

Как работает параметр namespace

- 1. Цель
- 2. Основные термины и определения
- 3. Что такое namespace
- 4. Пример изоляции экспериментов с помощью namespace
- 5. Когда использовать Namespace?
- 6. Функция inNamespace
 - Параметры функции inNamespace:
- 7. Как работает namespace для многомерных схем?

1. Цель

Основная цель убедиться как работает параметр namespace, в том числе и в многомерных тестах. После прочтения данной документации, у вас должны быть ответы на вопросы:

- Что есть namespace и как он работает?
- Действительно ли использование namespace ведет к изоляции 2 тестов?
- Действительно ли применение namespace в параллельных тестах размещает пользователей ортогонально между группами?

2. Основные термины и определения

Feature Flags - это инструмент, позволяющий управлять доступностью определённых функций (features) для разных групп пользователей.

Изолированные тесты (mutually exclusive tests) - это эксперименты, в которых участие пользователя строго ограничено одним экспериментом.

Пересекающиеся тесты (overlapping tests) - это эксперименты, в которых один и тот же пользователь может одновременно участвовать в нескольких тестах.

Хеш-функция - это алгоритм, который принимает входные данные произвольного размера и преобразует их в выходные данные фиксированного размера.

Соль — строка фиксированной длины, которая добавляется (методом concat) к входным данным перед применением хеш-функции.

Бакеты - набор пользователей, которым показывается одна из версий продукта.

Коллизия - ситуация, при которой пользователи попадают в один и тот же бакет.

Ортогональность - результаты одного эксперимента не влияют на результаты другого эксперимента.

3. Что такое namespace

Предположим, что у нас есть эксперименты, которые могут конфликтовать друг с другом. Namespace позволяет разделять пользователей между разными тестами таким образом, чтобы один и тот же пользователь не участвовал одновременно в нескольких тестах, которые могут влиять друг на друга.

Как это работает? Пользователям случайным образом присваивается значение от 0 до 1 для Namespace. Каждый эксперимент имеет диапазон значений. Пользователи участвуют в эксперименте только в том случае, если их значение попадает в диапазон эксперимента. Таким образом, пока два диапазона экспериментов не пересекаются, пользователи будут находиться только в одном из них.

[blocked URL](#)

4. Пример изоляции экспериментов с помощью namespace

Допустим, у нас есть два эксперимента, которые мы хотим изолировать:

1. Эксперимент А
2. Эксперимент В

Чтобы гарантировать, что эти эксперименты взаимоисключающие, мы можем поместить их в один namespace с неперекрывающимися диапазонами:

- Эксперимент А в namespace "lenta" с диапазоном [0, 0.5).

- Эксперимент В в том же namespace "lenta" с диапазоном [0.5, 1).

Это означает, что пользователь, чье значение хеша попадает в диапазон [0, 0.5), будет участвовать только в эксперименте А и не будет включен в эксперимент В, и наоборот.

5. Когда использовать Namespace?

Решение о проведении изолированных экспериментов зависит от вероятности возникновения взаимодействия эффектов (т.н. **эффект сложения**). В таких случаях мы не можем определить, какой тест внес больший вклад в стат значимый результат. Совместный эффект может быть не равен сумме эффекта, измеренных по отдельности. Эффект сложения менее вероятен, если тесты проводятся в разных разделах приложения или отслеживают разные цели.

Если пользователь одновременно участвует в обоих экспериментах, его впечатления и поведение могут быть искажены взаимным влиянием этих изменений. Это может затруднить точное определение, какое именно изменение повлияло на поведение пользователя и его восприятие сайта. [Объяснение от Валеры почему это плохо.](#)

6. Функция inNamespace

Функция `inNamespace` проверяет, попадает ли пользователь в заданный диапазон Namespace, что определяет его участие в эксперименте. Диапазоны определяются функцией `getBucketRanges` (см в [Хэш функция используемая в Growthbook](#))

```
def inNamespace(userId: str, namespace: Tuple[str, float, float]) -> bool:
    n = ghash("___" + namespace[0], userId, 1)
    if n is None:
        return False
    return n >= namespace[1] and n < namespace[2]
```

Параметры функции inNamespace:

- `userId`: Идентификатор пользователя, для которого проверяется принадлежность к Namespace.
- `namespace`: Кортеж, содержащий три элемента:
 - `namespace[0]`: Идентификатор Namespace, который добавляется к seed в функции хеширования для уникализации хеша в контексте этого Namespace.
 - `namespace[1]`: Начальное значение диапазона Namespace (от 0 до 1), определяющее минимальное значение хеша, при котором юзер считается включенным в выбранный Namespace.
 - `namespace[2]`: Конечное значение диапазона Namespace (от 0 до 1), определяющее максимальное значение хеша, при котором юзер считается включенным в выбранный Namespace.

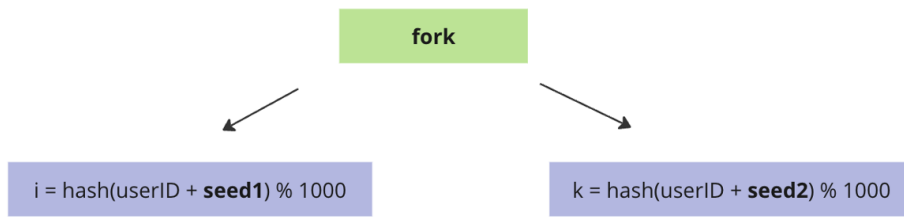
7. Как работает namespace для многомерных схем?

В **многомерной схеме** пользователь может попасть сразу в несколько экспериментов. Если пользователь одновременно участвует в нескольких экспериментах, то такие эксперименты называются **пересекающимися**.

Условия для многомерных тестов:

1. Тесты из разных слоев независимые.
2. Ортогональность тестов.

При использовании многомерных схем A/B тестирования с различными namespace-ами для каждого пользователя генерируется **отдельное хеш-значение для каждого namespace**. Это означает, что для одного и того же пользователя будут созданы разные хеш-значения в зависимости от пространства имен, в котором он участвует.



Таким образом, из вышеописанного можно заключить: для каждого слоя нужна своя уникальная соль (namespace), для каждого эксперимента нужна своя соль (feature flag).