**Introduction**  
  
SharePoint 2013 (and previous versions) uses a client side “token” to validate posts back to SharePoint to prevent attacks where the user might be tricked into posting data back to the server. This token is known by many names, form digest or message digest or request digest. The token is unique to a user and a site and is only valid for a (configurable) limited time.  
  
When building Apps or customization on top of SharePoint, especially using patterns such as Single Page Applications (SPA) or using frameworks such as knockout.js it is very common that you see errors due to that the token is invalidated, which is due to that you have not reloaded the page and the token has timed out. The purpose of this article is to show you how you can refresh this form digest using JavaScript.  
  
**How to use the Request Digest token**  
  
When working with CSOM or REST you need to add the Request Digest token to your request. Well, with CSOM (JSOM) it is done for you under the hood, but when you handcraft your REST queries you need to manually add them. It usually looks something like this:  
  
$.ajax({  
    url: \_spPageContextInfo.siteAbsoluteUrl + "/\_api/web/,,,",  
    method: "POST",  
    headers: {  
        "Accept": "application/json; odata=verbose",  
        "X-RequestDigest": $('#\_\_REQUESTDIGEST').val()  
    },  
    success: function (data) {},  
    error: function (data, errorCode, errorMessage) {}  
});  
  
On line #6 we add the “X-RequestDigest” header, with the value of the hidden input field “\_\_REQUESTDIGEST” – which is the form digest value. I will not dig deeper into this since it’s part of basically every SharePoint POST/REST sample on the interwebs.  
  
But what happens if you’ve built a page (SPA application) where you don’t reload the page and the users work on the page for longer times than the token timeout (default 30 minutes). Then they will get exceptions like these:  
  
HTTP/1.1 403 FORBIDDEN  
{"error":{"code":"-2130575252, Microsoft.SharePoint.SPException","message":{  
"lang":"en-US",  
"value":"The security validation for this page is invalid and might be corrupted.  
Please use your web browser's Back button to try your operation again."}}}  
  
**How to refresh the token**  
  
So how do we get a new and updated token? There are multiple ways to refresh the token, or retrieve a new and updated one. The two most common ways are to either use ye olde web services and call into the /\_vti\_bin/sites.asmx and use the GetUpdatedFormDigest method. To use this you have to create a SOAP message and then parse the response and retrieve the updated token. You can then either pass that new token for your subsequent requests, or even better update the hidden \_\_REQUESTDIGEST field. The second one is to use the new REST POST endpoint /\_api/contextinfo and then parse that response which can be either XML or JSON. This is how to do it the JSON way:  
  
$.ajax({  
    url: \_spPageContextInfo.webAbsoluteUrl + "/\_api/contextinfo",  
    method: "POST",  
    headers: { "Accept": "application/json; odata=verbose"},  
    success: function (data) {  
        $('#\_\_REQUESTDIGEST').val(data.d.GetContextWebInformation.FormDigestValue)  
    },  
    error: function (data, errorCode, errorMessage) {  
        alert(errorMessage)  
    }  
});  
  
It’s also worth noticing that for every REST query SharePoint actually generates a new Form Digest which is sent back as a response header (X-RequestDigest), so you could always read that response header and update the form digest.  
  
Also, if you’re not using JavaScript but instead building an app using other frameworks, platforms, languages etc – you can always use the two aforementioned methods to update the token. Well, you actually need to do it since you don’t have any hidden input field :)  
  
**How to do it the best and native way**  
  
The drawbacks with the methods above is that you have to either request a new form digest all the time to make sure that it is up to date, or catch the exception and retry your query. And as we all know this will lead to bad performance and/or cluttered JavaScript code (like we don’t have enough of that already!).  
  
Fortunately there is a native method for refreshing the token built-in to the SharePoint JavaScript libraries. It’s a function called UpdateFormDigest(). It’s defined in INIT.js. The method takes two parameters, first of all the URL to the current site (remember the token is only valid for one site) and secondly it takes an update interval parameter. The update interval value is also already given to us, using a global constant called \_spFormDigestRefreshInterval. And this is how you should use the function:  
  
UpdateFormDigest(\_spPageContextInfo.webServerRelativeUrl, \_spFormDigestRefreshInterval)  
As you can see, it’s very simple, only use built-in stuff and there’s no magic to it. Under the hood this method uses the /\_vti\_bin/sites.asmx method and it does it synchronously. This means that all you have to do is copy and paste this line into your own code just before your REST calls. The other smart thing about this method is that is uses the update interval and only updates the form digest when needed – so no extra calls.  
  
Setting an interval of 5 minutes has it auto-refreshing and I don't have to remember to call it before my ajax.  
  
setInterval(function() {  
UpdateFormDigest(\_spPageContextInfo.webServerRelativeUrl, \_spFormDigestRefreshInterval);  
}, 5 \* 60000);