

recharge

Enhanced Token Security Guide



Table of Contents

Overview	3
Update Shopify Storefront theme	4
Update Recharge Theme Engine theme	9



Overview

Recharge is launching a mandatory token security enhancement, therefore all customer portal links must contain **?token=xxxxxxx** at the end of the URL.

This guide will provide you with steps on how to update **Shopify Storefront** and **Recharge Theme Engine** themes, to update the code so it would leverage token security enhancement.



Update Shopify Storefront theme

By default, the Manage subscriptions redirect link to Recharge customer portal is stored in `templates/customers/account.liquid` asset on Shopify.

No action on your Storefront is required if:

- You did not add any custom code to the “Manage subscriptions” link
- You did not add additional hard-coded links pointing to the Recharge customer portal

Therefore, if your link looks like one of the two following examples, and you have no other links like this throughout your shopify portal, no action is required on the Storefront end.

Example 1 of unmodified “Manage Subscriptions” link:

```
<!-- Begin ReCharge code -->
<p id="rc_link_container" ><a href="/tools/recurring/login/">Manage Subscriptions</a></p>
<!-- End ReCharge code -->
```

Example 2 of unmodified “Manage Subscriptions” link:

```
<!--BeginRechargeCode-->
<p><a href="/tools/recurring/customer/{{customer.metafields.subscriptions.customer_string}}/subscriptions/">Manage Subscriptions</a></p>
<!--EndRechargeCode-->
```

(* If you have checked your customer portal links and you’re not sure if there were any changes made to it - let us know and our team will be happy to help and take a look at your code)

If you’ve successfully confirmed that no action is required on the Shopify Storefront end, please see the second chapter of this guide **“Update Recharge Theme Engine theme”** and make sure to update Theme Engine routes according to the instructions!



Action on your Storefront is required if:

- Your store has hardcoded links pointing to the Recharge customer portal
- You have modified the “Manage subscriptions” link in the [account.liquid](#) asset.

In which case you need **to update those links to include the token** at the end of the URL.

A link without the new temporary token is going to redirect the customer straight to the Login page. (I.e. if a customer has bookmarked their customer portal page, it will now redirect to the Login page).

How to update Shopify Storefront routes?

Recharge’s API endpoint will allow you to retrieve customer temporary tokens.

NOTE: API endpoint will work only if you’re using Shopify templates and the customer is logged in in Shopify.

All links pointing to the Recharge customer portal will need to dynamically include the token retrieved from the API.

API route returns security token and customer hash in the response that are used to generate secure links across the Recharge customer portal.

You would need to extract token and customer hash as JS object from `/tools/recurring/get_rc_token/<shopify_customer_id>` response:

NOTE: In Shopify templates, `<shopify_customer_id>` is contained within the [customer object](#) and can be accessed as `{{ customer.id }}`

Route:

```
https://storename.myshopify.com/tools/recurring/get_rc_token/<shopify_customer_id>
```

Method: [GET]**Parameters:**

[shopify_customer_id] - ID of the customer in Shopify

Regex pattern for extracting the token and customer hash from response:

```
/{"rc_customer":.*}/
```

Response:

```
{
  rc_customer: {
    token: "9bf78ab05ae6453a222fe61bcf864cc8",
    hash: "e09d0ea8f2237ff1"
  }
}
```

Example code requests:jQuery

```
$.get("https://storename.myshopify.com/tools/recurring/get_rc_token/<shopify_customer_id>", function( data ) {
  const rc_customer = JSON.parse(data.match(/{"rc_customer":.*}/g)[0]);
  console.log(rc_customer);
});
```

fetch

```
fetch("https://storename.myshopify.com/tools/recurring/get_rc_token/<shopify_customer_id>")
  .then(res => res.clone().text())
  .then(data => {
```



```
const rc_customer = JSON.parse(data.match(/{"rc_customer":.*}/g)[0]);
console.log(rc_customer);
});
```

[axios](#)

```
axios.get("https://storename.myshopify.com/tools/recurring/get_rc_token/<shopify_customer_id>")
.then(function ({data}) {
  // handle success
  const rc_customer = JSON.parse(data.match(/{"rc_customer":.*}/g)[0]);
  console.log(rc_customer);
})
.catch(function (error) {
  // handle error
  console.log(error);
})
.then(function () {
  // always executed
});
```

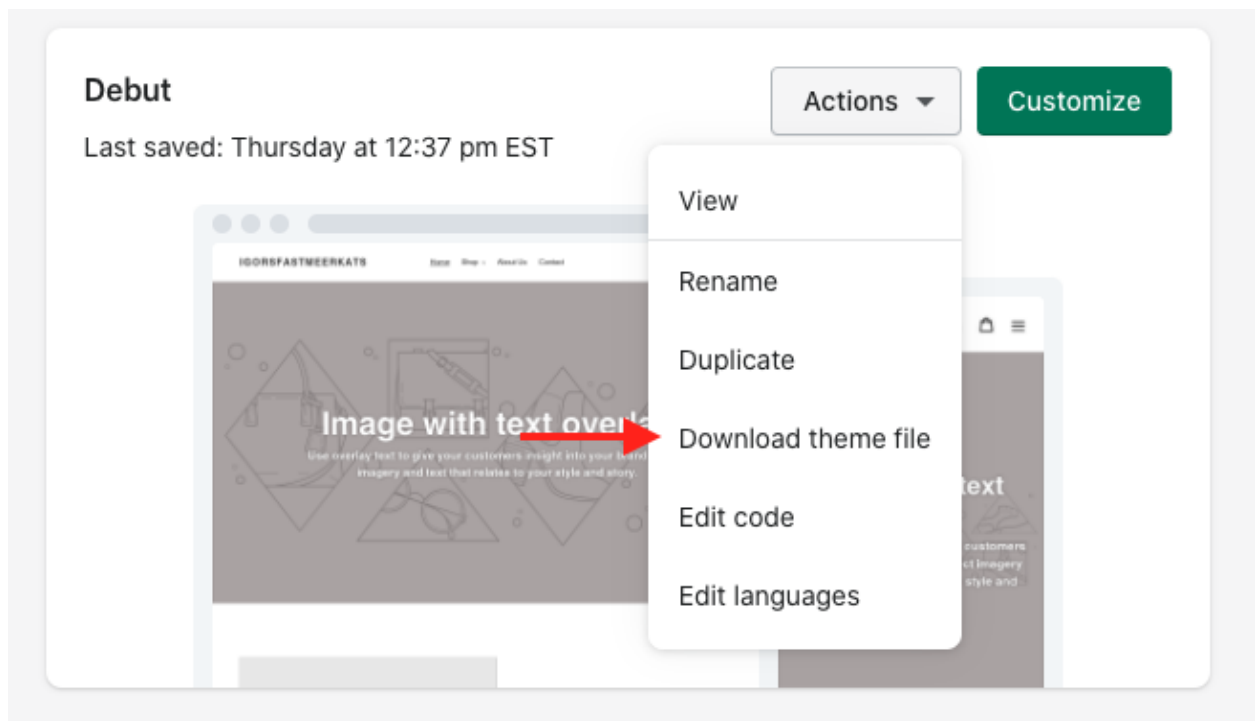
Steps how to retrieve token and customer hash using Recharge API endpoint:

- Log in as a user on the Shopify store
 - * if not logged in, `{rc_customer:{error:"you_are_not_logged_in"}}` will be returned.
- Send an AJAX call from any page within the store
- One of the 4 `rc_customer` responses will be returned:
 1. `<shopify_customer_id>` was not found in the database:
`{rc_customer:{error:"customer_not_found"}}`



2. not logged in on Shopify:
`{rc_customer:{error:"you_are_not_logged_in"}}`
3. <shopify_customer_id> you passed via the URL matches the one of the logged in user :
`{rc_customer:{token:"9bf78ab05ae6453a222fe61bcf864cc8", hash:"e09d0ea8f2237ff1"}}`
4. <shopify_customer_id> you passed via the URL **DOES NOT** match the one of the logged in user:
`{rc_customer:{error:"customer_id_mismatch"}}`

Updating Shopify Storefront -> Recharge routes:



1. Download theme files.
2. Open the theme folder in your preferred code editor.
3. Perform a global search on theme assets to locate all links to customer portal (**/tools/recurring/***) that don't include the token - VSCode has an integrated [global search](#) with a straightforward UI.
4. Update the links to include the token, as detailed above.

Update Recharge Theme Engine

If you have any custom javascript code inside the Theme Engine theme, that includes **GET** or **POST** request, please make sure that url contains token param. With Enhanced Token Security enabled, it is required, otherwise the request would not be valid.

By default, a token will be stored in a window object (`window.customerToken`) in the `base.html` template file of the TE theme.

You would need to append token param to your URL

```
URL + `?token=${window.customerToken}`
```

How to find the token?

- Inside Recharge app, click on menu and click on Theme Editor
- You will see a list of themes (published theme will have a badge *current theme*)
- Click on wrench icon on the right side, and click on Edit code
- Theme Editor should appear with the list of templates on the left side
- Click on `base.html` template file
- Inside `base.html` file look for

```
window.customerToken = getToken();
```

- `window.customerToken` is the value of the customer token

How to append the token to the URL?

- Inside Theme Engine template files, find the custom javascript **GET** or **POST** request

URL **without** token:

```
GET request
```

```
"{{ shopify_proxy_url if proxy_redirect else "" }}/portal/{{
```



```
customer.hash }}/request_objects"
```

```
"/tools/recurring/portal/{{ customer.hash }}/request_objects"
```

POST request:

```
"/tools/recurring/portal/{{ customer.hash }}/subscriptions/"
```

Request **without** token:

```
$.ajax({  
  url: "{{ shopify_proxy_url if proxy_redirect else "" }}/portal/{{  
customer.hash }}/request_objects",  
  type: "get",  
  dataType: "json",  
  data: {  
    schema:  
      '{ "subscriptions": { "product": {} } }',  
  },  
})  
.done(function (response) {  
  // Successful request made  
  console.log(response);  
})  
.fail(function (response) {  
  // Request failed  
  console.log(response);  
});
```

URL **with** token:

GET request

```
"{{ shopify_proxy_url if proxy_redirect else "" }}/portal/{{  
customer.hash }}/request_objects" + '?token=' + window.customerToken  
  
"/tools/recurring/portal/{{ customer.hash }}/request_objects" +
```



```
'?token=' + window.customerToken
```

POST request

```
"/tools/recurring/portal/{{ customer.hash }}/subscriptions/" +  
'?token=' + window.customerToken
```

Request **with** token:

```
$.ajax({  
  url: "{{ shopify_proxy_url if proxy_redirect else "" }}/portal/{{  
customer.hash }}/request_objects" + '?token=' + window.customerToken,  
  type: "get",  
  dataType: "json",  
  data: {  
    schema:  
      '{ "subscriptions": { "product": {} } }',  
  },  
})  
.done(function (response) {  
  // Successful request made  
  console.log(response);  
})  
.fail(function (response) {  
  // Request failed  
  console.log(response);  
});
```

- Append **window.customerToken** at the end of url

What if there is no window.customerToken in the [base.html](#) template file?

- If there is no such code, you would need to insert the following code at the top of the [base.html](#) file.



```
<script>
  function getToken() {
    const value = new RegExp("[?&]token=([^&#]*)",
    "i").exec(window.location.href);
    return value ? value[1] : '';
  }

  window.customerToken = getToken();
</script>
```

- Once the code is implemented, you would have access to the customer token value, stored in **window.customerToken**
- Inside Theme Engine template files, find the custom javascript **GET** or **POST** request
- Append **window.customerToken** at the end of url

