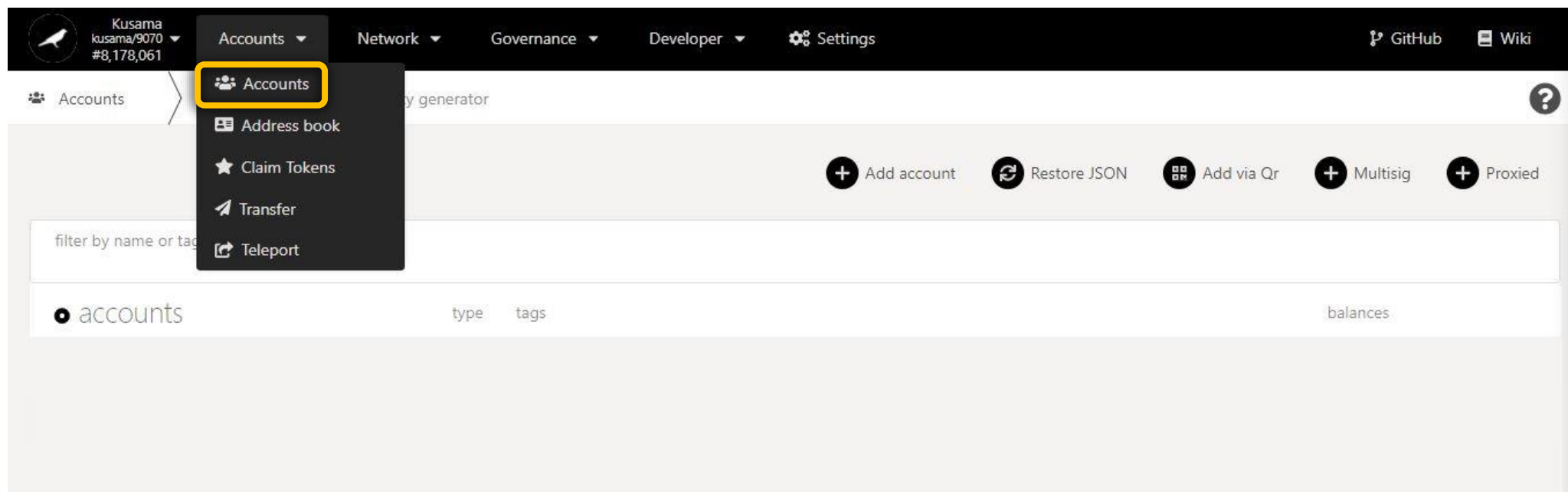


[PART I: Accounts \(Basic\)](#)

1. Accounts: Manage your accounts.



Basic information on technical words used in the ACCOUNTS section.

Options for creating/importing account(s).

Accounts can be **injected** from Polkadot-JS extension, **created** from Polkadot-JS Apps, or derived from **Ledger** hardware.

Any custom information/note about the account.

Existing account(s) enabled on the Kusama network.

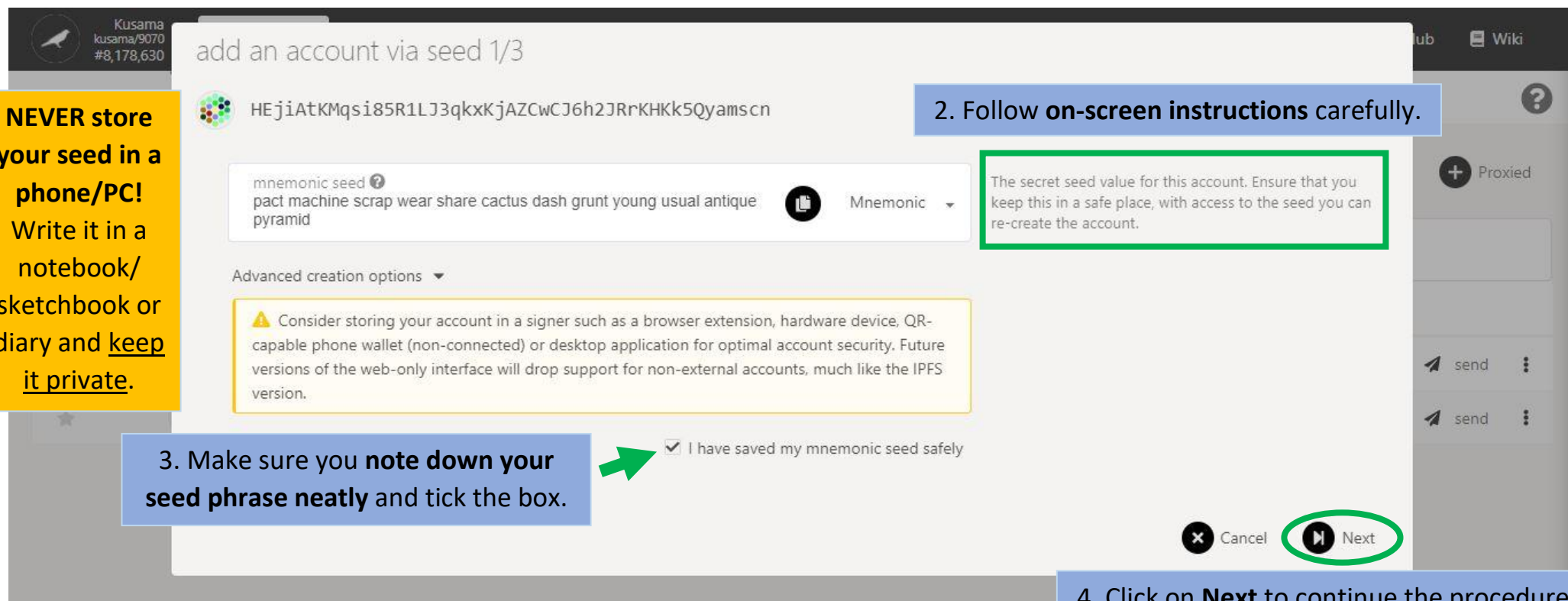
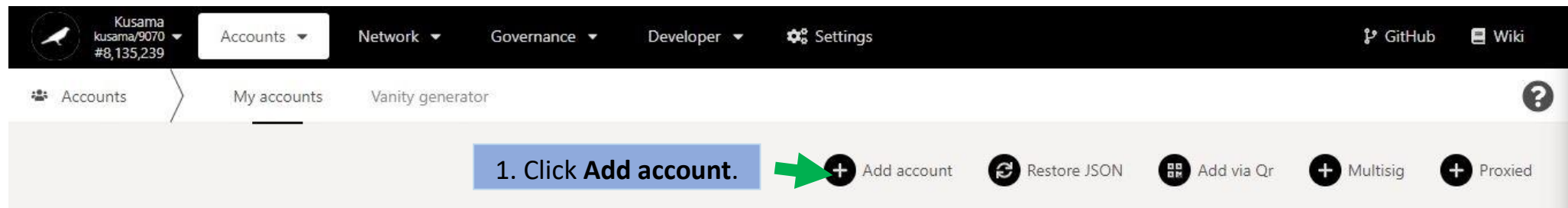
Overview of account's KSM balances.

The screenshot shows the Polkadot-JS Accounts interface. At the top, there's a navigation bar with 'Kusama kusama/9070 #8,135,239', 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. Below this, there's a 'My accounts' tab and a 'Vanity generator' link. A search bar 'filter by name or tags' is present. The main area displays a table of accounts. Two accounts are listed: 'ALTD@KSM (EXTENSION)' and 'ANAELLE LTD@KSM', both marked as 'injected' with 'no tags'. To the right of the accounts table, there's a 'balances' section showing three KSM balances: 10.3509 KSM, 1.0376 KSM, and 11.3886 KSM, each with a 'send' button. Annotations with yellow boxes and green arrows point to various parts of the interface: 'Basic information on technical words used in the ACCOUNTS section.' points to the top navigation bar; 'Options for creating/importing account(s).' points to the buttons 'Add account', 'Restore JSON', 'Add via Qr', 'Multisig', and 'Proxied'; 'Accounts can be injected from Polkadot-JS extension, created from Polkadot-JS Apps, or derived from Ledger hardware.' points to the 'injected' status of the accounts; 'Any custom information/note about the account.' points to the 'tags' column; 'Existing account(s) enabled on the Kusama network.' points to the account names; and 'Overview of account's KSM balances.' points to the balances section.

accounts	type	tags
ALTD@KSM (EXTENSION)	injected	no tags
ANAELLE LTD@KSM	injected	no tags

balances
10.3509 KSM
1.0376 KSM
11.3886 KSM

a) Create a Polkadot-JS account from a new Seed and save it in a JSON file.



GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2

add an account via seed 2/3

Account Information:

- Icon: LALA
- Seed: HEjiAtKMqsi85R1LJ3qkxKjAZCwCJ6h2JRrKHKk5Qyamsn

Form Fields:

- name: Lala
- password:
- password (repeat):

Strength Indicator: WEAK [Progress Bar] STRONG

Warning: ⚠ Consider storing your account in a signer such as a browser extension, hardware device, QR-capable phone wallet (non-connected) or desktop application for optimal account security. Future versions of the web-only interface will drop support for non-external accounts, much like the IPFS version.

Navigation: Cancel Prev **Next**

Annotations:

- Avoid using the same password for all your accounts.** (Yellow box pointing to password fields)
- 5. Follow the new instructions carefully.** (Blue box pointing to the form fields)
- 6. Click on Next to continue the procedure.** (Blue box pointing to the Next button)

Help Text (Green box):

- The name for this account and how it will appear under your addresses. With an on-chain identity, it can be made available to others.
- The password and password confirmation for this account. This is required to authenticate any transactions made and to encrypt the keypair.
- Ensure you are using a strong password for proper account protection.

add an account via seed 3/3

LALA
HEjiAtKMqsi85R1LJ3qkxKjAZCwCJ6h2JRrKHKk5Qyamsn

partial seed
pact wear dash usual

keypair type
sr25519

derivation path
<none provided>

We will provide you with a generated backup file after your account is created. As long as you have access to your account you can always download this file later by clicking on "Backup" button from the Accounts section.

Please make sure to save this file in a secure location as it is required, together with your password, to restore your account.

Cancel Prev **Save**

7. Follow the **final instructions** carefully.

8. Click on **Save** to complete the procedure.

GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2

The screenshot shows the 'Accounts' tab in the Polkadot-JS interface. The top navigation bar includes 'Kusama kusama/9070 #8,178,661', 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. Below the navigation bar, there are tabs for 'Accounts', 'My accounts', and 'Vanity generator'. The 'My accounts' tab is active, displaying a list of accounts. A yellow box with the text 'Star your most important accounts.' points to a star icon next to the 'LALA' account. A blue box with the text '9. Your account has been added to the *My accounts* list!' points to the 'LALA' account row. A blue box with the text '10. Click on the 3 vertical dots to view Account settings.' points to the three vertical dots icon next to the 'LALA' account. A yellow box with the text 'A JSON (backup) file has also been created for your account! It should be stored in a device that is not connected to the internet.' points to a file icon and the text 'HEjiAtKMqsi85R1....json' at the bottom left. The account list has columns for 'accounts', 'type', 'tags', and 'balances'. The 'LALA' account is of type 'sr25519' and has a balance of '0.0000 KSM'. The total balance shown is '11,4009 KSM'.

accounts	type	tags	balances
★ ALTD@KSM (EXTENSION)	injected	no tags	10.3633 KSM
★ ANAELLE LTD@KSM	injected	no tags	1.0376 KSM
★ LALA	sr25519	no tags	0.0000 KSM

11,4009 KSM

HEjiAtKMqsi85R1....json

GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2

The screenshot shows the 'Accounts' page in the Polkadot.js interface. The top navigation bar includes 'Kusama kusama/9070 #8,178,673', 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. The main content area has tabs for 'Accounts', 'My accounts', and 'Vanity generator'. Below the tabs, there are buttons for '+ Add account', 'Restore JSON', and 'Add via QR'. A search bar labeled 'filter by name or tags' is present. A table lists accounts with columns for a star icon, account name, type, and tags. The first account is 'ALTD@KSM (EXTENSION)' with type 'injected' and 'no tags'. A context menu is open for this account, showing options like 'Set on-chain identity', 'Derive account via derivation path', 'Create a backup file for this account', 'Change this account's password', 'Forget this account', 'Make recoverable', 'Initiate recovery for another', 'Delegate democracy votes', and 'Add proxy'. A blue callout box with a green arrow points to the 'Create a backup file for this account' option.

11. Click on **Create a backup file [...]** to download your account's JSON file.

	accounts	type	tags
★	ALTD@KSM (EXTENSION)	injected	no tags
★	ANAELLE LTD@KSM	injected	no tags
★	LALA	sr25519	no tags

- Set on-chain identity
- Derive account via derivation path
- Create a backup file for this account
- Change this account's password
- Forget this account
- Make recoverable
- Initiate recovery for another
- Delegate democracy votes
- Add proxy

only this network ☒

GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2

12. Follow on-screen instructions carefully.

backup account

LALA
HEjiAt...yamsn

An encrypted backup file will be created once you have pressed the "Download" button. This can be used to re-import your account on any other machine.

Save this backup file in a secure location. Additionally, the password associated with this account is needed together with this backup file in order to restore your account.

password ?
.....

Cancel Download

13. Click on **Download to complete the procedure.**

b) Import a Polkadot-JS account from a JSON file.

The screenshot shows the Polkadot-JS Accounts interface. At the top, there's a navigation bar with 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. Below this, there's a sub-navigation bar with 'Accounts', 'My accounts', and 'Vanity generator'. The main area displays a list of accounts, with a button to 'Add account'. A green arrow points to the 'Restore JSON' button. A blue box with the text '1. Click Restore JSON.' is positioned above the button. Below the 'Restore JSON' button, there's a dialog titled 'add via backup file'. The dialog has a text input field for the account name, a 'backup file' input field with a question mark icon and the text 'click to select or drag and drop the file here', and a 'password' input field with a question mark icon. A green box highlights the 'password' field and the text 'The password previously used to encrypt this account.' A yellow box contains a warning icon and text: 'Consider storing your account in a signer such as a browser extension, hardware device, QR-capable phone wallet (non-connected) or desktop application for optimal account security. Future versions of the web-only interface will drop support for non-external accounts, much like the IPFS version.' At the bottom right of the dialog are 'Cancel' and 'Restore' buttons. A blue box with the text '2. Follow on-screen instructions carefully.' is positioned above the dialog.

1. Click **Restore JSON**.

2. Follow **on-screen instructions** carefully.

add via backup file

5xxxxx...xxxxxx

backup file ?
click to select or drag and drop the file here

password ?

⚠ Consider storing your account in a signer such as a browser extension, hardware device, QR-capable phone wallet (non-connected) or desktop application for optimal account security. Future versions of the web-only interface will drop support for non-external accounts, much like the IPFS version.

Cancel Restore

GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2

The screenshot shows the 'add via backup file' dialog in the Polkadot-JS Accounts interface. The dialog is titled 'add via backup file' and features a LALA icon and the account name 'HEjiAt...yamscn'. It contains two input fields: 'backup file' with a help icon and a file path 'HEjiAtKMqsi85R1U3qkxKjAZCwCJ6h2JRrKHKk5Qyamscn.json (600 bytes)', and 'password' with a help icon and masked characters. To the right of these fields are explanatory texts: 'Supply a backed-up JSON file, encrypted with your account-specific password.' and 'The password previously used to encrypt this account.' Below the password field is a yellow warning box with a triangle icon and the text: 'Consider storing your account in a signer such as a browser extension, hardware device, QR-capable phone wallet (non-connected) or desktop application for optimal account security. Future versions of the web-only interface will drop support for non-external accounts, much like the IPFS version.' At the bottom right are 'Cancel' and 'Restore' buttons. A green arrow points to the warning box, and a green circle highlights the 'Restore' button.

3. Double-check warning messages.

4. Click on **Restore** to complete the procedure.

GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

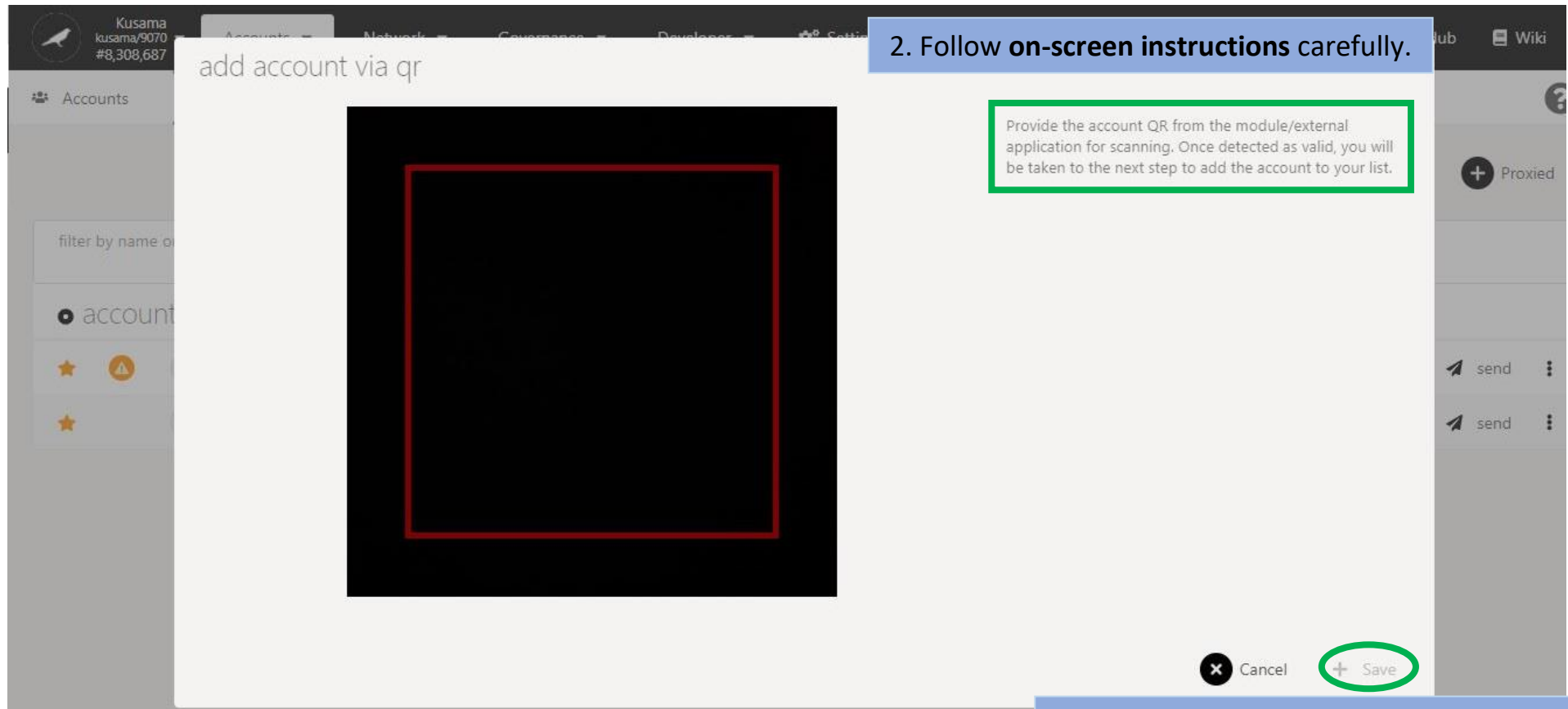
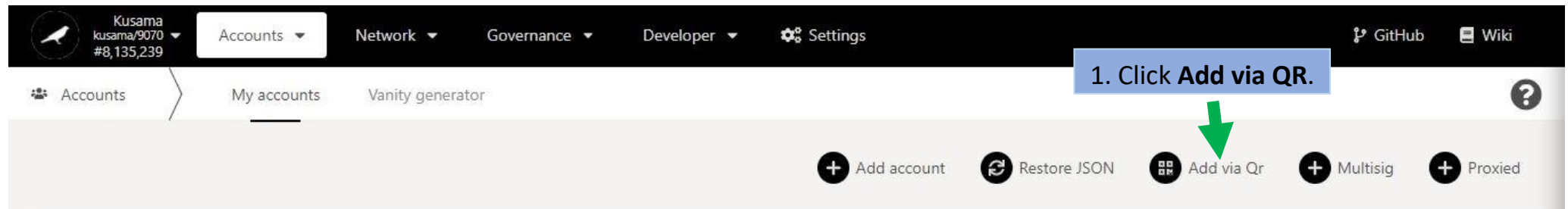
Version 1.2

The screenshot shows the Polkadot-JS Accounts interface. At the top, there's a navigation bar with 'Kusama kusama/9070 #8,212,268', 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. Below this, there's a sub-navigation bar with 'Accounts', 'My accounts', and 'Vanity generator'. The 'My accounts' section is active, showing a list of accounts. A green box highlights the 'LALA' account, which has a balance of 0.0000 KSM. A green notification banner at the top right says 'restore LALA account restored'.

	accounts	type	tags	balances	
★	ALTD@KSM (EXTENSION)	injected	no tags	10.3768 KSM	send
★	ANAELLE LTD@KSM	injected	no tags	1.0376 KSM	send
★	LALA	sr25519	no tags	0.0000 KSM	send
				11.4144 KSM	

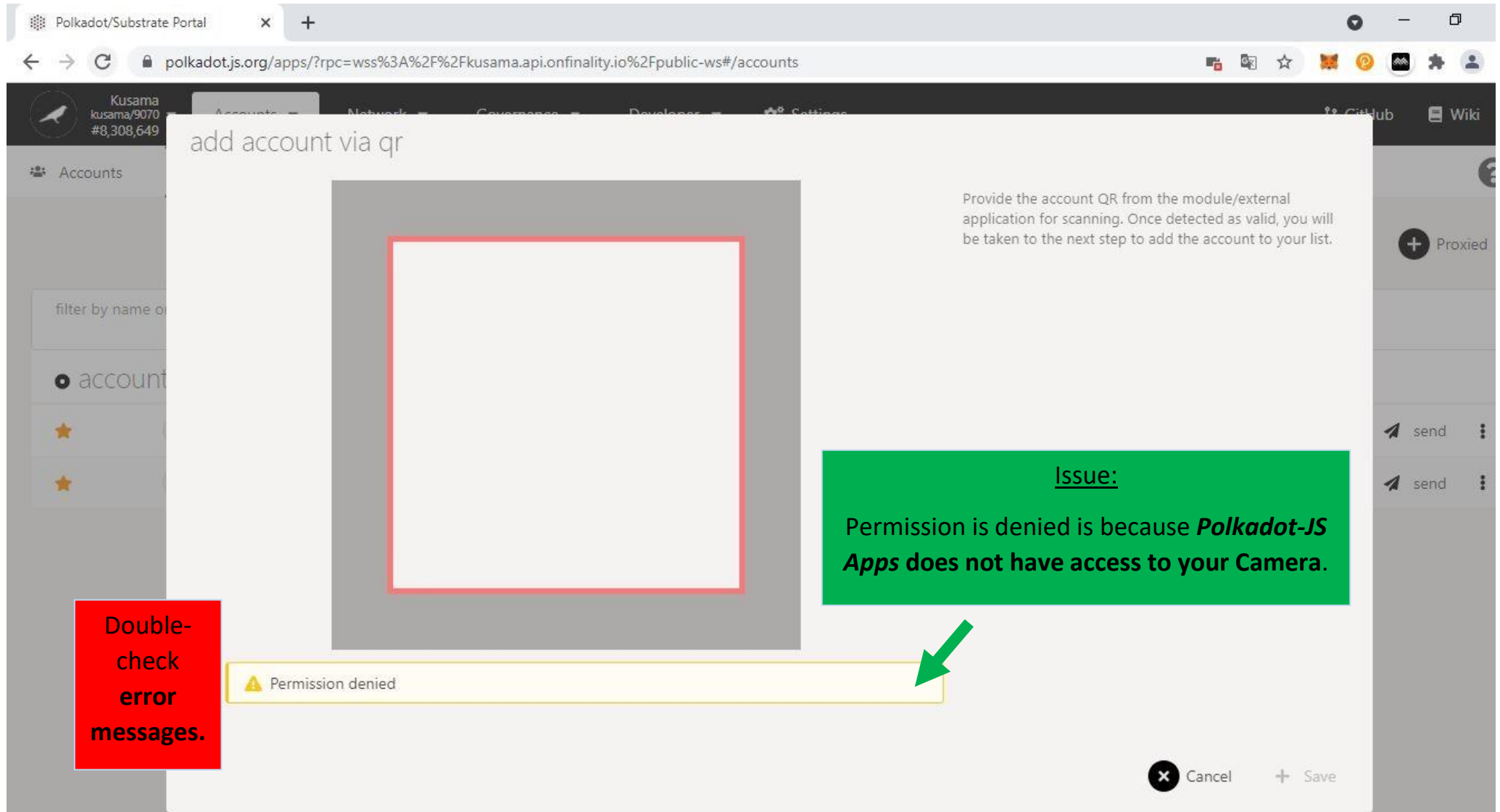
5. Your account has been restored in the *My accounts* list!

c) Add a Polkadot-JS account from a QR Code.



3. Click on **Save** to complete the procedure.

[Troubleshooting 1/3]



[Troubleshooting 2/3]

The screenshot shows the Polkadot/Substrate Portal interface. The browser address bar displays the URL: `polkadot.js.org/apps/?rpc=wss%3A%2F%2Fkusama.api.onfinality.io%2Fpublic-w`. The top navigation bar includes links for Accounts, Network, Governance, and Developer. The 'Accounts' section is active, showing a list of accounts. The 'My accounts' link is circled in green. A table of accounts is visible, with columns for 'accounts', 'type', and 'tags'. The table contains two entries: 'ALTD@KSM (EXTENSION)' and 'ANAELLE LTD@KSM', both marked as 'injected' with 'no tags'. Overlaid on the screenshot are four blue callout boxes with white text, numbered 1 through 4, providing troubleshooting steps. A green box with white text provides a 'Possible solution' and a numbered step. A settings menu is also shown, with a green arrow pointing to the 'Allow QR Camera Access' checkbox, which is checked. The settings menu includes options for Theme (Light/Dark), Manage Website Access, External Accounts and Access, Language (English), and an option to Open extension in new window.

1. Click the **Polkadot-JS extension icon**.

2. Click the Polkadot-JS extension **settings icon**.

Possible solution:

1. Make sure that your Polkadot-JS extension settings allow QR Camera Access.

3. Tick the **QR Camera Access box** and return to Polkadot-JS Apps.

4. Click on **My accounts** to refresh your browser, then restart the procedure.

[Troubleshooting 3/3]

1. Click the **Chrome** browser Camera icon.

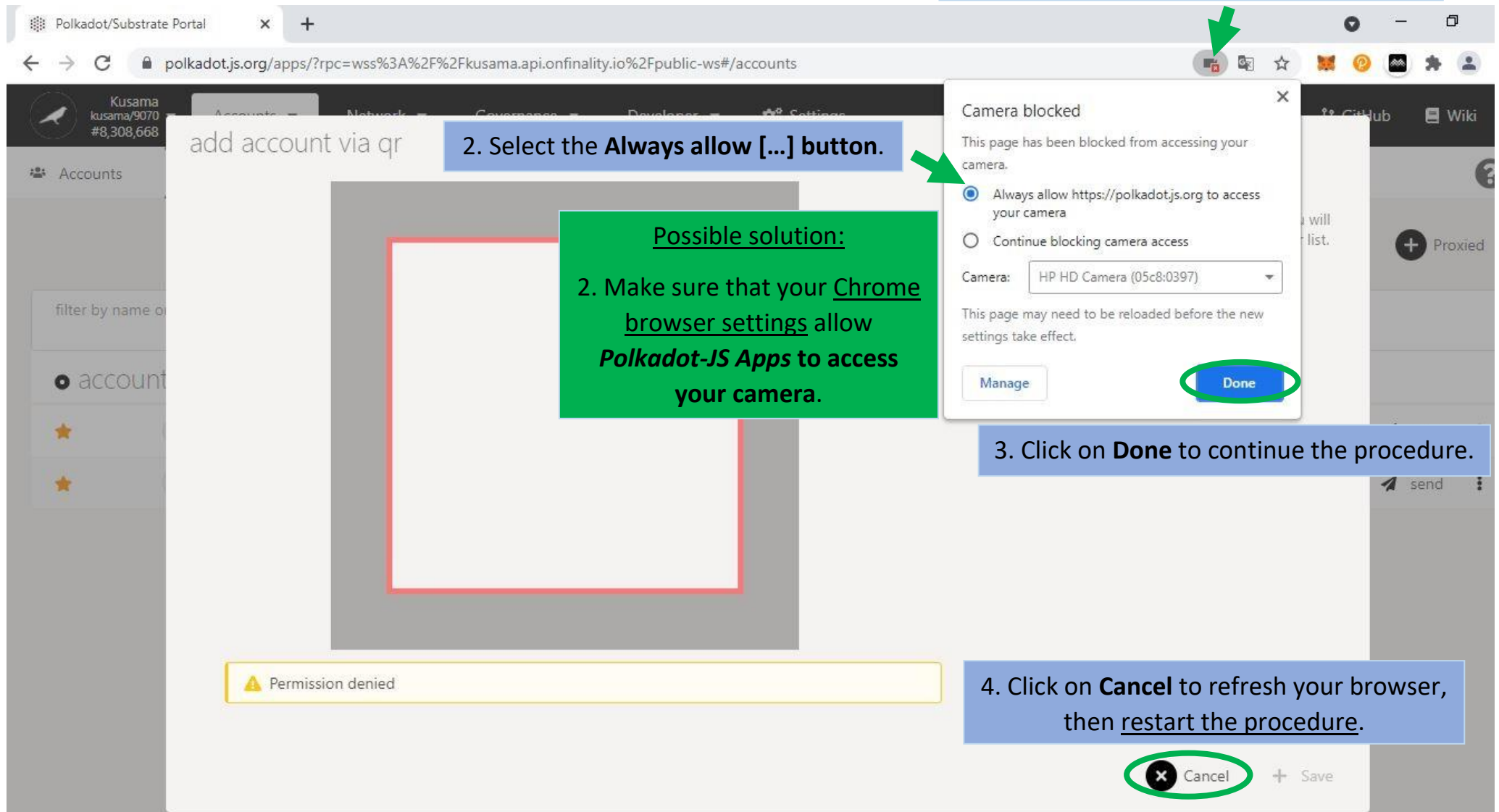
2. Select the **Always allow [...]** button.

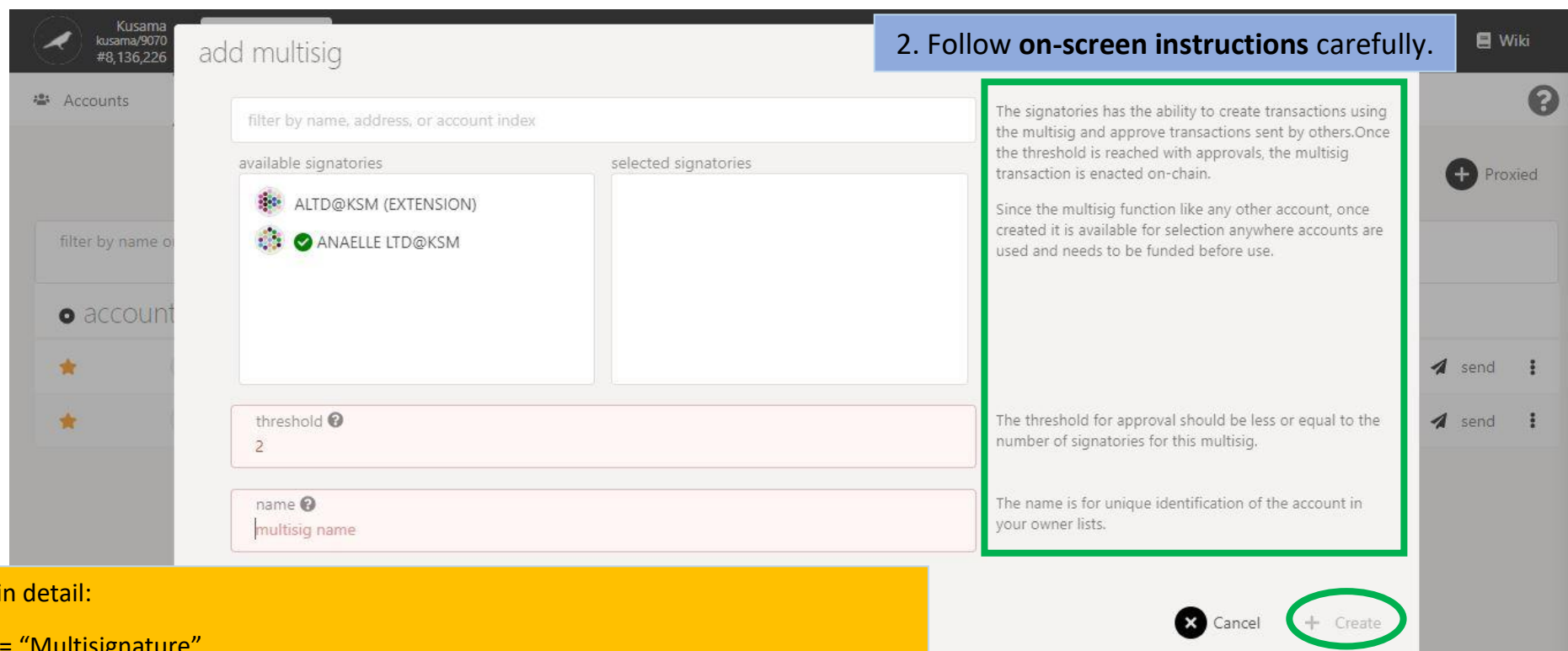
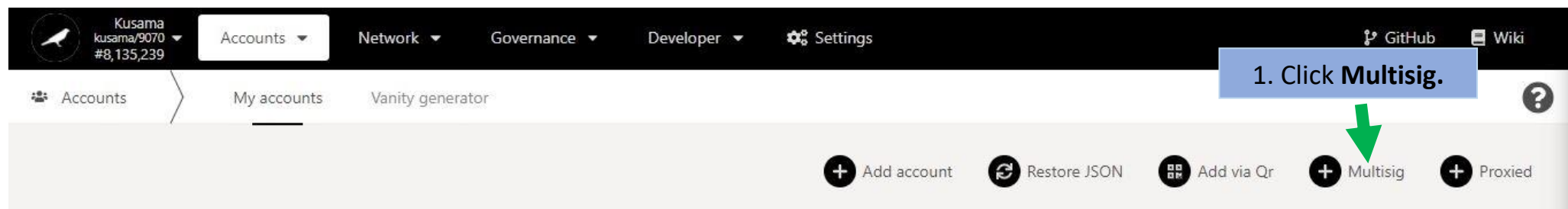
Possible solution:

2. Make sure that your Chrome browser settings allow **Polkadot-JS Apps** to access your camera.

3. Click on **Done** to continue the procedure.

4. Click on **Cancel** to refresh your browser, then restart the procedure.



d) Set up a Polkadot-JS Multisig account.

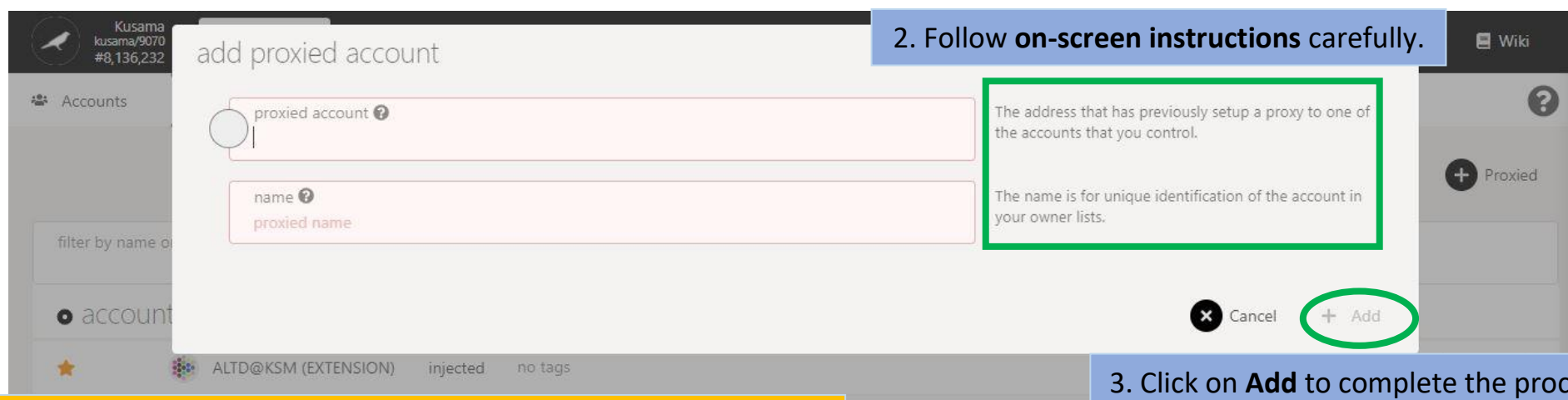
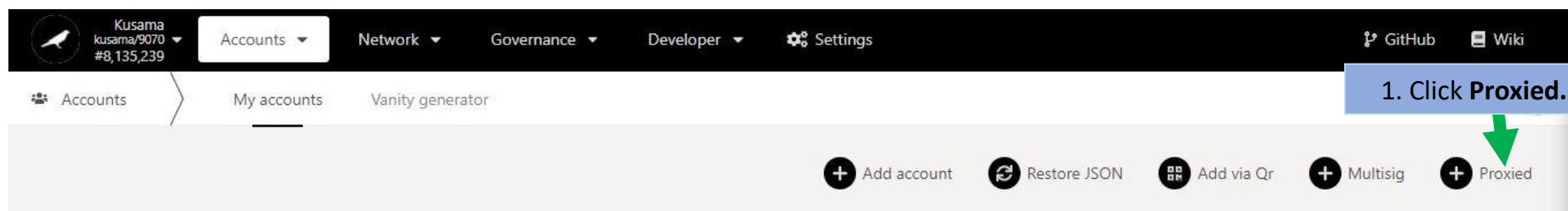
Multisig in detail:

Multisig = “Multisignature”.

Multisig accounts allow collectives, associations, organisations, and institutions to **manage transactions for a single account** without relying on one person.

Any account that is added as a signatory can submit/sign transactions; but the transactions will not be submitted on-chain until the **threshold** of signatories is reached.

3. Click on **Create** to complete the procedure.

e) Set up a Polkadot-JS Proxied account.

Proxy & Proxied accounts in detail:

Proxy = "Caretaker".

Proxied = "Cared for".

Proxy & Proxied accounts allow individuals to **manage transactions for a family member or a peer** if the person becomes unavailable or incapacitated.

The family member/peer must first **set up your account as their proxy account** before you can **add their account as your proxied account**.

To learn how to set up a proxy account for your account, refer to **Part VI: Accounts (Advanced setups)**.

f) Check your account information, receive funds at your KSM address and view your KSM balances.



The screenshot shows the 'Accounts' section of the Polkadot.js interface. At the top, there's a navigation bar with 'Kusama kusama/9070 #8,135,239', 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. Below this, there's a sub-navigation bar with 'Accounts', 'My accounts', and 'Vanity generator'. The main area displays a list of accounts. Three blue callout boxes with green arrows provide instructions: 1. Click on the account name to view its information. (points to 'ALTD@KSM (EXTENSION)'), 2. Click on the identicon to copy your KSM address. (points to the identicon of 'ANAELLE LTD@KSM'), and 3. Click on the dropdown arrow to view your detailed KSM balances. (points to the dropdown arrow next to '10.3509 KSM').

filter by name or tags

1. Click on the **account name** to view its information.

3. Click on the **dropdown arrow** to view your detailed KSM balances.

2. Click on the **identicon** to copy your KSM address.

accounts	type	tags	balances
★  ALTD@KSM (EXTENSION)	injected	no tags	10.3509 KSM ▼ send ⋮
★  ✓ ANAELLE LTD@KSM	injected	no tags	1.0376 KSM ▼ send ⋮
			11.3886 KSM

GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2

Kusama
kusama/9070
#8,136,085

Accounts

Network

Governance

Developer

Settings

GitHub

Wiki

AccountsMy accountsVanity generator

+ Add account

Restore JSON

Add via Qr

+ Multisig

+ Proxied


filter by name or tags

accounts

type

tags



★

 ALTD@KSM (EXTENSION)

injected

no tags

★

  ANAELLE LTD@KSM

injected

no tags

KSM balance in detail:

Transferrable = KSM existential deposit (0.000034 KSM) + available KSM.

Locked = KSM not currently transferrable because it is bonded or vested.

Bonded = KSM locked for staking nominations. Must be unbonded over 7 days to become transferrable.

Democracy = KSM locked for voting. Must be unlocked at the end of the voting period to become transferrable. Note: If a voluntary locking period of 8 days has been set during the last vote, the voting balance will be unlocked after 8 days.

Reserved = KSM locked for submitting council candidatures, treasury proposals, tips, on-chain identities, etc. Must be cleared to become transferrable.

balances

10.3509 KSM ▲

transferrable 0.3427 KSM

locked ⓘ 10.0082 KSM

bonded ⓘ 10.0082 KSM

democracy ⓘ 9.8733 KSM

1.0376 KSM ⓘ

transferrable 1.0043 KSM

reserved 0.0333 KSM

11.3886 KSM

send

send

g) Generate a vanity address.

The screenshot shows the Polkadot-JS Accounts interface. At the top, there is a navigation bar with links for Accounts, Network, Governance, Developer, and Settings. Below this, the 'Accounts' section is active, showing 'My accounts' and 'Vanity generator'. The 'Vanity generator' section contains three input fields: 'Search for' (with 'Kusama' entered), 'keypair crypto type' (with 'Schnorrkel (sr25519, recommended)' selected), and 'case sensitive' (with 'Yes' selected). A green box highlights these three fields. A blue callout box with the text '1. Click Vanity generator.' points to the 'Vanity generator' tab. Another blue callout box with the text '2. Fill-in or select the information required.' points to the input fields. A third blue callout box with the text '3. Click on Start generation to complete the procedure.' points to a 'Start generation' button, which is circled in green.

Kusama
kusama/9070
#8,135,263

Accounts ▾ Network ▾ Governance ▾ Developer ▾ Settings

Accounts ▾ My accounts ▾ Vanity generator

1. Click Vanity generator.

Search for ?
Kusama

keypair crypto type ?
Schnorrkel (sr25519, recommended)

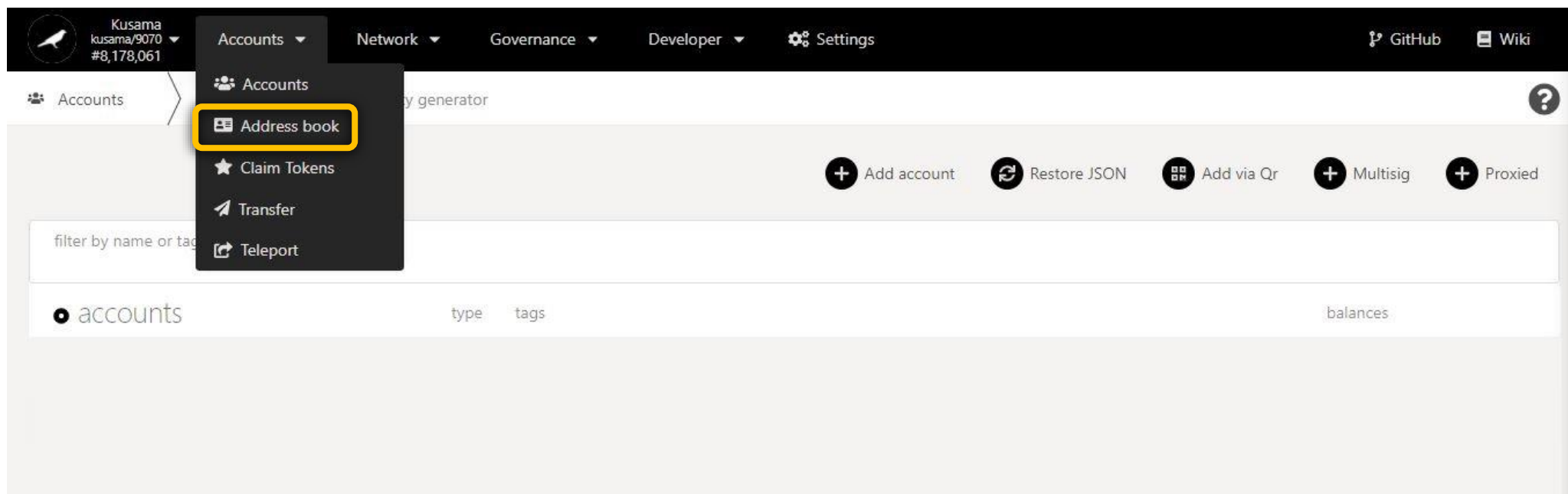
case sensitive ?
Yes

2. Fill-in or select the information required.

Start generation

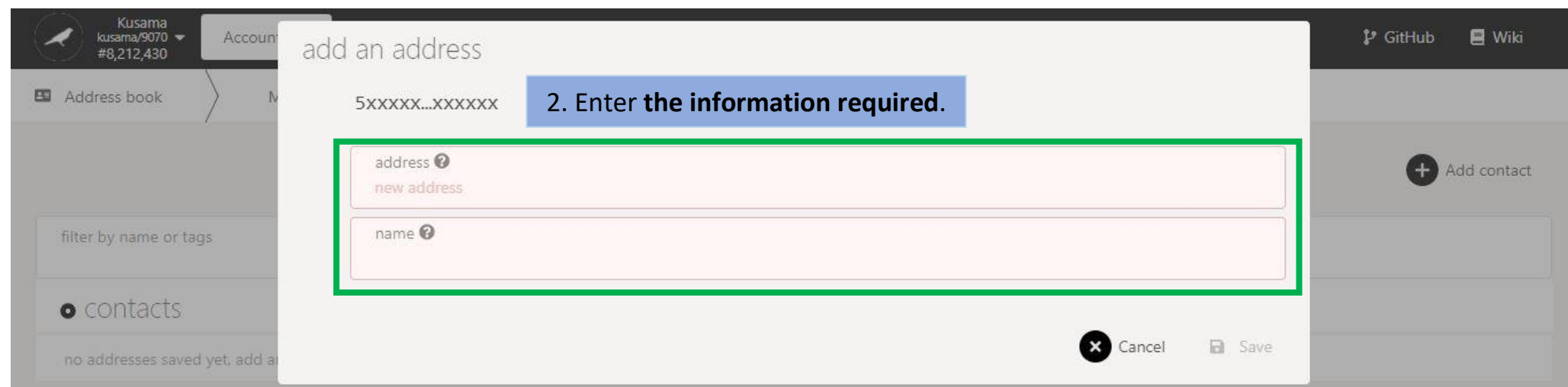
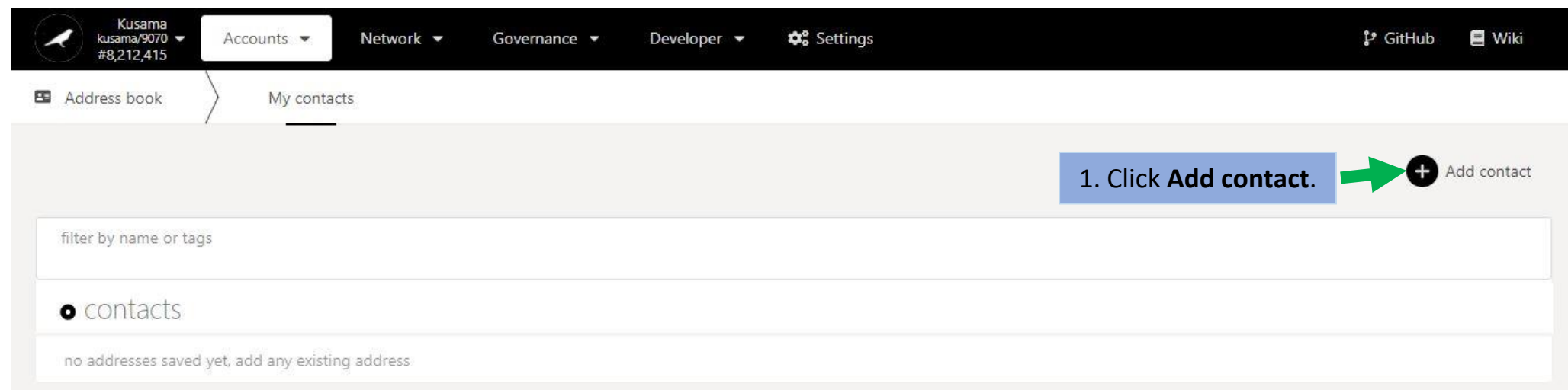
3. Click on Start generation to complete the procedure.

2. Address book: Bookmark your favourite KSM addresses as contacts.



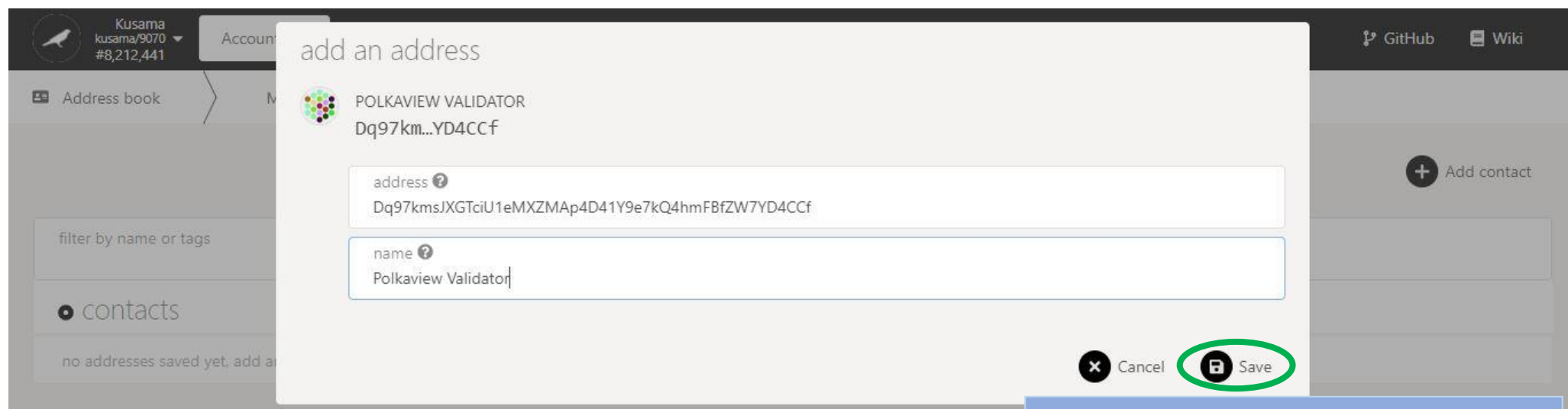
GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2

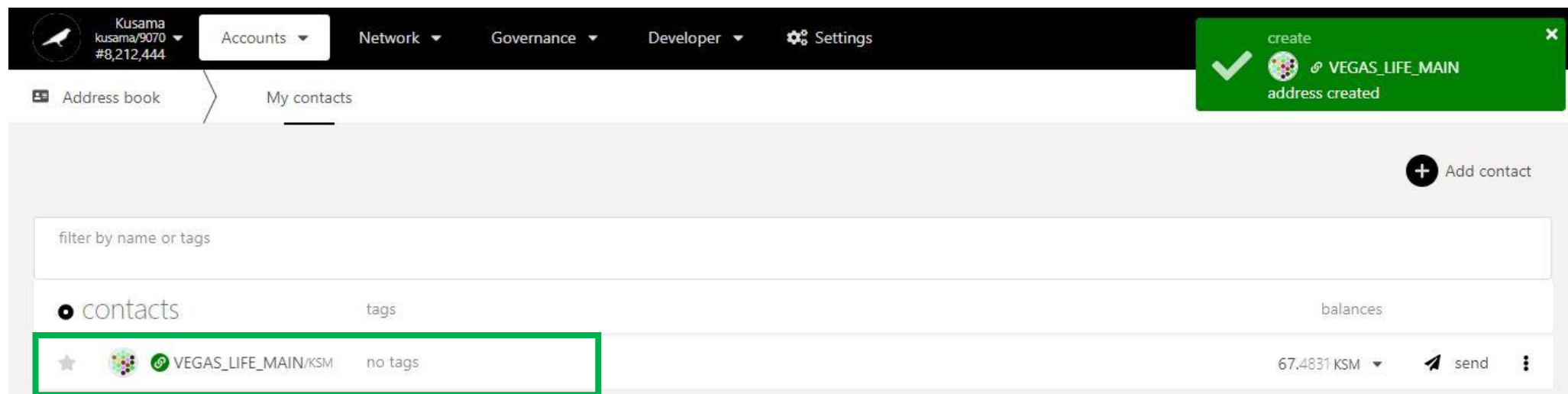


GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2



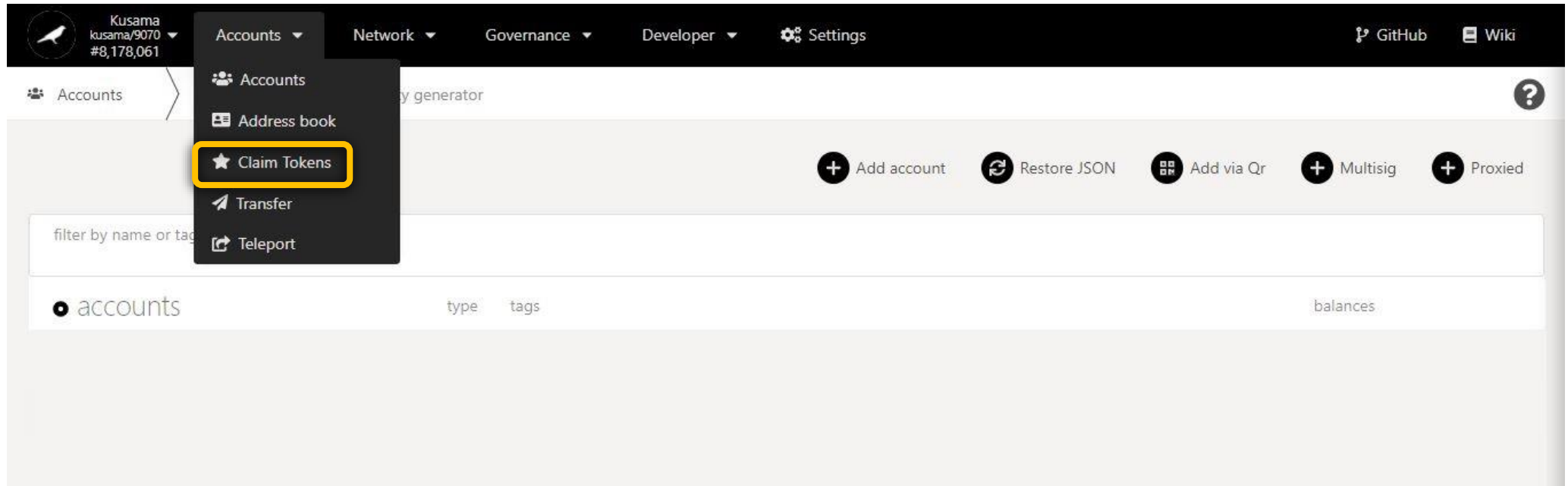
3. Click on **Save** to complete the procedure.



4. Your contact has been added in the *My contacts* list!


Note: If your contact has an identity, it will automatically replace the name you provided.

3. Claim tokens: Claim your KSM from your ETH address.



GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.2




Kusama
kusama/9070
#8,135,295


Accounts ▾


Network ▾

Governance ▾

Developer ▾

 Settings

 GitHub


 Wiki

★ Claim Tokens

Claim tokens

claim your KSM tokens

1. Select your Kusama account.

 claim to account ?

✓ ANAELLE LTD@KSM

HjcErRijmpoiBiKEHT3edPXM3NFycJogVwDP...

2. Enter the ETH address from the sale.

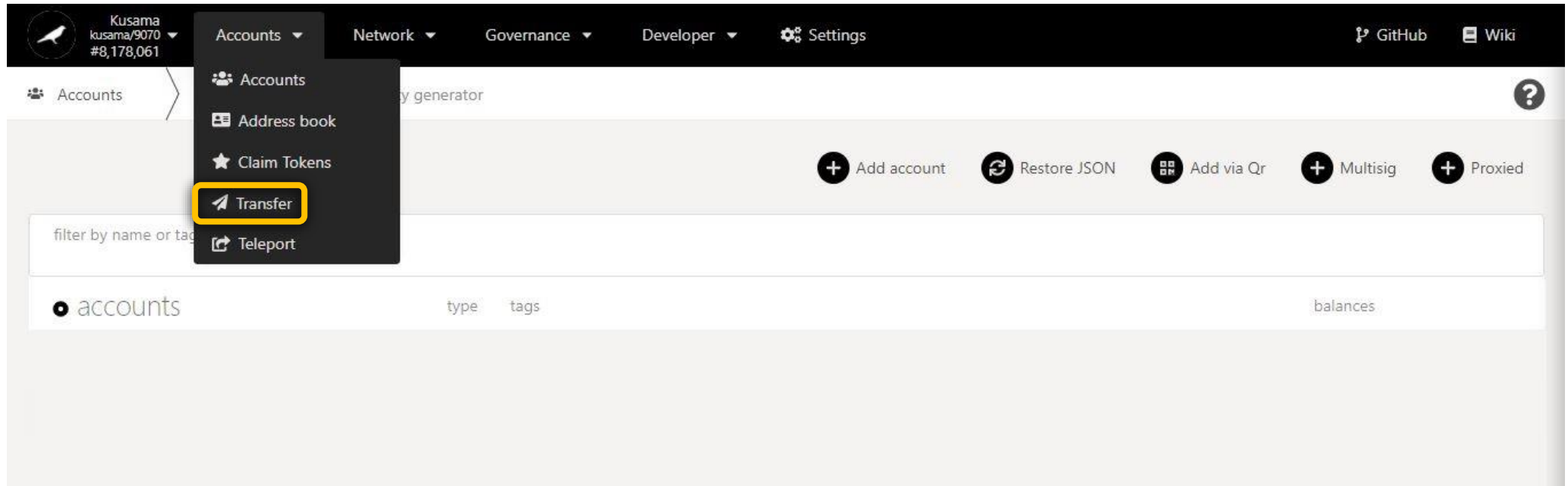
Pre-sale ethereum address ?

➔ Continue

1. Enter the information required.

2. Click on **Continue** to view and complete subsequent steps in the procedure.

4. Transfer: Send KSM to another KSM address.



The receiving account must be an account from **your My Accounts or My contacts list**.

The screenshot shows the 'send funds' modal in Polkadot-JS. It includes a 'send from account' dropdown with 'ANAELLE LTD@KSM' selected, showing a balance of 1.0043 KSM. Below it is a 'send to address' dropdown with 'ALTD@KSM (EXTENSION)' selected, showing a balance of 0.3427 KSM. There are input fields for 'amount' (set to 0) and 'existential deposit' (set to 33.3333). A toggle switch for 'Transfer with account keep-alive checks' is currently turned on. At the bottom are 'Cancel' and 'Make Transfer' buttons. A green arrow points from the 'ANAELLE LTD@KSM' account in the 'send from account' dropdown to the yellow callout box. Another green arrow points from the 'Transfer with account keep-alive checks' toggle switch to the yellow callout box. A green box highlights the explanatory text on the right side of the modal.

1. Follow on-screen instructions carefully.

The transferred balance will be subtracted (along with fees) from the sender account.

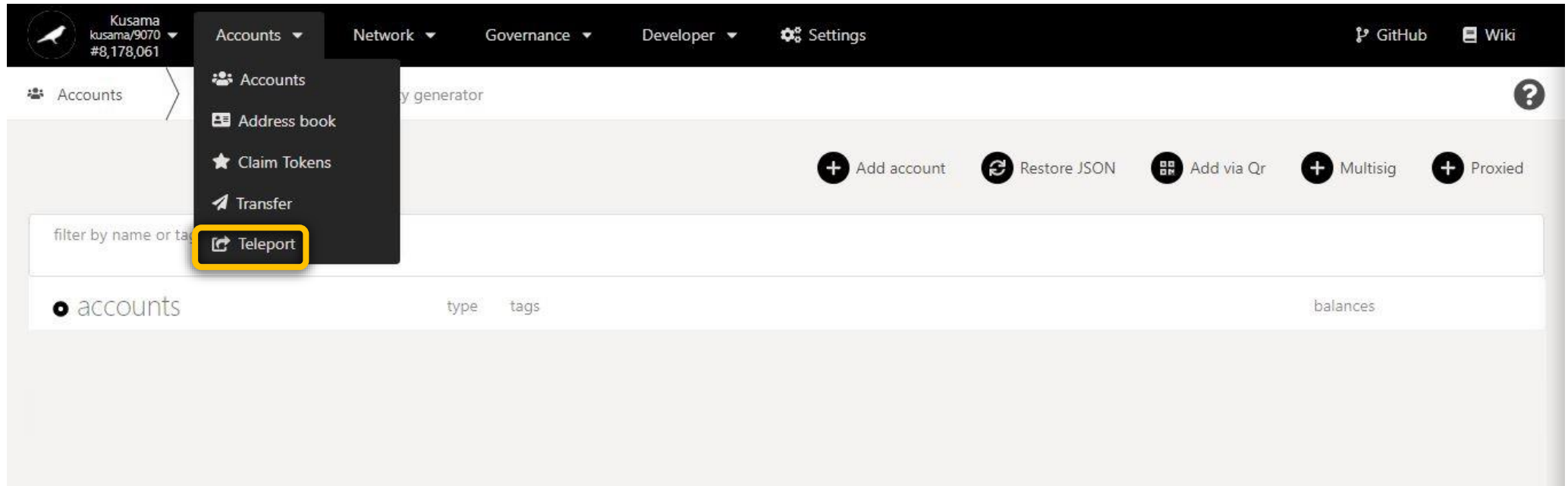
The beneficiary will have access to the transferred fees when the transaction is included in a block.

If the recipient account is new, the balance needs to be more than the existential deposit. Likewise if the sending account balance drops below the same value, the account will be removed from the state.

If you want to **empty the account completely**, switch the keep-alive checks **OFF**. The existential deposit will be transferred, and the account will be deactivated.

2. Click on **Make Transfer** to complete the procedure.

5. Teleport: Send your KSM to a parachain address.



Always double-check the **parachain** information.

teleport assets

send from account: transferrable 1.0043 KSM
ANAELLE LTD@KSM

destination chain: Statemine

send to address: ALTD@KSM (EXTENSION)

amount: 0 KSM

destination transfer fee: 7.9999 micro

destination existential deposit: 3.3333 micro

1. Follow on-screen instructions carefully.

The transferred balance will be subtracted (along with fees) from the sender account.

The destination chain for this asset teleport. The transferred value will appear on this chain.

The beneficiary will have access to the transferred amount when the transaction is included in a block.

If the recipient account is new, the balance needs to be more than the existential deposit on the recipient chain.

The amount deposited to the recipient will be net the calculated cross-chain fee.

Cancel Teleport

2. Click on **Teleport** to complete the procedure.

Always double-check the **existential deposit** information.