

```
'use server';
```

```
import Decimal from 'decimal.js';
```

```
import {
```

```
  getUserCredit,
```

```
  getUserCreditPlan,
```

```
  supabaseAdmin,
```

```
} from '../supabase/admin';
```

```
import { getOrganizationOwner } from './organization';
```

```
export async function checkRemainingCredits(
```

```
  userId: string,
```

```
  organizationPublicId: string | null,
```

```
) {
```

```
  let id = userId;
```

```
  if (organizationPublicId) {
```

```
    // Fetch the organization owner's user ID
```

```
    const orgId = await getOrganizationOwner(organizationPublicId);
```

```
    if (!orgId)
```

```
      return {
```

```
        status: 400,
```

```
        remainingCredits: 0,
```

```
        message: 'Organization owner not found',
```

```
        credit_plan: 'default',
```

```
      };
```

```

    id = orgId ?? '';
  }

const [credits, credit_plan] = await Promise.all([
  getUserCredit(id),
  getUserCreditPlan(id),
]);

const remainingCredits = credits.credit + credits.free_credit;

if (remainingCredits <= 0 && credit_plan === 'default') {
  const status = organizationPublicId ? 402 : 400;
  const message = organizationPublicId
    ? 'Insufficient organization credits'
    : 'No remaining credits';

  return {
    status,
    message,
    remainingCredits: 0,
    credit_plan,
  };
}

return {

```

```
    status: 200,  
    remainingCredits,  
    message: 'Success',  
    credit_plan,  
  };  
}
```

```
export async function calculateRemainingCredit(  
  totalAPICost: number,  
  userId: string,  
  organizationPublicId: string | null,  
) : Promise<  
  | {  
    status: number;  
    message: string;  
  }  
  | undefined  
> {  
  if (totalAPICost <= 0) return;  
  
  let id = userId;  
  
  if (organizationPublicId) {  
    // Fetch the organization owner's user ID  
    const orgId = await getOrganizationOwner(organizationPublicId);  
    if (!orgId)
```

```
return {  
  status: 500,  
  message:  
    'Internal server error - invoice organization not found for calc rem credits',  
};
```

```
id = orgId ?? "  
}
```

```
try {  
  const { credit, free_credit } = await getUserCredit(id);  
  
  const decimalTotalAPICost = new Decimal(totalAPICost);  
  let newCredit = new Decimal(credit);  
  let newFreeCredit = new Decimal(free_credit);  
  
  if (newFreeCredit.greaterThan(0)) {  
    if (newFreeCredit.greaterThanOrEqualTo(decimalTotalAPICost)) {  
      newFreeCredit = newFreeCredit.minus(decimalTotalAPICost);  
    } else {  
      const remainingCost = decimalTotalAPICost.minus(newFreeCredit);  
      newFreeCredit = new Decimal(0);  
      newCredit = newCredit.minus(remainingCost);  
    }  
  } else {  
    newCredit = newCredit.minus(decimalTotalAPICost);
```

```
}
```

```
await supabaseAdmin.from('swarms_cloud_users_credits').upsert(
```

```
{
```

```
  user_id: id,
```

```
  credit: Number(newCredit),
```

```
  free_credit: Number(newFreeCredit),
```

```
},
```

```
{
```

```
  onConflict: 'user_id',
```

```
},
```

```
);
```

```
console.log('Remaining credit calculated successfully');
```

```
return {
```

```
  status: 200,
```

```
  message: 'Remaining credit calculated successfully',
```

```
};
```

```
} catch (error) {
```

```
  console.error('Error calculating remaining credit:', error);
```

```
  return { status: 500, message: 'Internal server error - Calc rem credits' };
```

```
}
```

```
}
```