

```

import { trpc } from '../utils/trpc/trpc';

import { getStripe } from '../utils/stripe/client';

import { useMemo } from 'react';

/**
 * Custom hook for managing subscription-related functionality.
 *
 * This hook provides methods and data for handling subscription status, creating subscription
 * portals, and managing user credits.
 *
 * @returns An object containing methods and data related to subscription management.
 */

const useSubscription = () => {

  const chargeAccountPortal =

    trpc.payment.createStripePaymentSession.useMutation();

  const userCredit = trpc.panel.getUserCredit.useQuery();


  const openChargeAccountPortal = () => {

    chargeAccountPortal.mutateAsync().then((url) => {

      document.location.href = url;

    });

  };


  const getSubscription = trpc.payment.getSubscriptionStatus.useQuery();

  const makeCustomerPortal = trpc.payment.createStripePortalLink.useMutation();

  const makeSubscriptionSession =

```

```

trpc.payment.createSubscriptionCheckoutSession.useMutation();

const createSubscriptionPortal = () => {
  makeSubscriptionSession.mutateAsync().then(async (sessionId) => {
    const stripe = await getStripe();
    if (stripe) stripe.redirectToCheckout({ sessionId });
  });
};

const openCustomerPortal = () => {
  makeCustomerPortal.mutateAsync().then((url) => {
    document.location.href = url;
  });
};

return {
  credit: userCredit.data,
  creditLoading: userCredit.isLoading,
  data: getSubscription,
  statusLoading: getSubscription.isLoading,
  status: getSubscription.data?.status,
  isLoading: getSubscription.isLoading,
  isSubscribed: getSubscription.data?.status === 'active' ?? false,
  isCanceled: getSubscription.data?.isCanceled,
  createSubscriptionPortalLoading: makeSubscriptionSession.isPending,
  createSubscriptionPortal,
  openCustomerPortalLoading: makeCustomerPortal.isPending,

```

```
openCustomerPortal,  
openChargeAccountPortalLoading: chargeAccountPortal.isPending,  
openChargeAccountPortal,  
};  
};  
  
export default useSubscription;
```