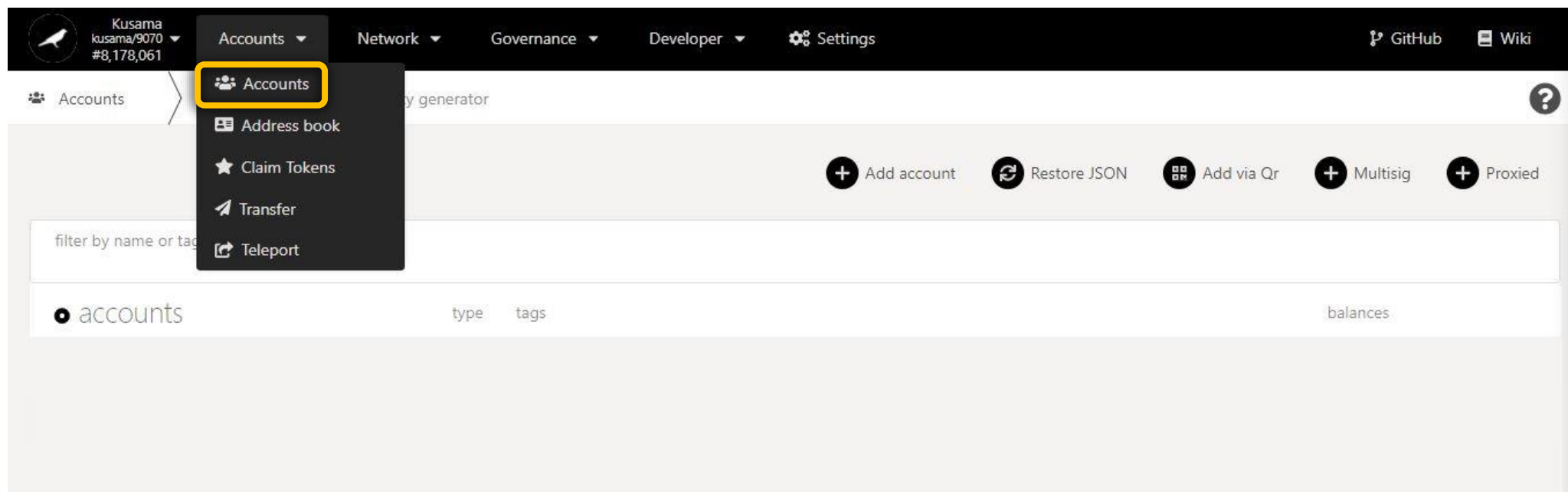


[PART I: Accounts \(Basic\)](#)

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.

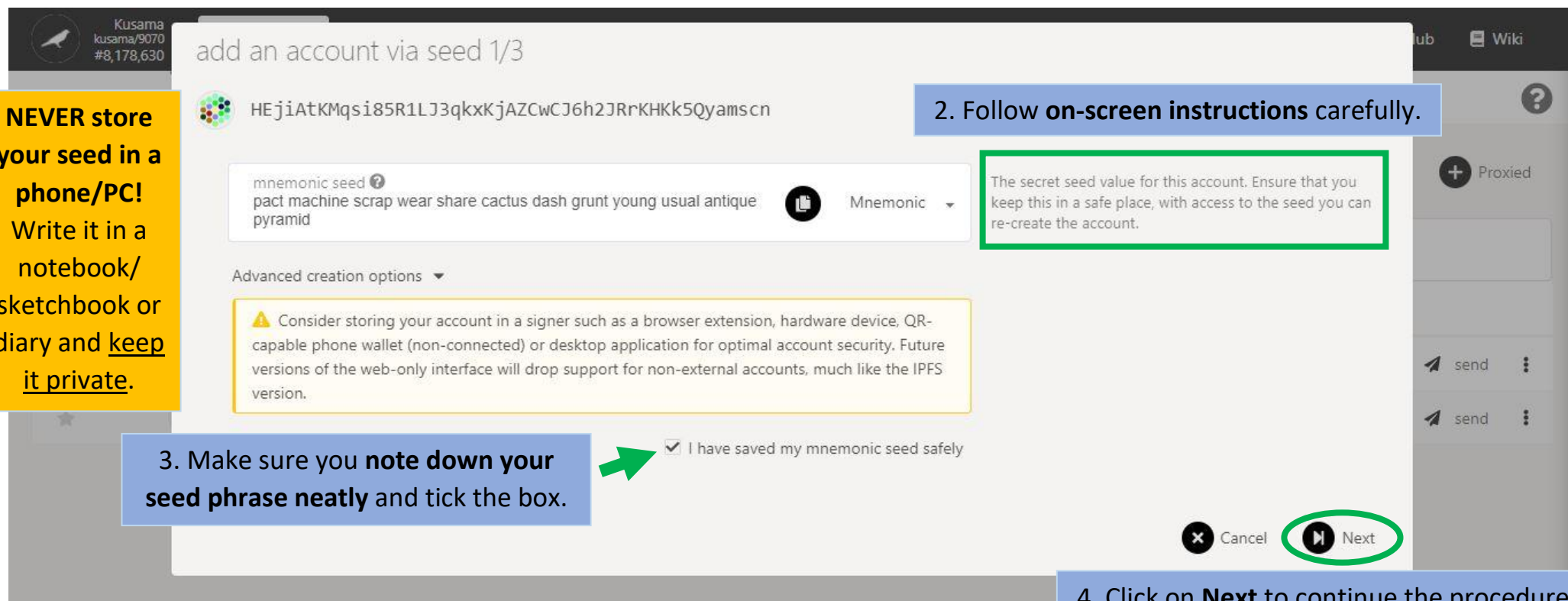
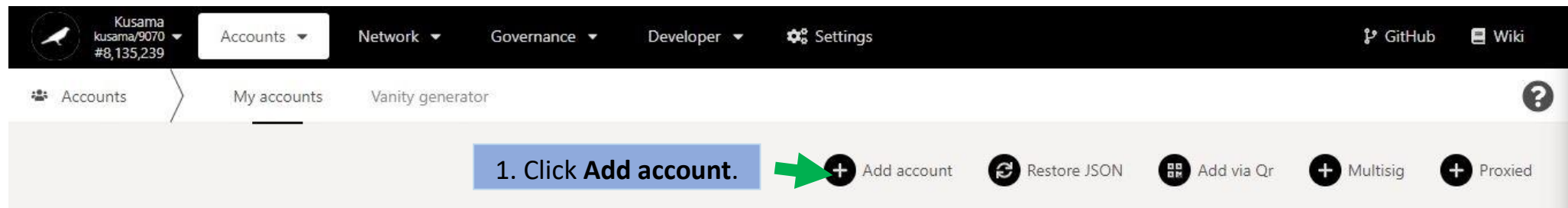
Existing account(s) enabled on the Kusama network.

Overview of account's KSM balances.

The screenshot shows the Polkadot-JS Accounts interface. At the top, there's a navigation bar with '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 list of accounts. Two accounts are visible: 'ALTD@KSM (EXTENSION)' and 'ANAELLE LTD@KSM', both marked as 'injected' with 'no tags'. To the right of the account list, there's a 'balances' section showing three KSM balances: 10.3509 KSM, 1.0376 KSM, and 11.3886 KSM. A green box highlights the account names and the balances section. A green arrow points from the 'type' column header to the 'injected' status of the first account. Another green arrow points from the 'tags' column header to the 'no tags' status of the first account. A green arrow points from the top right corner to the 'Wiki' link. A green arrow points from the top right corner to the 'Add account' button. A green arrow points from the top right corner to the 'Add via Qr' button. A green arrow points from the top right corner to the 'Multisig' button. A green arrow points from the top right corner to the 'Proxied' button. A green arrow points from the top right corner to the 'send' button. A green arrow points from the top right corner to the 'send' button. A green arrow points from the top right corner to the 'send' button.

filter by name or tags	accounts	type	tags	balances
	ALTD@KSM (EXTENSION)	injected	no tags	10.3509 KSM
	ANAELLE LTD@KSM	injected	no tags	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 1.1

add an account via seed 2/3

Account Information:

- Name: 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)
- 5. Follow the new instructions carefully.** (Blue box)
- 6. Click on Next to continue the procedure.** (Blue box)

Instructions:

- 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.

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

Version 1.1

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 1.1

The screenshot shows the 'Accounts' tab in the Polkadot-JS application. 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.' has a green arrow pointing to the star icon next to the 'LALA' account. The 'LALA' account is highlighted with a green border. A blue box with the text '9. Your account has been added to the *My accounts* list!' points 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.' has a green arrow pointing to the JSON file icon and name 'HEjiAtKMqsi85R1....json' at the bottom. A blue box with the text '10. Click on the 3 vertical dots to view Account settings.' has a green arrow pointing to the three vertical dots icon next to the 'LALA' account. The table of accounts has columns for 'accounts', 'type', 'tags', and 'balances'. The 'LALA' account is listed with type 'sr25519' and no tags. The balance for 'LALA' is 0.0000 KSM. The total balance for all accounts is 11.4009 KSM.

Star your most important accounts.

filter by name or tags

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

9. Your account has been added to the *My accounts* list!

10. Click on the 3 vertical dots to view Account settings.

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.

HEjiAtKMqsi85R1....json

Show all X

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

Version 1.1

The screenshot shows the Polkadot.js web interface. At the top, the header includes the Kusama logo, the address 'kusama/9070 #8,178,673', and navigation tabs for 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. Below the header, the 'Accounts' section is active, showing a list of accounts. A blue callout box with the text '11. Click on **Create a backup file [...]** to download your account's JSON file.' points to the 'Create a backup file for this account' option in the context menu that appears when the 'ALTD@KSM (EXTENSION)' account is selected. The context menu also includes options like 'Set on-chain identity', 'Derive account via derivation path', 'Change this account's password', 'Forget this account', 'Make recoverable', 'Initiate recovery for another', 'Delegate democracy votes', 'Add proxy', and a toggle for 'only this network'.

Kusama
kusama/9070
#8,178,673

Accounts Network Governance Developer Settings

Accounts My accounts Vanity generator

+ Add account Restore JSON Add via QR

filter by name or tags

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

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

- 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 1.1

The screenshot shows the Polkadot-JS Accounts interface. A modal dialog titled "backup account" is open. The dialog contains the following text:

12. Follow on-screen instructions carefully.

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.

Below the text is a password input field labeled "password" with a question mark icon. The field contains a series of dots.

At the bottom of the dialog are two buttons: "Cancel" and "Download". The "Download" button is circled in green.

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

The background interface shows a list of accounts. The first account is "LALA" with the address "HEjiAt...yamsn". The second account is "ANAELLE LTD@KSM" with the address "sr25519". The third account is "LALA" with the address "sr25519".

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 shows 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 overlaid on the 'Restore JSON' button. Below this, a dialog box titled 'add via backup file' is open. It contains a text input field for the account name, a file selection field for the backup file, and a password input field. A green box highlights the password field with the text '2. Follow on-screen instructions carefully.' and a green arrow pointing to it. A yellow box contains a warning message: '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.' The dialog has 'Cancel' and 'Restore' buttons at the bottom right.

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 1.1

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 the LALA logo 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 1.1

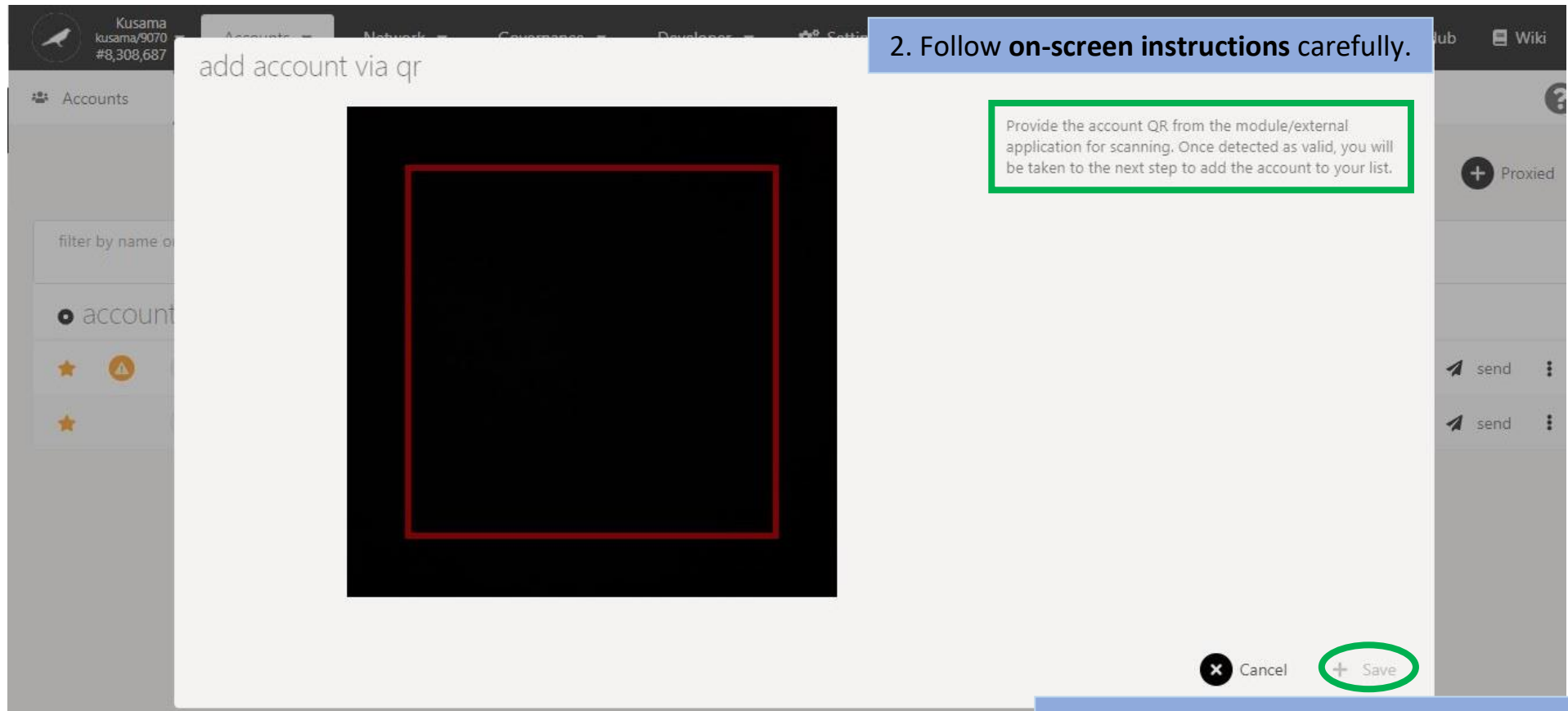
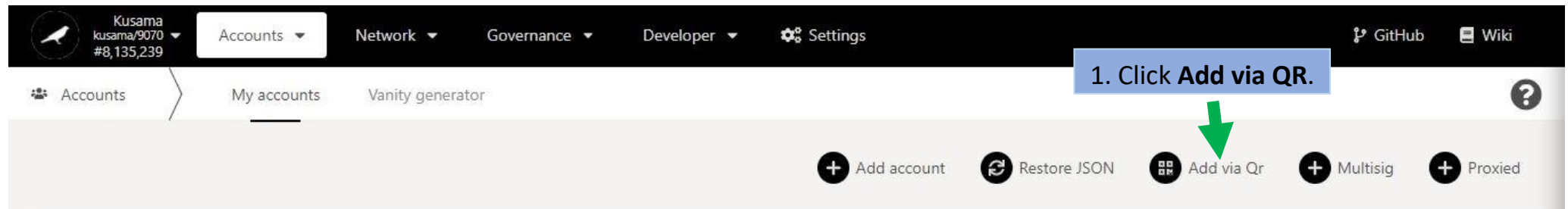
The screenshot shows the 'Accounts' section of the Polkadot-JS interface. The top navigation bar includes 'Kusama kusama/9070 #8,212,268', 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. A green notification banner at the top right states 'restore LALA account restored'. Below the navigation bar, there are tabs for 'Accounts', 'My accounts', and 'Vanity generator'. The 'My accounts' tab is active, showing a list of accounts. The list has columns for 'accounts', 'type', 'tags', and 'balances'. Three accounts are listed: 'ALTD@KSM (EXTENSION)' with a balance of 10.3768 KSM, 'ANAELLE LTD@KSM' with a balance of 1.0376 KSM, and 'LALA' with a balance of 0.0000 KSM. The 'LALA' account is highlighted with a green box. At the bottom right of the list, the total balance is shown as 11.4144 KSM.

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

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]

The screenshot shows the Polkadot/Substrate Portal interface. A modal window titled 'add account via qr' is open. Inside the modal, there is a large red square placeholder for a QR code. To the right of the placeholder, text reads: 'Provide the account QR from the module/external application for scanning. Once detected as valid, you will be taken to the next step to add the account to your list.' Below the QR placeholder, a yellow error message box displays a warning icon and the text 'Permission denied'. A green arrow points from a green text box to this error message. The green text box contains the text: 'Issue: Permission is denied is because **Polkadot-JS Apps** does not have access to your Camera.' At the bottom right of the modal, there are 'Cancel' and 'Save' buttons. In the background, the portal's sidebar shows the 'Accounts' section with a filter by name and a list of accounts. The top navigation bar includes links for Accounts, Network, Governance, Developer, and Settings. The top right corner shows links for GitHub and Wiki.

add account via qr

Provide the account QR from the module/external application for scanning. Once detected as valid, you will be taken to the next step to add the account to your list.

Issue:

Permission is denied is because **Polkadot-JS Apps** does not have access to your Camera.

Permission denied

Cancel Save

Double-check error messages.

[Troubleshooting 2/3]

The screenshot shows the Polkadot/Substrate Portal interface. The browser's address bar displays the URL: `polkadot.js.org/apps/?rpc=wss%3A%2F%2Fkusama.api.onfinality.io%2Fpublic-w`. The page header includes navigation links: 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 two entries: 'ALTD@KSM (EXTENSION)' and 'ANAELLE LTD@KSM'. A green box with the text 'Possible solution: 1. Make sure that your Polkadot-JS extension settings allow QR Camera Access.' is overlaid on the page. A settings menu is open, showing options for Theme (Light/Dark), Manage Website Access, and External Accounts and Access. The 'Allow QR Camera Access' checkbox is checked. A blue box with the text '3. Tick the QR Camera Access box and return to Polkadot-JS Apps.' points to the checkbox. Other blue boxes provide instructions: '1. Click the Polkadot-JS extension icon.', '2. Click the Polkadot-JS extension settings icon.', and '4. Click on My accounts to refresh your browser, then restart the procedure.'

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**.

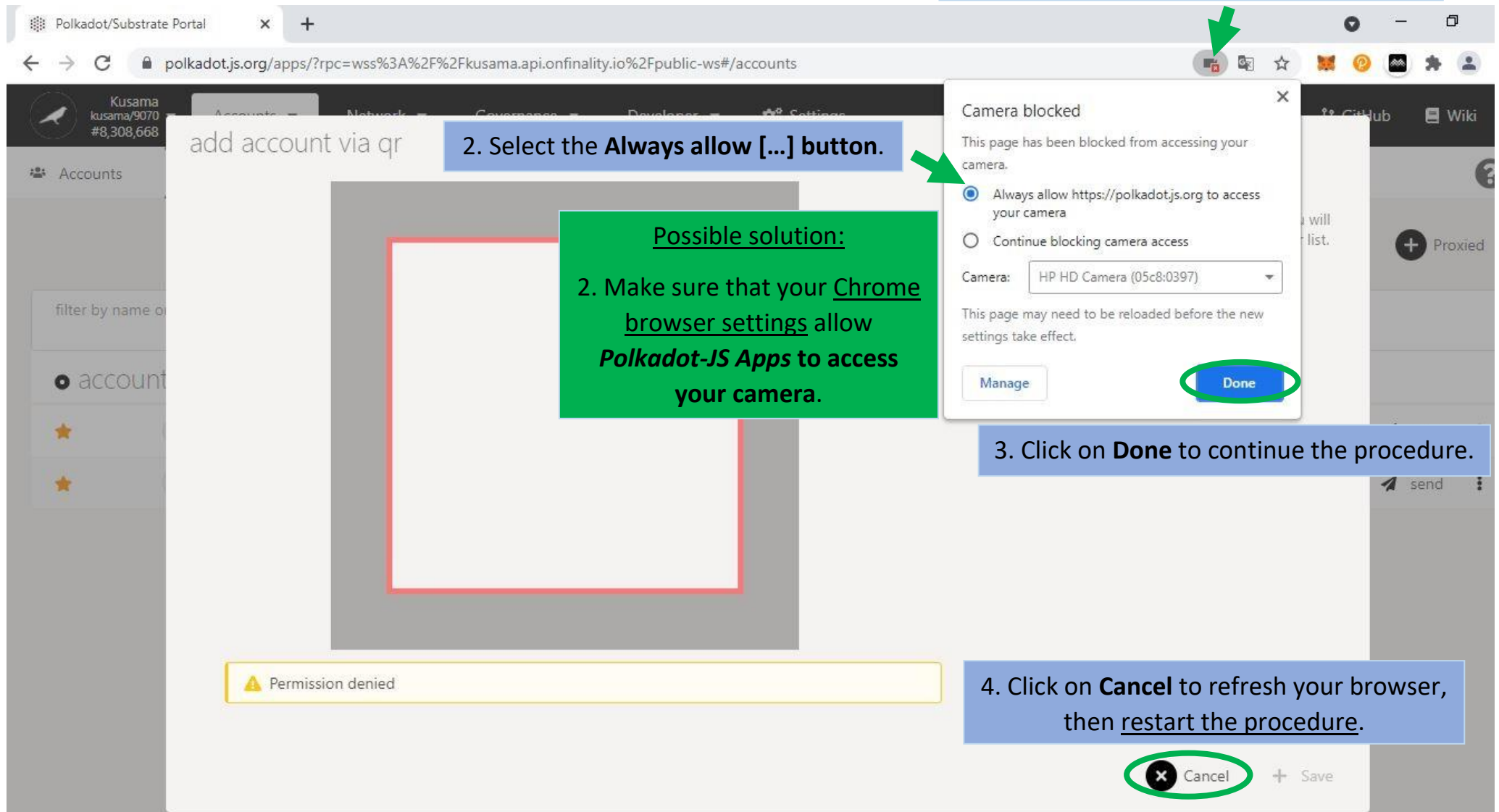
2. Select the **Always allow [...]** button.

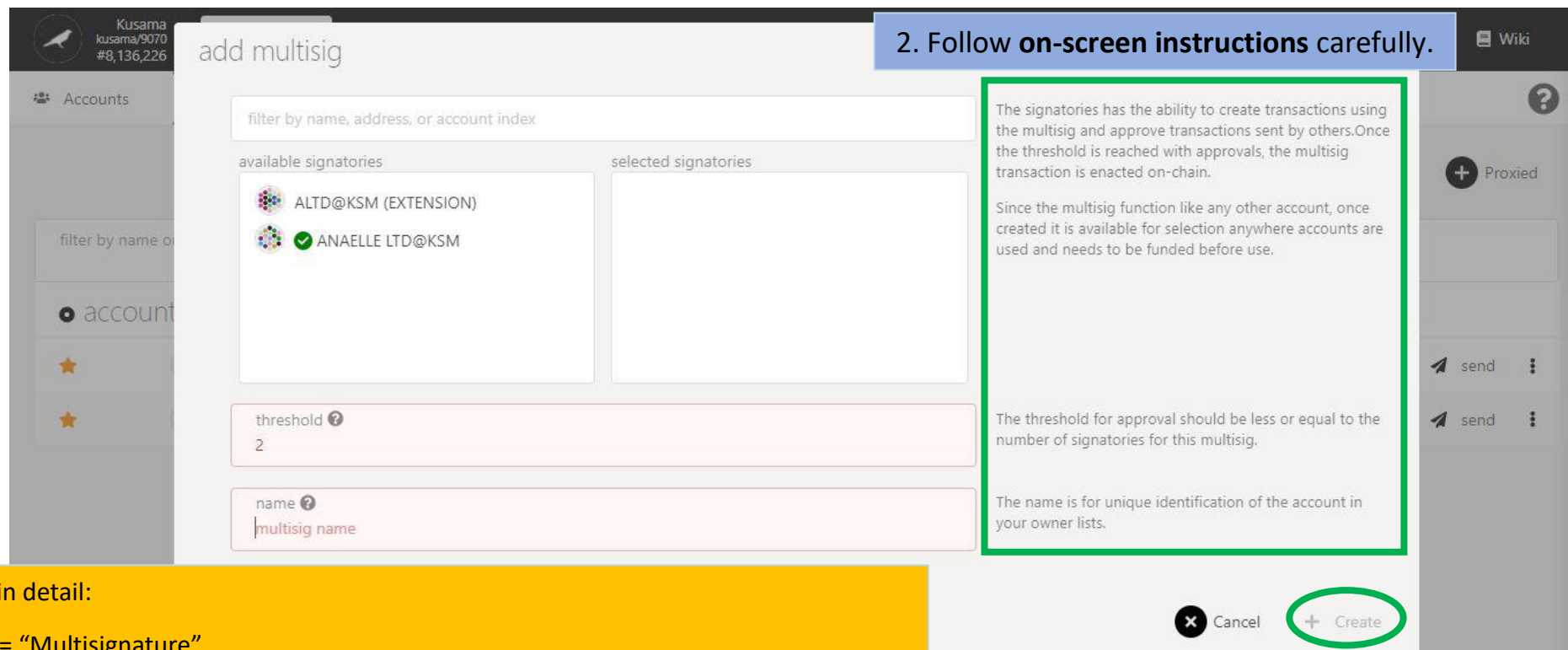
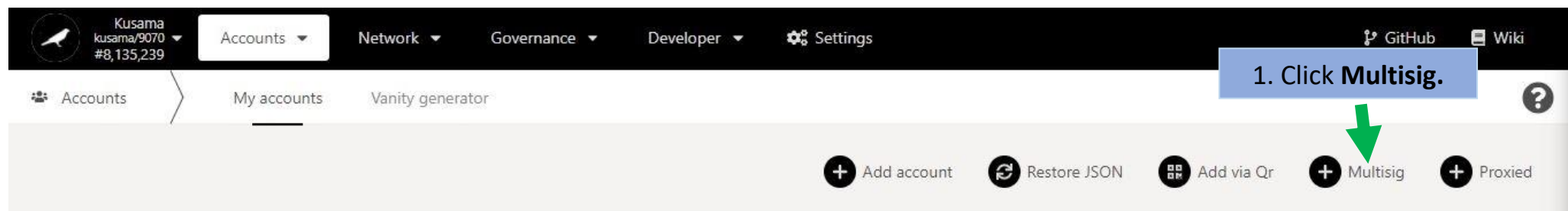
Possible solution:

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.

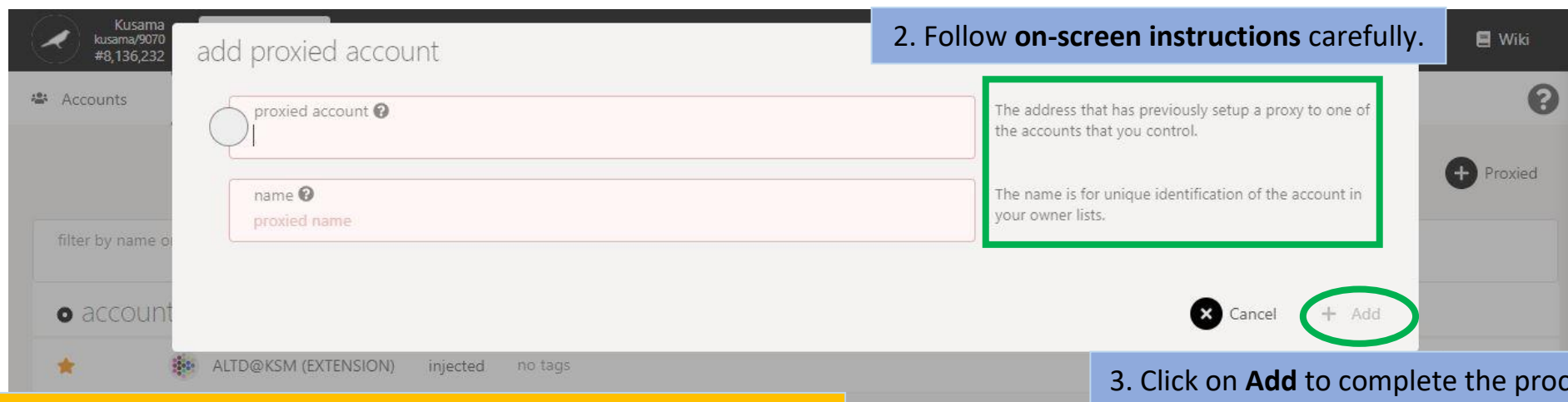
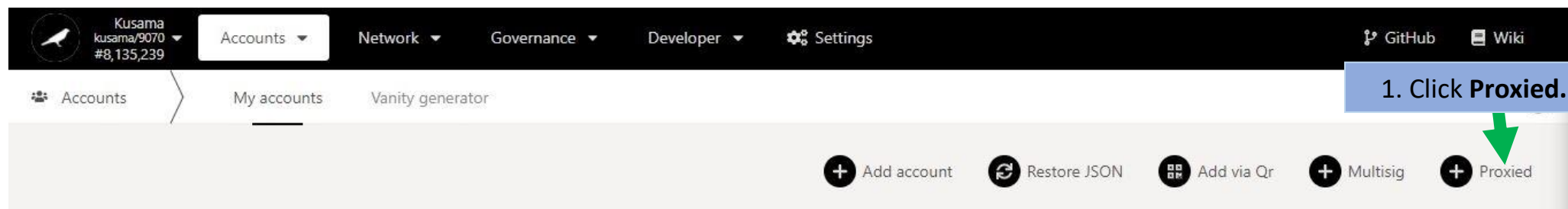
Multisig in detail:

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:

Proxied = “Cared for”.

Proxy = “Caretaker”.

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, the 'Accounts' tab is active, showing 'My accounts' and a 'Vanity generator'. A search bar 'filter by name or tags' is present. Above the account list are buttons: '+ Add account', 'Restore JSON', 'Add via Qr', '+ Multisig', and '+ Proxied'. The account list has columns for 'accounts', 'type', 'tags', and 'balances'. Two accounts are listed: 'ALTD@KSM (EXTENSION)' and 'ANAELLE LTD@KSM'. The first account's balance is '10.3509 KSM' and the second's is '1.0376 KSM'. A total balance of '11.3886 KSM' is shown at the bottom. Three numbered instructions with green arrows point to specific elements: 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'); 3. Click on the dropdown arrow to view your detailed KSM balances. (points to the dropdown arrow next to '10.3509 KSM').

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

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

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

accounts	type	tags	balances
ALTD@KSM (EXTENSION)	injected	no tags	10.3509 KSM
ANAELLE LTD@KSM	injected	no tags	1.0376 KSM
			11.3886 KSM

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

Version 1.1

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 'Accounts' section of the Polkadot.js interface. The top navigation bar includes 'Kusama', 'kusama/9070', '#8,135,263', and dropdown menus for 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. On the right are links for 'GitHub' and 'Wiki'. Below the navigation bar, the 'Accounts' tab is active, showing 'My accounts' and 'Vanity generator'. A green arrow points to the 'Vanity generator' tab with the instruction '1. Click Vanity generator.' The 'Vanity generator' form is highlighted with a green border and contains three input fields: 'Search for' (with 'Kusama' entered), 'keypair crypto type' (with 'Schnorrkel (sr25519, recommended)' selected), and 'case sensitive' (with 'Yes' selected). A blue box with the instruction '2. Fill-in or select the information required.' points to these fields. At the bottom right of the form, a 'Start generation' button is circled in green, with a blue box and the instruction '3. Click on Start generation to complete the procedure.' pointing to it.

Kusama
kusama/9070
#8,135,263

Accounts Network Governance Developer Settings

GitHub Wiki

Accounts My accounts Vanity generator

1. Click Vanity generator.

Search for ?
Kusama

keypair crypto type ?
Schnorrkel (sr25519, recommended)

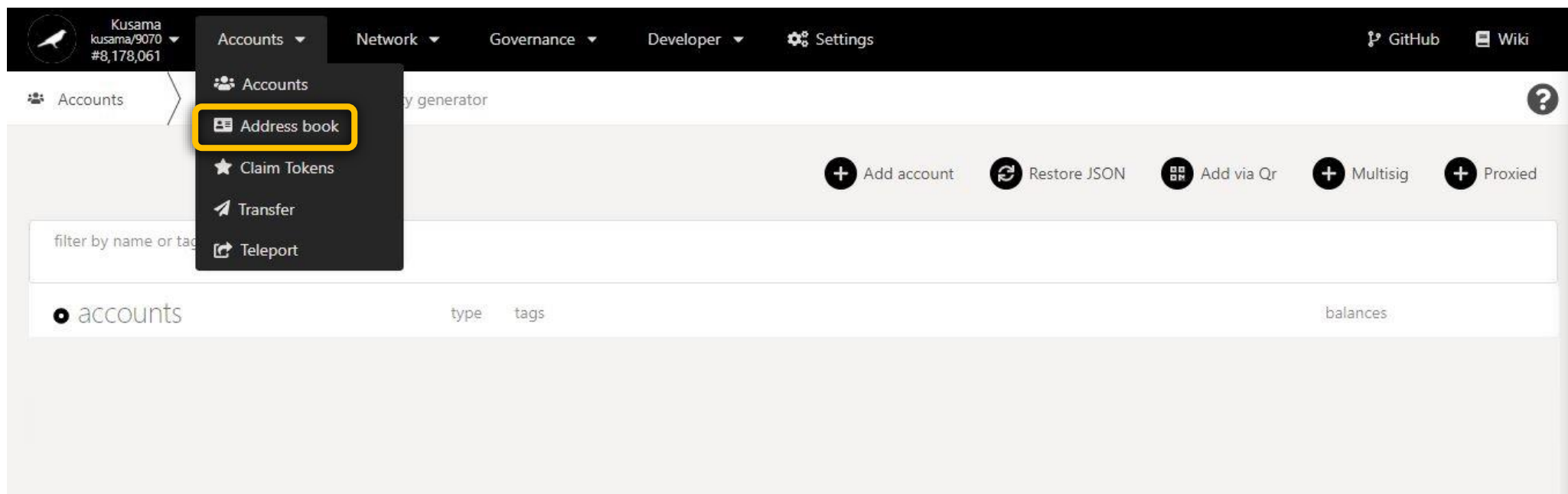
case sensitive ?
Yes

2. Fill-in or select the information required.

Start generation

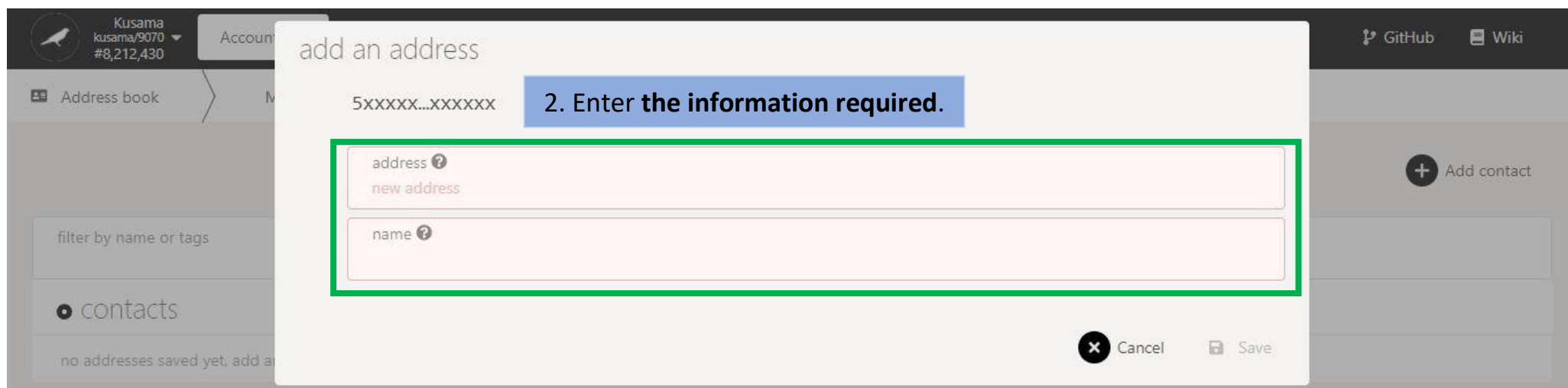
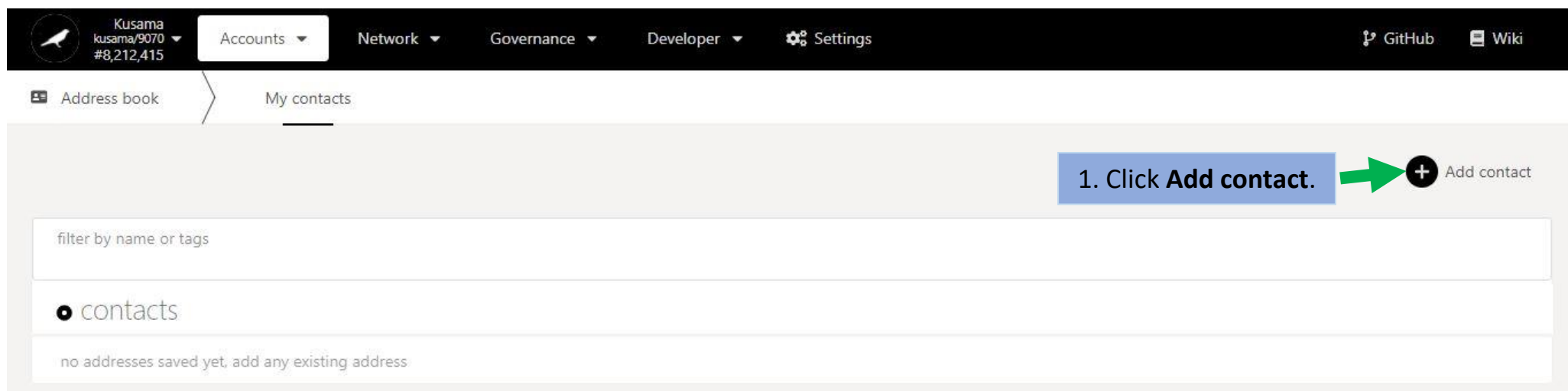
3. Click on Start generation to complete the procedure.

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



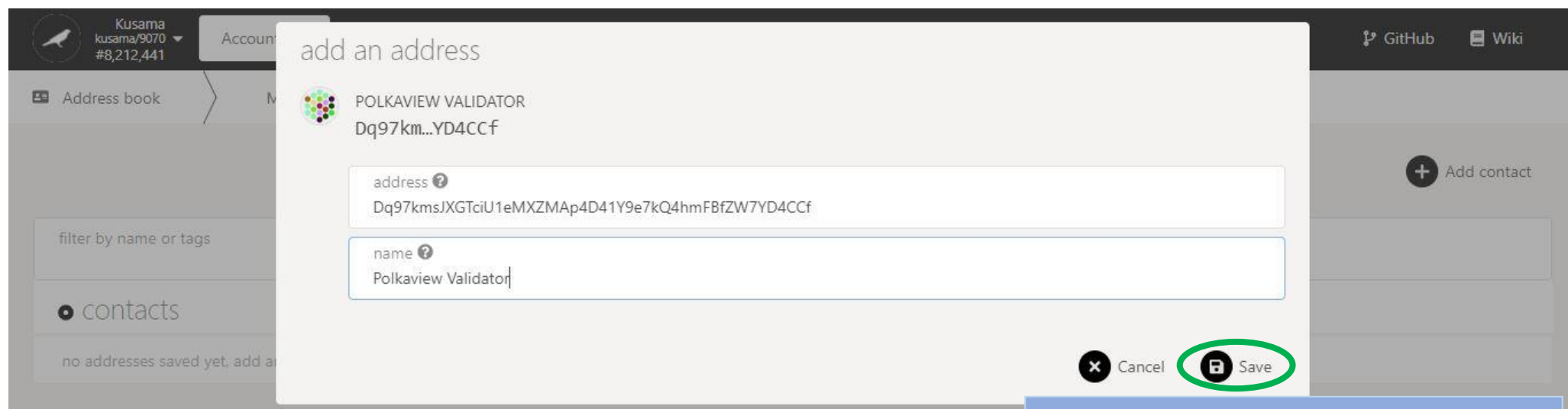
GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.1

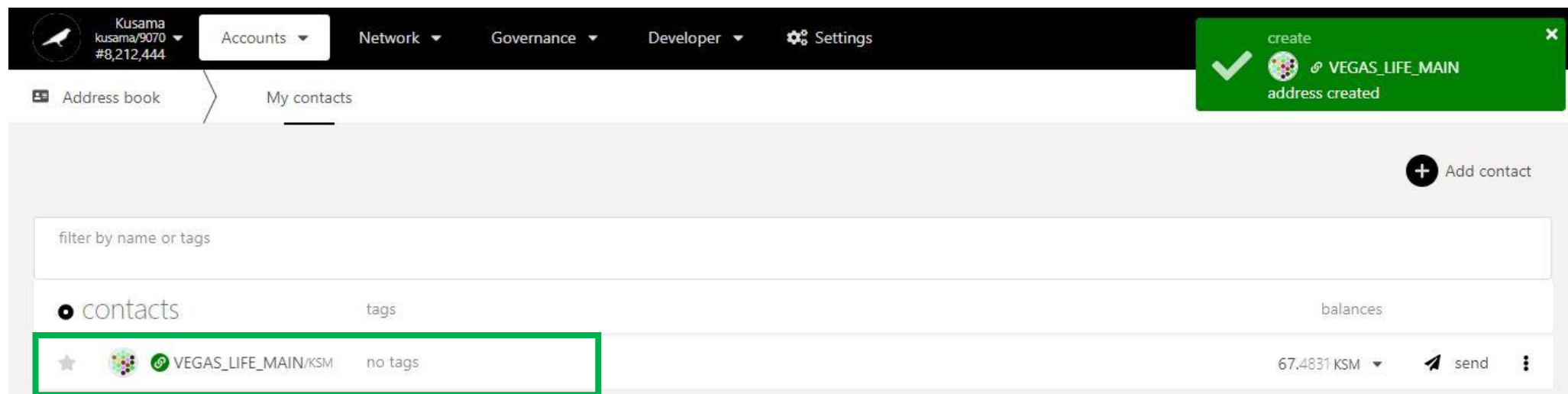


GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 1.1



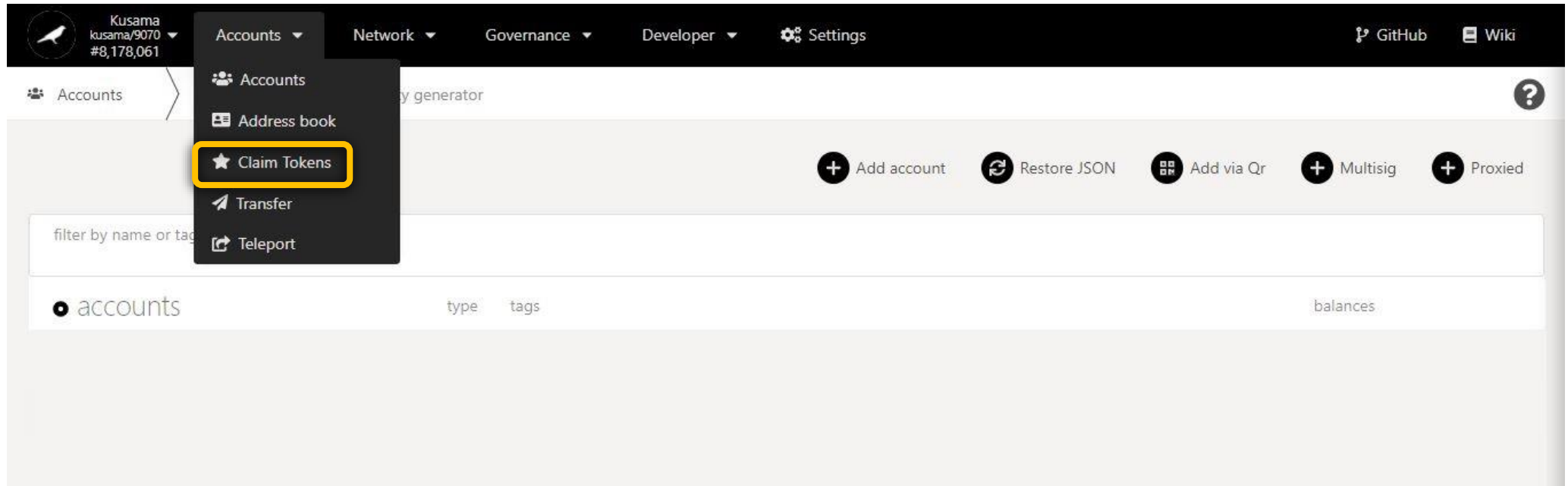
3. Click on **Save** to complete the procedure.



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 1.1




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.

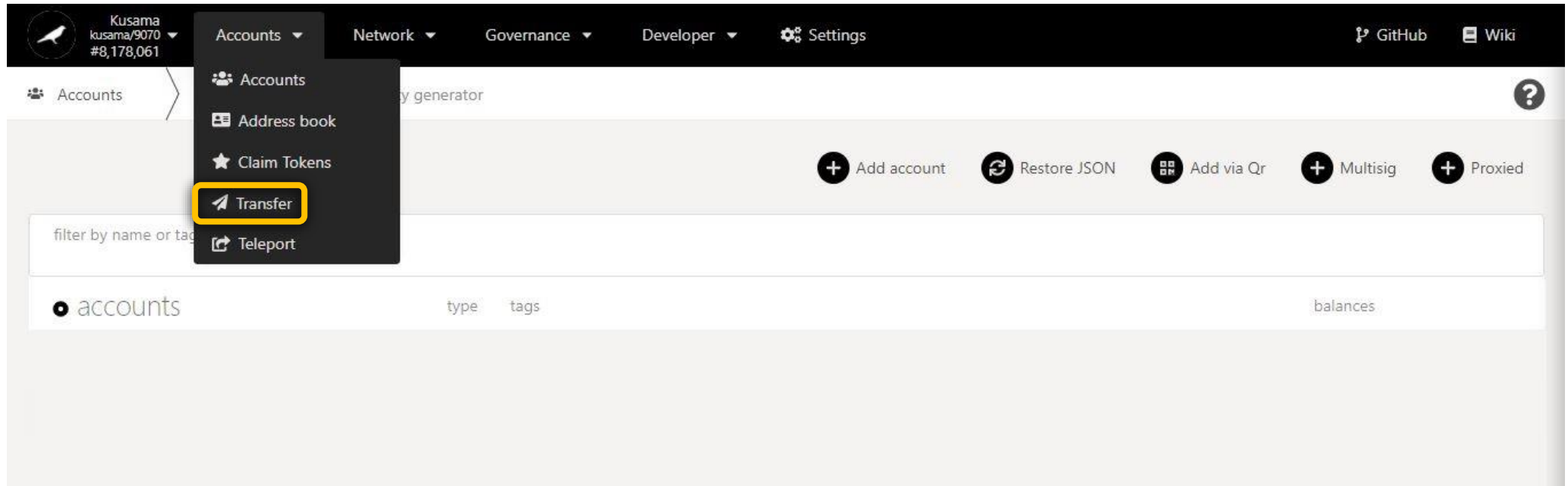
Pre-sale ethereum address ?

➔ 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. A green arrow points from the 'claim your Kusama' section to the 'send from account' option. Another green arrow points from the 'Transfer with account keep-alive checks' toggle to the 'Keep-alive checks' text box. A third green arrow points from the 'Make Transfer' button to the '2. Click on Make Transfer' instruction box.

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.

amount 0 KSM

existential deposit 33.3333 micro

Transfer with account keep-alive checks ☒

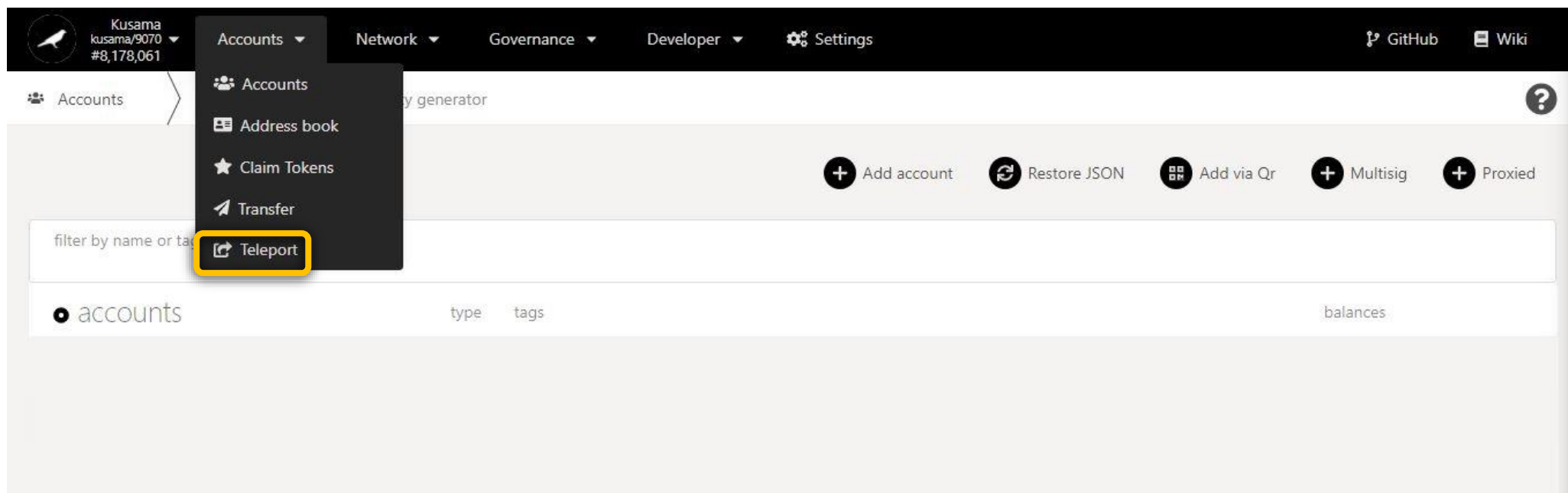
With the keep-alive option set, the account is protected against removal due to low balances.

Cancel Make Transfer

2. Click on Make Transfer to complete the procedure.

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.

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



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

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.

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

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