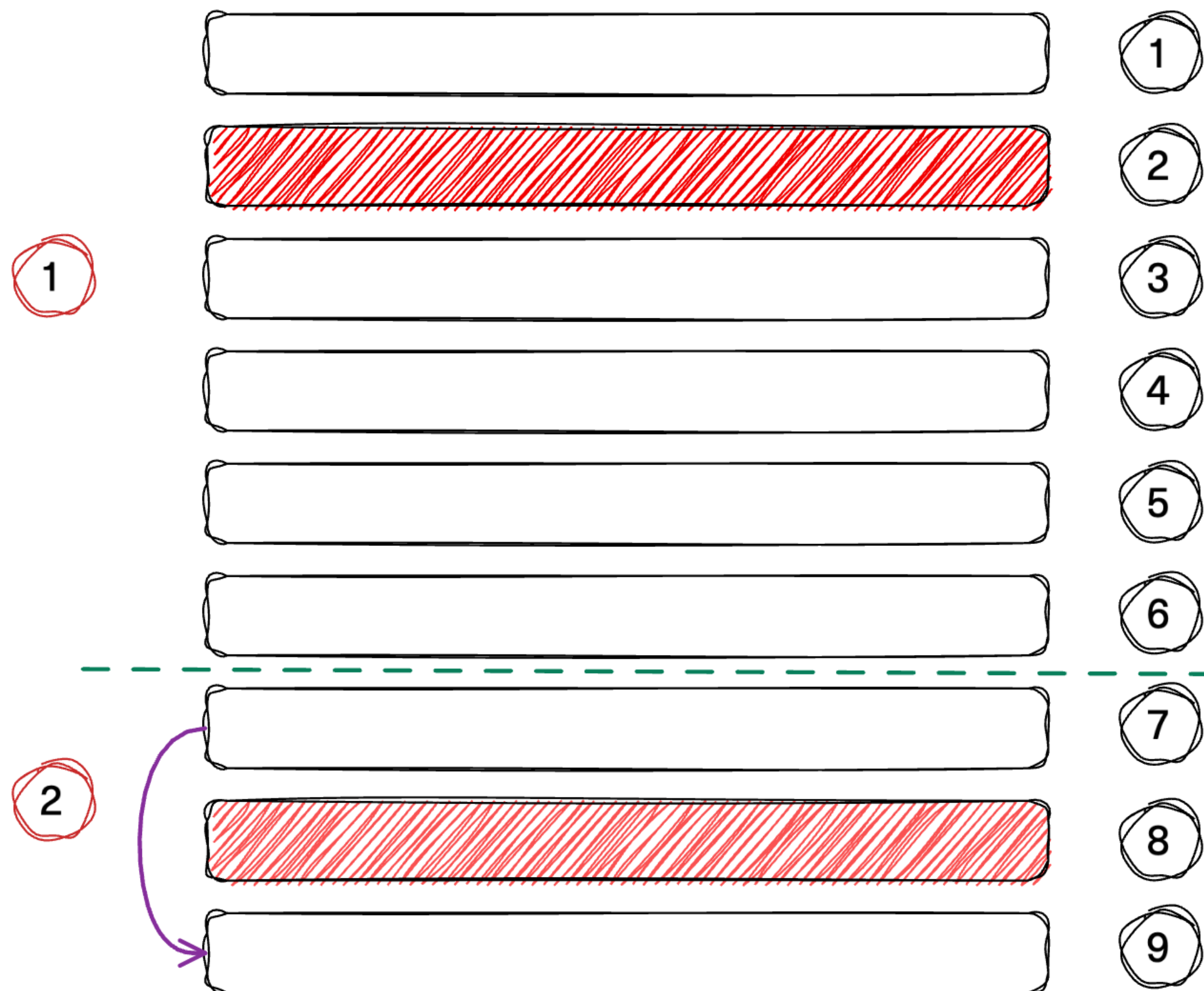


```
select *  
from internships_int  
order by id  
limit 5  
offset 5
```

```
select *  
from internships_int  
where id > 5  
order by id  
limit 5
```

Как работает keyset

Реальность

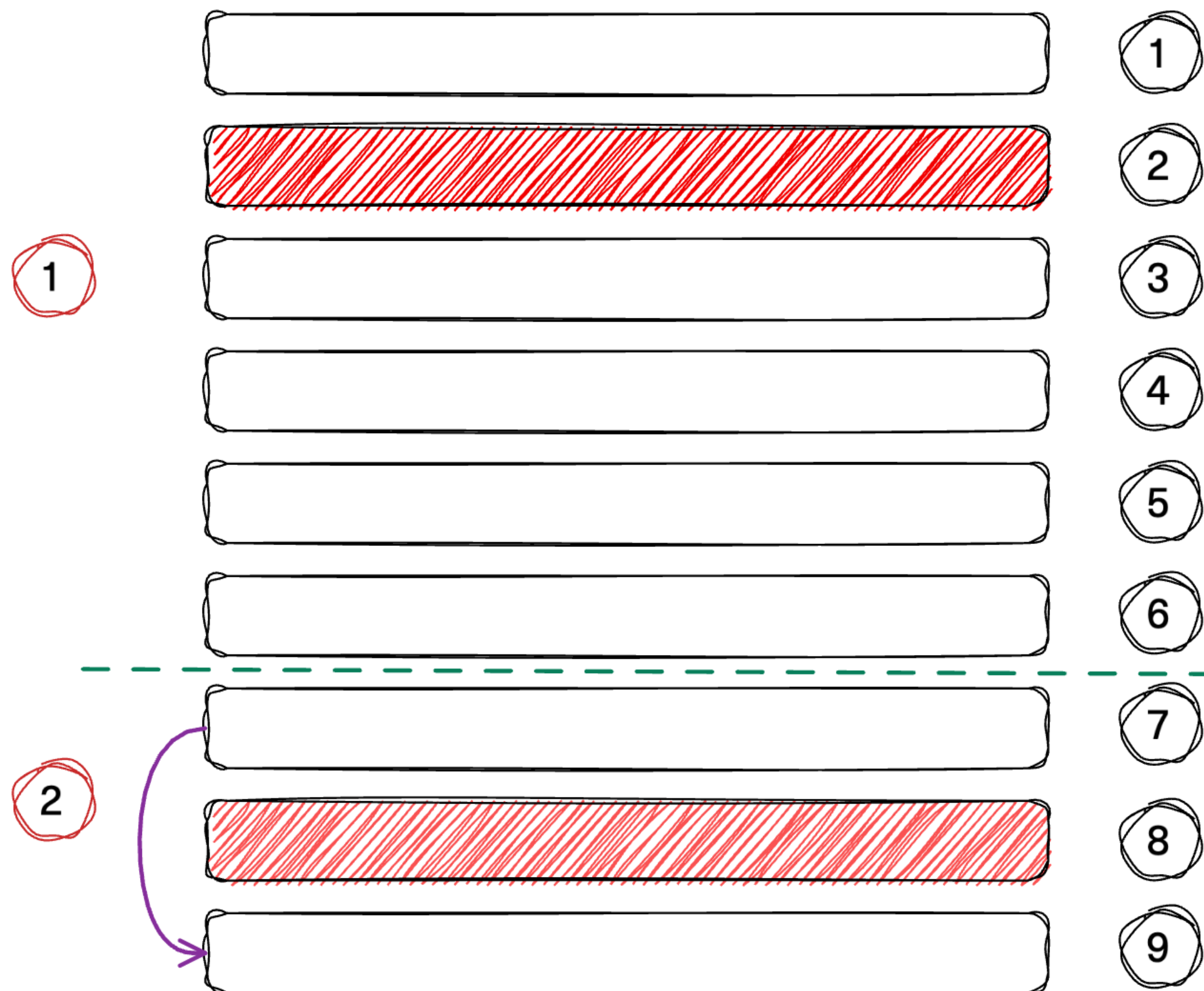


```
select *  
from internships_int  
order by id  
limit 5  
offset 5
```

```
select *  
from internships_int  
where id > 6  
order by id  
limit 5
```

Как работает keyset

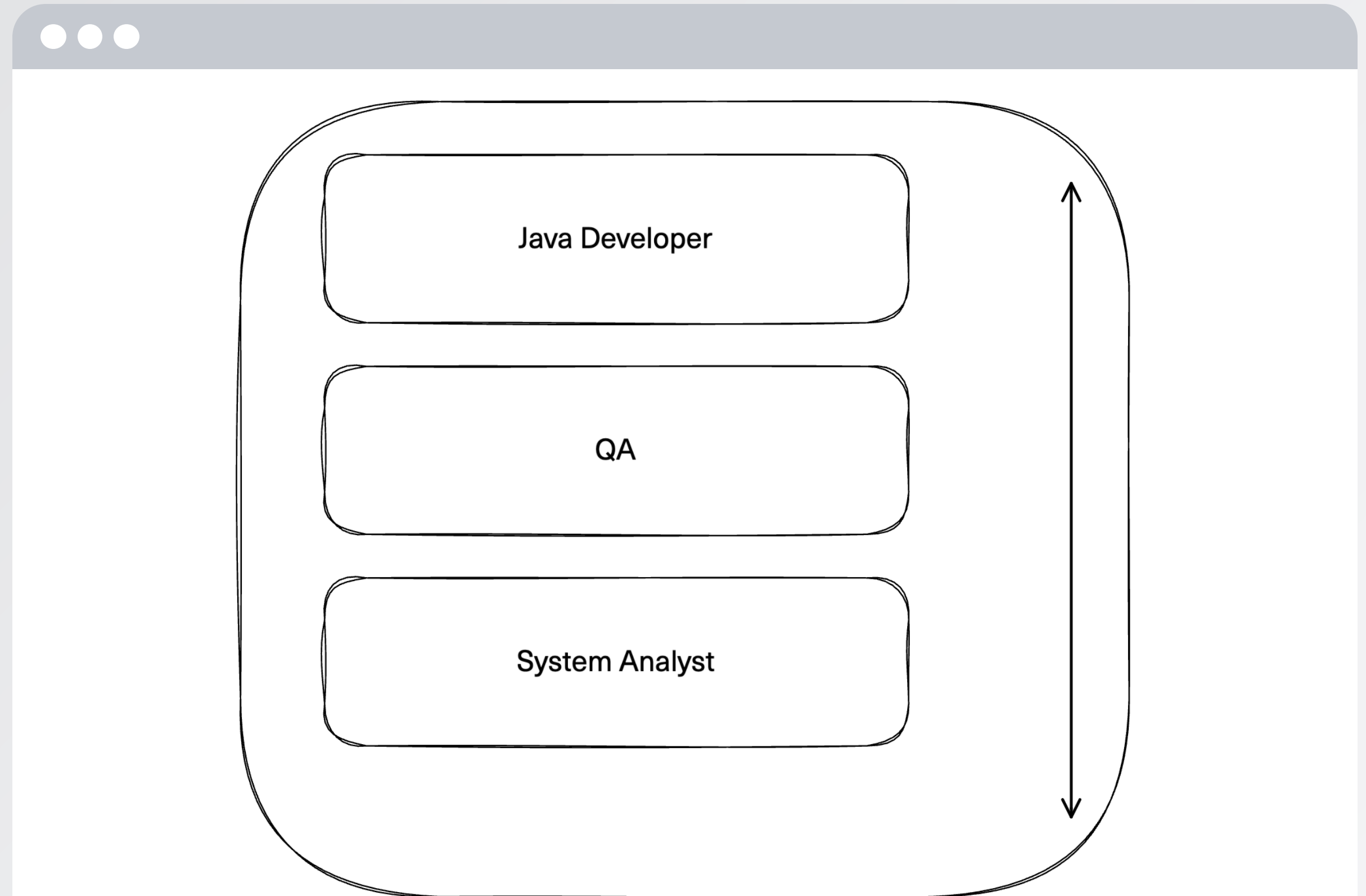
Реальность



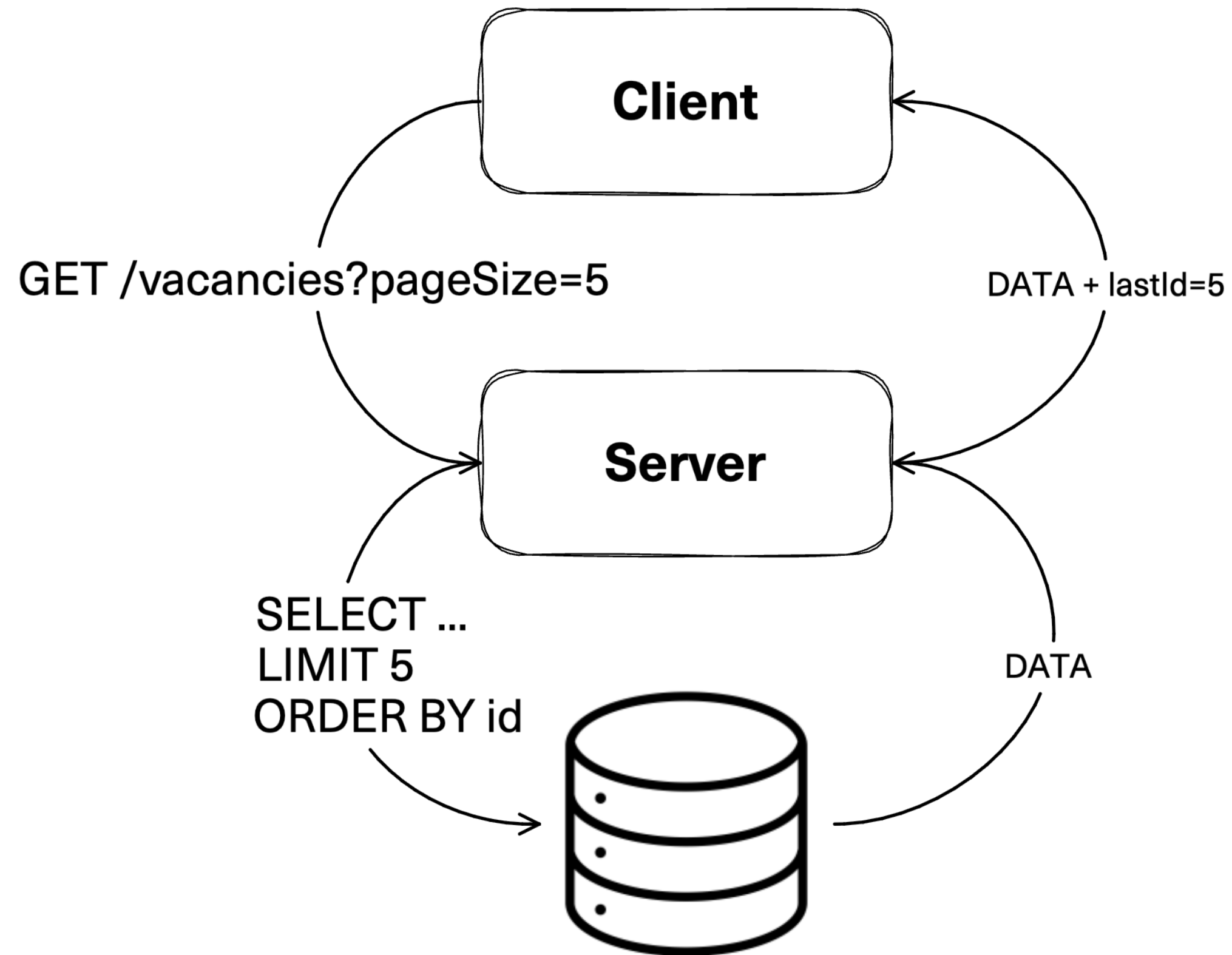
```
select *  
from internships_int  
order by id  
limit 5  
offset 5
```

```
select *  
from internships_int  
where id > :lastId  
order by id  
limit 5
```

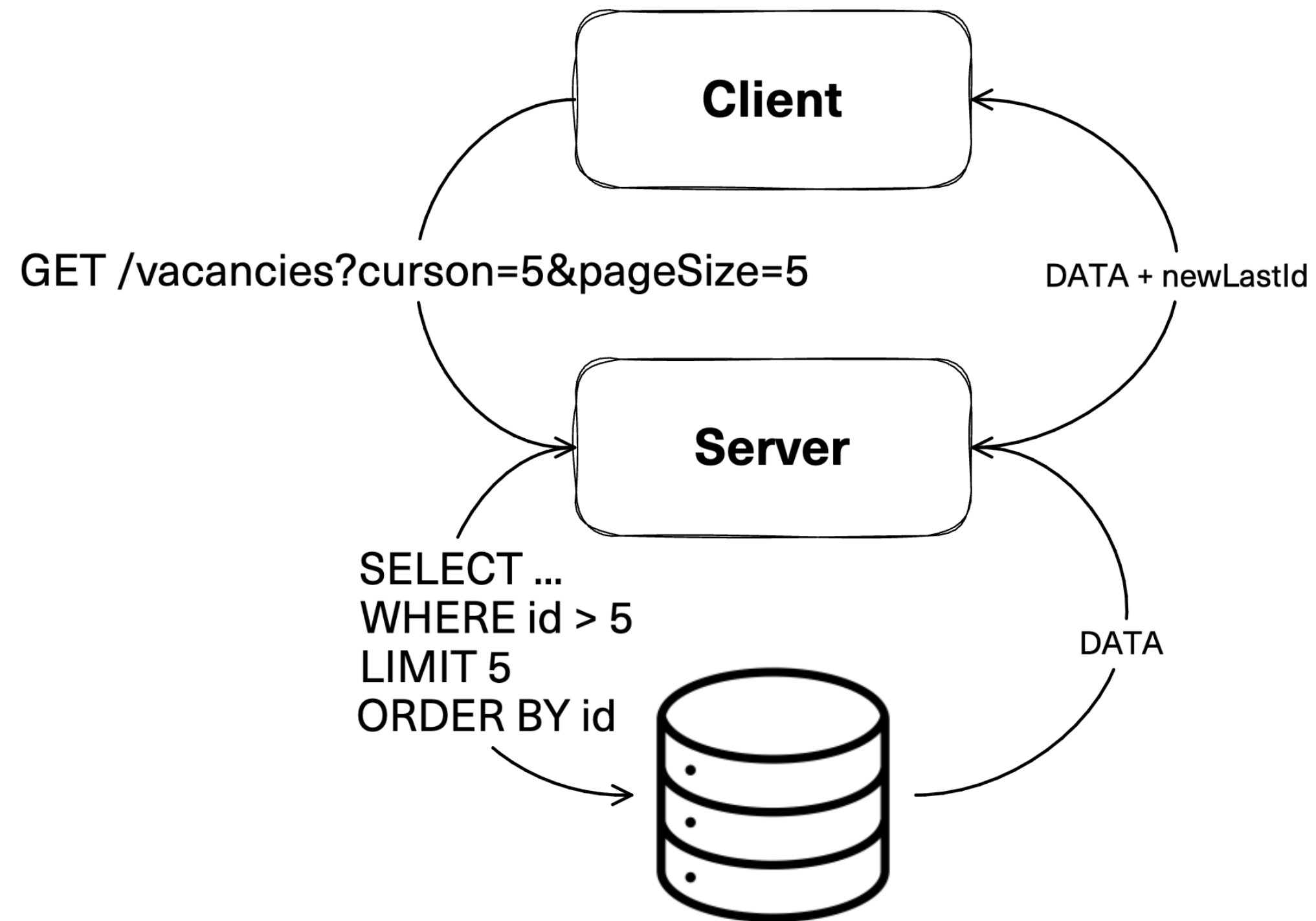

GET /vacancies?cursor=5&pageSize=5



Путь запроса – первая страница



Путь запроса – последующие страницы



План запроса

```
explain (analyze, costs off, buffers)
select * from internships_int
where tag = 'Java' and id > 9990
order by id limit 10;
```

A-Z QUERY PLAN

Limit (actual time=0.012..0.014 rows=2 loops=1)

Buffers: shared hit=3

-> Index Scan using internships_int_pkey on internships_int (actual time=0.012..0.013 rows=2 loops=1)

Index Cond: (id > 9990)

Filter: ((tag)::text = 'Java'::text)

Rows Removed by Filter: 8

Buffers: shared hit=3

Planning:

Buffers: shared hit=81

Planning Time: 0.837 ms

Execution Time: 0.029 ms



План запроса

```
explain (analyze, costs off, buffers)  
select * from internships_int where tag = 'Java'  
order by id limit 10 offset 1999;
```

A-Z QUERY PLAN

Limit (actual time=1.621..1.622 rows=1 loops=1)

Buffers: shared hit=69

-> Sort (actual time=1.514..1.574 rows=2000 loops=1)

Sort Key: id

Sort Method: quicksort Memory: 189kB

Buffers: shared hit=69

-> Seq Scan on internships_int (actual time=0.010..1.198 rows=2000 loops=1)

Filter: ((tag)::text = 'Java'::text)

Rows Removed by Filter: 8000

Buffers: shared hit=66

Planning:

Buffers: shared hit=75

Planning Time: 0.871 ms

Execution Time: 1.692 ms



План запроса

```
explain (analyze, costs off, buffers)
select * from internships_int
where tag = 'Java' and id > 9990
order by id limit 10;
```

A-Z QUERY PLAN

Limit (actual time=0.012..0.014 rows=2 loops=1)

Buffers: shared hit=3

-> Index Scan using internships_int_pkey on internships_int (actual time=0.012..0.013 rows=2 loops=1)

Index Cond: (id > 9990)

Filter: ((tag)::text = 'Java'::text)

Rows Removed by Filter: 8

Buffers: shared hit=3

Planning:

Buffers: shared hit=81

Planning Time: 0.837 ms

Execution Time: 0.029 ms



OFFSET

Преимущества

- Интуитивный API
- Поддержка от фреймворков
- Просто реализовать

Недостатки

- Производительность

KEYSET

Преимущества

- Производительность
- Простой API

Недостатки

- Нет возможности перейти к конкретной странице
- Не работает из коробки с UUID v4
- Для каждой колонке участвующей в сортировке должен быть создан индекс

Реальная система

Frontend



Server

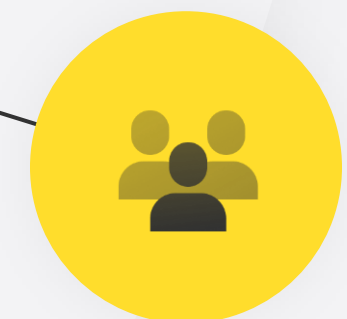
Analytics



Mobile



Partners



PageToken – как решение всех проблем

GET /vacancies

?maxResults = 5

&pageToken = aGVoZSBjYXVnaHQgbWU

PageToken – как решение всех проблем

GET /vacancies

?maxResults = ЧИСЛО

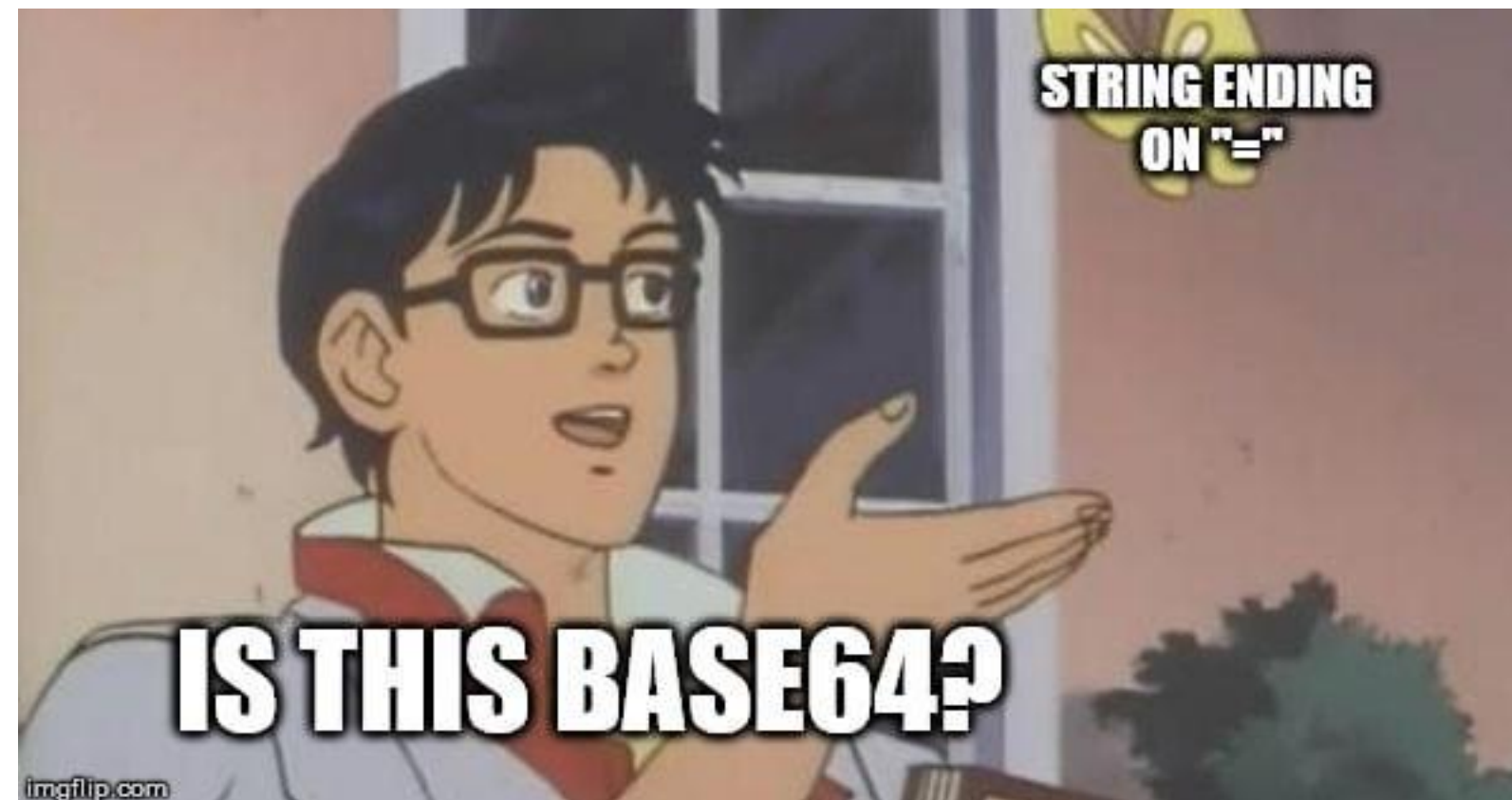
&pageToken = СТРОКА

PageToken – как решение всех проблем

GET /vacancies

?maxResults = 5

&pageToken = {"lastId": 5}



PageToken версионирование

GET /vacancies

?maxResults = 5

&pageToken = **v1**:offset=5

GET /vacancies

?maxResults = 5

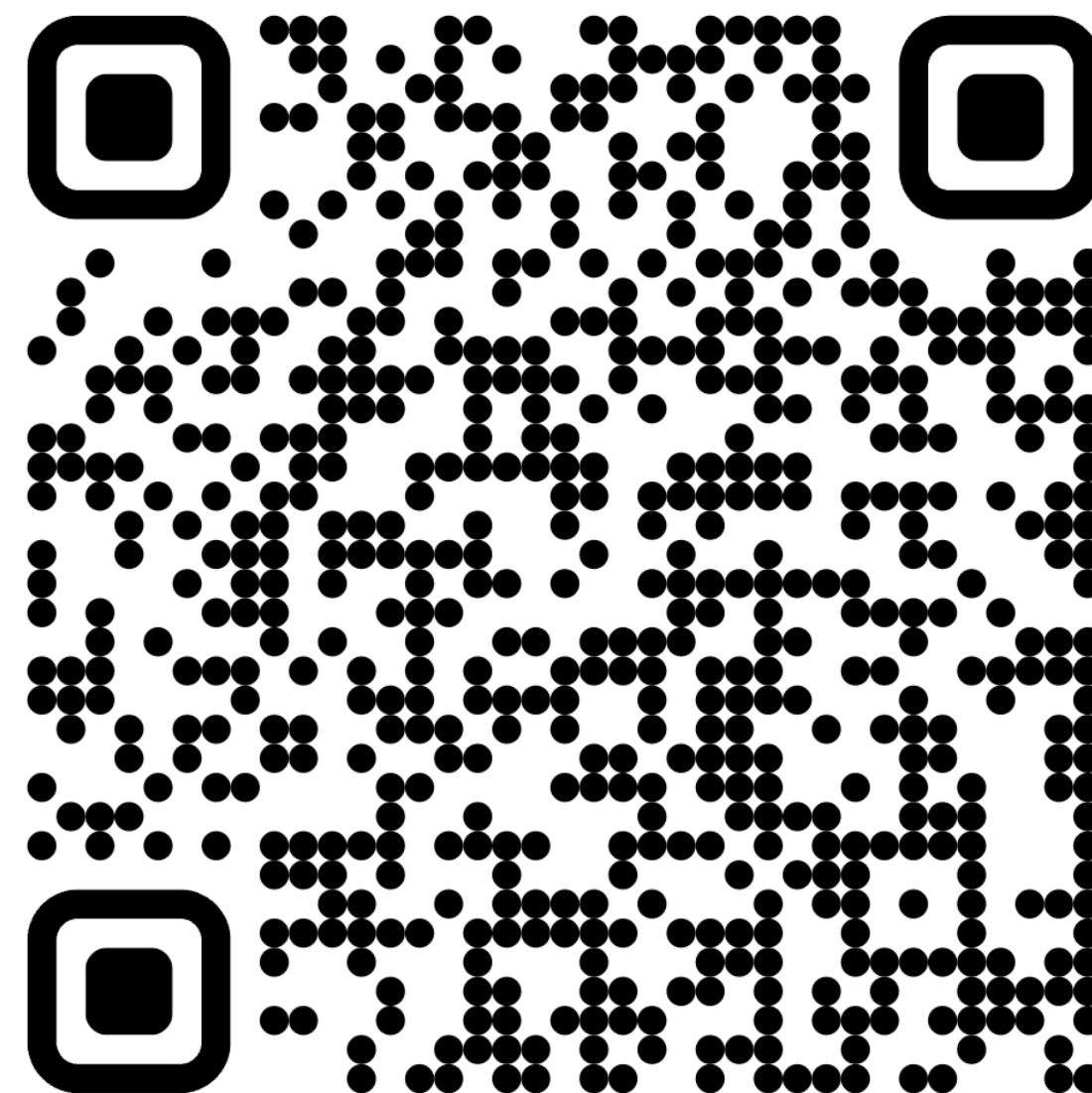
&pageToken = **v2**:aGVoZSBjYXVnaHQgbWU

Итог



- ✓ Доверяй, но проверяй
- ✓ Непрозрачный контракт это круто
- ✓ Делай так, чтобы думать потом

Вопросы



Помоги мне стать лучше

