

```

import { supabaseAdmin } from '../supabase/admin';

export async function checkRateLimit(
  userId: string,
  rateLimitMaxRequests = 100,
): Promise<boolean> {
  // Fetch user's existing request count and last request timestamp
  const { data: userLimitData, error } = await supabaseAdmin
    .from('swarms_cloud_rate_limits')
    .select('*')
    .eq('user_id', userId)
    .limit(1)
    .single();

  if (error) {
    if (error.code === 'PGRST116') {
      // No rows returned, insert a new record
      const currentTime = Date.now();
      await supabaseAdmin.from('swarms_cloud_rate_limits').insert([
        {
          user_id: userId,
          request_count: 1,
          last_request_at: new Date(currentTime).toISOString(),
        },
      ]);
    }
  }

  return true;
}

```

```

    } else {

        console.error('Error fetching rate limit data:', error);

        return false;

    }

}

const currentTime = Date.now();

const lastRequestAtString = userLimitData?.last_request_at;

// Check if user is in a new minute

const currentMinute = new Date(currentTime).getMinutes();

const lastRequestMinute = lastRequestAtString

    ? new Date(lastRequestAtString).getMinutes()

    : -1; // -1 if no previous request

// Reset request count if new minute or no previous request

if (currentMinute !== lastRequestMinute) {

    await supabaseAdmin

        .from('swarms_cloud_rate_limits')

        .update({

            request_count: 1,

            last_request_at: new Date(currentTime).toISOString(),

        })

        .eq('user_id', userId);

} else {

    // Update request count (already incremented in previous logic)

```

```
await supabaseAdmin
  .from('swarms_cloud_rate_limits')
  .update({
    request_count: (userLimitData.request_count ?? 0) + 1,
  })
  .eq('user_id', userId);
}

// Check if user has exceeded request limit
if (
  userLimitData.request_count &&
  userLimitData.request_count >= rateLimitMaxRequests
) {
  console.log('User exceeded rate limit:', userId);
  return false;
}

return true;
}
```