



## Урок Спринт 2.12

# Как выбрать стратегию кеширования

## Аннотация

*В этом уроке мы рассмотрим различные стратегии кеширования, их применение в реальных системах и обсудим, как использовать кеш для оптимизации работы приложений.*

Первый вопрос, который нужно задать перед внедрением кеширования: "Какие данные мне кешировать?" Это зависит от того, как часто данные запрашиваются. Есть "горячие" данные, к которым обращаются регулярно, и "холодные" данные, запросы к которым происходят редко. Кеширование горячих данных делает ваше приложение быстрее и эффективнее, тогда как кеширование холодных данных может не принести значительных преимуществ.

В программировании существуют различные стратегии кеширования. Самые популярные — это Cache-Aside и Read-Through.

**Cache-Aside** подход подразумевает, что приложение сначала проверяет наличие данных в кеше, и если их там нет, то загружает их из основного источника и сохраняет в кеше.

```
func getData(key string) Data {  
    data, found := cache.Get(key)  
    if !found {  
        // Загрузка из базы данных и сохранение в кэше  
        data = loadDataFromDB(key)  
        cache.Set(key, data, cache.DefaultExpiration)  
    }  
    return data  
}
```

**Read-Through** подход предполагает, что кеш самостоятельно обрабатывает загрузку данных из первичного источника, если они отсутствуют в кеше.

## Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»