

- **Cache-Aside**

Cache-Aside – это паттерн кеширования данных, который часто используется в сочетании с API для улучшения производительности и снижения нагрузки на базу данных.

Cache-Aside предназначен для хранения результатов часто запрашиваемых операций чтения в кеше, что уменьшает нагрузку на базу данных и ускоряет время отклика API. При изменении данных в базе данных кеш инвалидируется(обновляется), и при следующем запросе данные снова загружаются из базы данных и сохраняются в кеше.

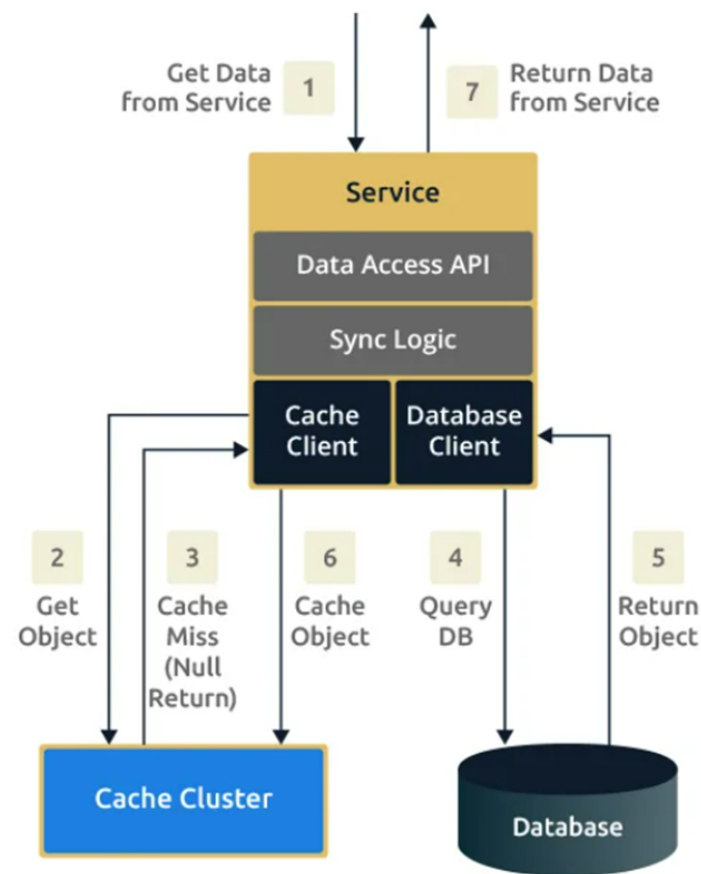
Основные характеристики:

1. Кеширование данных на стороне клиента или сервера для ускорения операций чтения.
2. Инвалидация кеша при изменении данных, чтобы поддерживать актуальность данных.
3. Загрузка данных в кеш при первом запросе, если они отсутствуют в кеше.
4. Возможность настройки времени жизни кеша для управления актуальностью данных и использования ресурсов.

Недостатки:

1. Сложность реализации и поддержки, так как требуется управлять кешем и обрабатывать инвалидацию(обновление) данных.
2. Возможность несогласованности данных из-за задержек при инвалидации(обновления) кеша или различных настроек времени жизни кеша.
3. Дополнительные затраты на инфраструктуру кеширования и управление ресурсами.
4. Снижение производительности при частых изменениях данных, когда кеш часто инвалидируется, и данные постоянно загружаются из базы данных.
5. Дополнительная точка отказа в системе.

Cache-Aside может быть полезным паттерном для улучшения производительности API и снижения нагрузки на базу данных, однако его использование может привести к сложности реализации и возможным проблемам с консистентностью данных. Применяйте этот паттерн, когда ваши данные не изменяются часто, и вы хотите ускорить операции чтения.



При получении запроса мы сначала проверяем кеш (1 и 2). Если в кеше есть нужное значение — отдаём клиенту, если нет (как на картинке шаг 3), тогда идем в бд забирать значение (4 и 5), записываем его в кеш(6) и потом уже отдаем клиенту(7).

Когда возможно нужно применять этот паттерн:

- Когда операции чтения значительно преобладают над операциями записи.

- Когда важна высокая производительность и быстрое время отклика API.
- Когда снижение нагрузки на базу данных является приоритетом.
- Когда данные не меняются слишком часто, и возможная несогласованность данных в кеше является приемлемой.

Когда возможно не нужно применять этот паттерн:

- Когда данные часто меняются, и инвалидация кеша будет происходить слишком часто.
- Когда требуется строгая консистентность данных между кешем и базой данных.
- Когда проект не имеет достаточных ресурсов для поддержки инфраструктуры кеширования.
- Если существующая инфраструктура и без применения Cache-Aside может обеспечить необходимую производительность и быстродействие.

Реальный пример:

Такой паттерн использует компания Twitter. Twitter имеет высокую нагрузку на операции чтения, и многие пользователи получают одну и ту же информацию (например, популярные твиты). Twitter использует Cache-Aside для хранения результатов часто запрашиваемых операций чтения, что уменьшает нагрузку на базу данных и улучшает производительность сервиса для миллионов пользователей.