Peter Njeru (https://peternjeru.co.ke/blog/)

FOR MPESA API AND SYSTEM CONSULTATIONS, CONTACT PETER VIA HTTPS://CONSULT.PETERNJERU.CO.KE/

(https://twitter.com/Njeru__Peter)

M-Pesa / Daraja API Credentials

This is one of those articles you write when you have just had it, when like a hundred people keep asking the same question over and over again and can't seem to understand, or just don't want to do the work, and want to be spoon fed, or simply just can't understand what the heck to do about it or go about it. I hope this article clears as much of this as possible.

This is about M-Pesa's credentials, in the

Search …

SEARCH

RECENT POSTS

Protected: Automated M-Pesa Transaction Report Processing (https://peternjeru.co.ke/blog/automated-m-pesa-transaction-report-processing/)

Portal and outside on their Daraja API. And here, the word **Credential** should be taken to mean any form of code or security that seeks to authenticate the user to Daraja or M-Pesa, no matter the type or format, for the user to be able to access its services. This article is also very specific to Daraja.

M-Pesa basically has 2 sets of credentials:

- **Portal Credentials** – these are used by users on the M-Pesa portal, also known as operators.
- **API credentials** – these are credentials created for use in API calls to the M-Pesa API via Daraja.

There are other credentials on other older M-Pesa APIs, but they are not our focus here.

Portal Credentials

When you access the M-Pesa portal (located at https://org.ke.m-pesa.com/ (https://org.ke.m-pesa.com/)), you have a user whom you use to log in. That user has a password which you either assigned to them if you created them, or was given by Safaricom. That user is also known as an operator, and more specifically a **web operator**. That means they are allowed to access M-Pesa via the web portal. Their password represents the first type of security credential for M-Pesa, a simple password backed by a one time verification code for every login. This password is as simple as they come. The only requirement is that it has at least one special character, one capital letter and one digit. Note that M-Pesa does

M-Pesa / Daraja API Credentials (https://peternjeru.co.ke/blog/m-pesa-daraja-api-credentials/)
Pay via ProxyAPI on WooCommerce (https://peternjeru.co.ke/blog/pay-via-proxyapi-on-woocommerce/)
Pay Via Proxy API (https://peternjeru.co.ke/blog/pay-via-proxy-api/)
Safaricom Tills API Integrations (https://peternjeru.co.ke/blog/safaricom-tills-api-integrations/)

RECENT COMMENTS

How To Login To Mpesa Portal - LoginHulk (https://loginhulk.com/how-to-login-to-mpesa-portal/) on M-Pesa Portal Certificates (https://peternjeru.co.ke/blog/m-pesa-portal-certificates/#comment-32)
mpesa web portal - infopvp (https://infopvp.com/2022/04/23/mpesa-web-portal/) on M-Pesa Portal Certificates (https://peternjeru.co.ke/blog/m-pesa-portal-certificates/#comment-31)
How To Login To Mpesa Portal - Free-directorysubmit.com

not recognize the **'@'** character as a special character.

For a web operator, their password has an expiry period, around 3 months (last I checked). That means you gotta keep your password updated regularly, or you will be locked out and the web operator's status changed to **dormant**. When that happens, if you are **not** the Business Administrator (this is a role assigned to the first primary operator of a shortcode), you can have the Business Admin for your shortcode re-enable you, otherwise you gotta get in touch with Safaricom M-Pesa Business team to re-enable you. If you plan to use a web operator for API requests, always remember to renew the password before the expiry, otherwise all requests will fail with the "The Initiator Information Is Invalid" error.

Now, this tutorial assumes you want to integrate into the M-Pesa APIs. And that means you will need an **API operator**. This is the API equivalent of the web operator, meaning they are only allowed to perform API calls, and can never log into the web portal. Their passwords also never expire (last I checked). They are created by the Business Admin of a shortcode to enable the business owner to perform API transactions. Once created, they are then also known by their API name: **Initiators**. Thus their assigned usernames become known as **Initiator Names** and the assigned passwords become **Initiator Passwords**. Remember you have to assign them the right roles to be able to perform the required API calls.

(https://free-directorysubmit.com/how-to-login-to-mpesa-portal/) on M-Pesa Portal Certificates (https://peternjeru.co.ke/blog/m-pesa-portal-certificates/#comment-30)
M Pesa Login - Login - l-pesa.com (https://corashack.com/m-pesa-login/) on M-Pesa Portal Certificates (https://peternjeru.co.ke/blog/m-pesa-portal-certificates/#comment-29)
Peter Njeru on About (https://peternjeru.co.ke/blog/about/#comment-13)

ARCHIVES

February 2022 (https://peternjeru.co.ke/blog/2022/02/)
April 2020 (https://peternjeru.co.ke/blog/2020/04/)
February 2020 (https://peternjeru.co.ke/blog/2020/02/)
January 2020 (https://peternjeru.co.ke/blog/2020/01/)
September 2019 (https://peternjeru.co.ke

For the API operators/Initiators, their passwords represent the second level of credentials for M-Pesa. The passwords need to be encrypted using an RSA (https://en.wikipedia.org/wiki/RSA_(cryptosystem))Public Key certificate provided by Safaricom before being used in an API. The Certificates are available in PEM on the developer portal documentation (https://developer.safaricom.co.ke/docs). Once encrypted, the resulting cipher-text (https://en.wikipedia.org/wiki/Ciphertext) is then what is used as the `SecurityCredential` or `Initiator Encrypted Password` in the body of API requests to Daraja and other M-Pesa APIs e.g. in the Daraja API B2C Sample request below, the `SecurityCredential` parameter contains the cipher-text resulting from the encryption process described mentioned above:

```
{
    "InitiatorName": "apitest361",
    "SecurityCredential": "iDIurkC
G4r9F7lCCNNkctseY9MgiddL6F1VmmJH1D
Sk5n5U4HPkE3gj1+UFNQ+3MUIzIkR7EgBo
v9/rqzeXJVb8JeTTD/10XSdTU5eJJeWsqI
1VSXpIe/j4xa7HNuezxRJ/KtTWu3bdq9he
ozyXMn/5ErogjfwIlfQ4Bkhw8+9SEGoNkR
A95Idp1PSg/ElPOVnrKuPFTf0HS/iTmuKv
VBkY6oAFSObVx7DmqqoL2K4P8ZNh3EDVfQ
tCQGUnF1OLOab5X46wqVW5p7UJ/5kVFdN8
Yuw172VNOjyNILjUXwqqAhS4WKXC28cXd4
faYrG5E5AQfGGjNIlYRqWse8w23fw==",
    "CommandID": "[CommandID]",
    .... //other data
}
```

That is the **ONLY** place you ever use an **encrypted password** in API requests to M-Pesa. Nowhere else! Remember that.

/blog/2019/09/)
August 2019 (https://peternjeru.co.ke/blog/2019/08/)
July 2019 (https://peternjeru.co.ke/blog/2019/07/)
January 2019 (https://peternjeru.co.ke/blog/2019/01/)

CATEGORIES

daraja (https://peternjeru.co.ke/blog/category/daraja/)
mpesa (https://peternjeru.co.ke/blog/category/mpesa/)
Uncategorized (https://peternjeru.co.ke/blog/category/uncategorized/)

Those are the two sets of portal credentials you can expect to work with on the M-Pesa portal side. Moving to the APIs…

API Credentials

When you get to Daraja, you are expected to authenticate your API calls to Daraja (if you don't know what am talking about please take a detour (https://peternjeru.co.ke/safdaraja/ui/)). Thus authentication on Daraja takes us to the other set of credentials: API credentials.

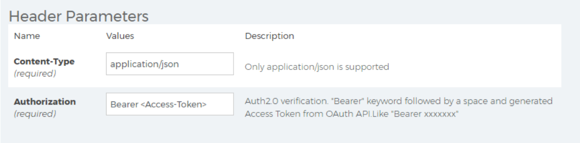
Note that the API operator/initiator credentials are created on the portal, but used on the APIs only.

When sending any request to Daraja, you need to send a specific header as part of the headers. This is the **Authentication (https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Authorization)** header. For almost all API calls to Daraja, you need to first fetch an access token that will be authenticated against the API. To get this token you use the Generate Token (https://developer.safaricom.co.ke/APIs/Authorization) API. This API requires you to send the same Authentication header parameter to get the access token but with a different value. That header is an M-Pesa API credential, and is made up of base64 encoded **consumer key** and **consumer secret** from an App (https://developer.safaricom.co.ke/MyApps) of your choosing on the developer portal, joined together by a full colon (again, all that is here (https://peternjeru.co.ke/safdaraja/ui/)). That header must always be

META

Log in (https://peternjeru.co.ke/blog/wp-login.php)
Entries feed (https://peternjeru.co.ke/blog/feed/)
Comments feed (https://peternjeru.co.ke/blog/comments/feed/)
WordPress.org (https://wordpress.org/)

present in every request, under the Authentication key, and expires after 3600 seconds, or 1 hour. So keep in mind to refresh it regularly. Note that you **only** encode the consumer credentials, and that you only ever use the tokens in the **HTTP header**, nowhere else. The screenshot below shows where the header is applied when making an API request from the portal. The `<Access-Token>` variable is replaced with the result of the base64 encoding described above. Remember to always include the prefix required in the header e.g. the word `Bearer` in the screenshot, followed by a space, then the token. Otherwise requests will fail with an "Invalid Access Token" error



Header Parameters

Name	Values	Description
Content-Type *(required)*	application/json	Only application/json is supported
Authorization *(required)*	Bearer <Access-Token>	Auth2.0 verification. "Bearer" keyword followed by a space and generated Access Token from OAuth API.Like "Bearer xxxxxxx"

When used in actual API requests, the header is usually used in the format `Authorization: Bearer <Access-Token>`. So take note of how to include headers in your selected language / environment.

Last, but definitely not least, we have the Lipa na M-Pesa password. This is a credential that is **only** used in the body of Lipa na M-Pesa requests. To use or apply this credential, you first need to get a passkey from Safaricom, or if on the sandbox, use the provided passkey (this seems to no longer be the case for the new Daraja portal). Then create the password by base64-encoding your concatenated shortcode + passkey + timestamp. That gives you the value to use in the `Password` parameter of LnM requests. Sample usage is shown below:

```
{
        "BusinessShortCode": "174379",
        "Password": "MTc0Mzc5YmZiMjc5ZjlhYTliZGJjZjE1OGU5N2RkNzFhNDY3Y2QyZTBjODkzMDU5YjEwZjc4ZTZiNzJhZGExZWQyYzkxOTIwMjAwNDMwMTgzOTQ5",
        "Timestamp": "20200430183949",
        .... //other data
}
```

Note that the `Timestamp` value in the request must be the same as the timestamp used in the creation of the Password. Also the timestamp must be in the format **YYYYMMDDHHmmss**. Otherwise you also get the “MerchantValidate – Wrong credentials” error. The Date format pattern usually differs between languages, thus the base rules are:

- The year must be 4 digits long
- The Month should use its numeric value, not name e.g. January is **01**, not Jan, January or whatever else you are thinking about.
- The Day is the day of the month, between 1 and 31.
- The Month, Day, Hour, Minutes and Seconds must be 2 digits long, prefixed by 0 (zero-padding) where needed.
- The hour should be 24-hour format.

And that’s the only place you ever use the LnM password.

Thus, in summary, you pretty much always use either one or two credentials when working with M-Pesa APIs. If you ever find yourself using all three, just delete all your code and start over.

If I can summarize the APIs and their usage of the credentials, it would be as below:

API	M-Pesa Initiator Password	Authorization Header (Type)	LnM Password
Generate Token		✔ (Basic)	
Reversal	✔	✔ (Bearer)	
Transaction Status	✔	✔ (Bearer)	
Account Balance	✔	✔ (Bearer)	
B2C	✔	✔ (Bearer)	
C2B (Register URL)		✔ (Bearer)	
M-Pesa Express (LnM)		✔ (Bearer)	✔

I have tried and created sample codes on how to generate each type of credential for Java, PHP and JS. Unfortunately for JS I haven't found a way to encrypt the password from the server side, thus the available code only works on Browser side. If you have sample Node code to encrypt the M-Pesa Initiator password, please host on GitHub and share a link, I shall put it here for others to check out:

- Java (https://github.com /peternjeru/mpesa-encryption-

encoding-java)
- PHP (https://github.com/peternjeru/mpesa-encryption-encoding-php)
- JS (https://github.com/peternjeru/mpesa-encoding-js)

Again, if none of the above make sense, start here (https://peternjeru.co.ke/safdaraja/ui/).

Note that the certificate in the repos above is the G2 API certificate, thus will most probably not work with Daraja. Download a fresh copy of the certificates from the Daraja portal to work with on Daraja.

That should give you a good head start in that stuff. Also note the very generous use of the word **only** for those peeps who use a single credential everywhere they want. Now, I don't wanna hear anymore credentials questions.

For all M-Pesa APIs consultation, contact Peter via https://consult.peternjeru.co.ke/ (https://consult.peternjeru.co.ke/) or through the number (+254)705-740 384

Posted on April 30, 2020 (https://peternjeru.co.ke/blog/m-pesa-daraja-api-credentials/) by Peter Njeru (https://peternjeru.co.ke/blog/author/admin/)

PAY VIA PROXYAPI ON WOOCOMMERCE (HTTPS://PETERNJERU.CO.KE/BLOG/PAY-VIA-PROXYAPI-ON-WOOCOMMERCE/)

AUTOMATED M-PESA TRANSACTION REPORT PROCESSING (HTTPS://PETERNJERU.CO.KE/BLOG/AUTOMATED-M-PESA-TRANSACTION-REPORT-PROCESSING/)

RELATED ARTICLES

Protected: Automated M-Pesa Transaction Report Processing

(https://peternjeru.co.ke/blog/automated-m-pesa-transaction-report-processing/)

Pay via ProxyAPI on WooCommerce

(https://peternjeru.co.ke/blog/pay-via-proxyapi-on-woocommerce/)

Pay Via Proxy API

(https://peternjeru.co.ke/blog/pay-via-proxy-api/)

ALSO ON PETER NJERU

1 Comment 1 **Login ▾**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

♡ **Share**

Best **Newest** **Oldest**

F **Francis Wanyeki**

2 years ago

Find the node encryption code here

top

Theme: Techieblog by WP Crumbs (https://wpcrumbs.com/)