

# ?? ????? ?? redis ??? ??? ??????? ?? ?????????????????

Το redis είναι μια in-memory key-value βάση δεδομένων η οποία χρησιμοποιείτε κυρίως για διαδικασίες caching. Ως cache την χρησιμοποιούμε και εμείς σε σελίδες όπως το mydata.semantic.gr & ergani.emblem.gr. Δεδομένα που παραμένουν συχνά ίδια μπορούμε να τα βάλουμε στο Redis ώστε αντί να γίνεται επικοινωνία με τη βάση δεδομένων και με το σκληρό δίσκο κατεπέκταση να γίνεται το διάβασμα των δεδομένων απο τη μνήμη η οποία είναι πολύ πιο γρήγορη.

Επιπλέον αν έχουμε πολλαπλές εφαρμογές που μπορεί να βρίσκονται σε άλλους σερβερ και πρέπει να επικοινωνήσουμε με τον κεντρικό σέρβερ αν έχεις ένα redis μπορείς να κρατήσεις "ζεστά" τα δεδομένα που σε ενδιαφέρουν άμεσα χωρίς επικοινωνία με τη βάση.

Ειδικά σε σερβερ με πολύ διαθέσιμη RAM είναι αρκετά εύκολη επιλογή το Redis για ελάφρυνση του φόρτου της βάσης δεδομένων και επιτάχυνση της εφαρμογής. Παρακάτω είναι η τεχνοτροπία που χρησιμοποιείτε στο ergani.emblem.gr για να διαβάζουμε και να γράφουμε στη cache

```
public function getLimitedNotificationsByUser(int $iduser): array
{
    $redisClient = $this->getRedisClient();
    $cacheKey = "NotificationsModel:getLimitedNotificationsByUser:" . $iduser . ":" .
($ _SESSION['usertype'] ?? 'prod');
    $cacheTTL = 1800; // seconds
    $cachedData = $redisClient->get($cacheKey);

    if ($cachedData !== false) {
        return json_decode($cachedData, true);
    }

    $arr = $this->SelectSqlSlave("SELECT * FROM notifications WHERE iduser = ? ORDER BY
seen ASC, id DESC LIMIT 30 ", ['i', $iduser]);

    $redisClient->setex($cacheKey, $cacheTTL, json_encode($arr));

    return $arr;
}
```

Στο cache key βάζουμε όλες τις μεταβλητές της λειτουργίας ώστε να μην cacharουμε δεδομένα απο άλλους χρήστες και πάρουμε δεδομένα απο άλλους. Ελέγχουμε αρχικά αν υπάρχουν στην cache τα δεδομένα βάση κλειδιού και αν υπάρχουν επιστρέφουμε τα δεδομένα της cache. Διαφορετικά κάνουμε το select στη βάση και επιστρέφουμε τα δεδομένα της βάσης. Τέλος εφόσον πήραμε τα δεδομένα απο τη βάση τα κάνουμε cache.

Είναι σημαντικό σε περίπτωση που γίνουν αλλαγή στα δεδομένα να κάνουμε invalidate της cache

```
public function add(int $iduser, string $message): array
{
    $redisClient = $this->getRedisClient();
    $cacheKey = "NotificationsModel:getLimitedNotificationsByUser:" . $iduser . ":" .
($ _SESSION['usertype'] ?? 'prod');
    $redisClient->del($cacheKey);

    return $this->otherSql("INSERT INTO notifications (type, iduser, seen, message) VALUES
(?,?,?,?)", ["siis", "Info", $iduser, 0, $message]);
}
```

---

Revision #2

Created 2025-12-11 12:21:33 UTC by Admin

Updated 2025-12-11 12:32:02 UTC by Admin