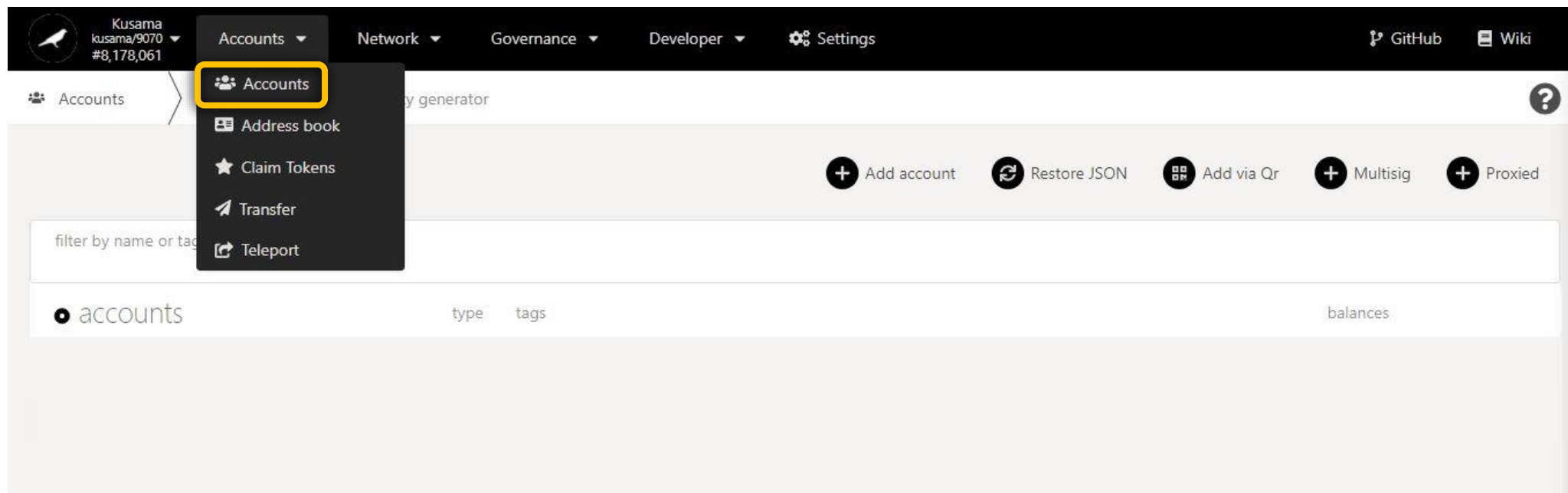


[PART I: Accounts \(Basic\)](#)

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.

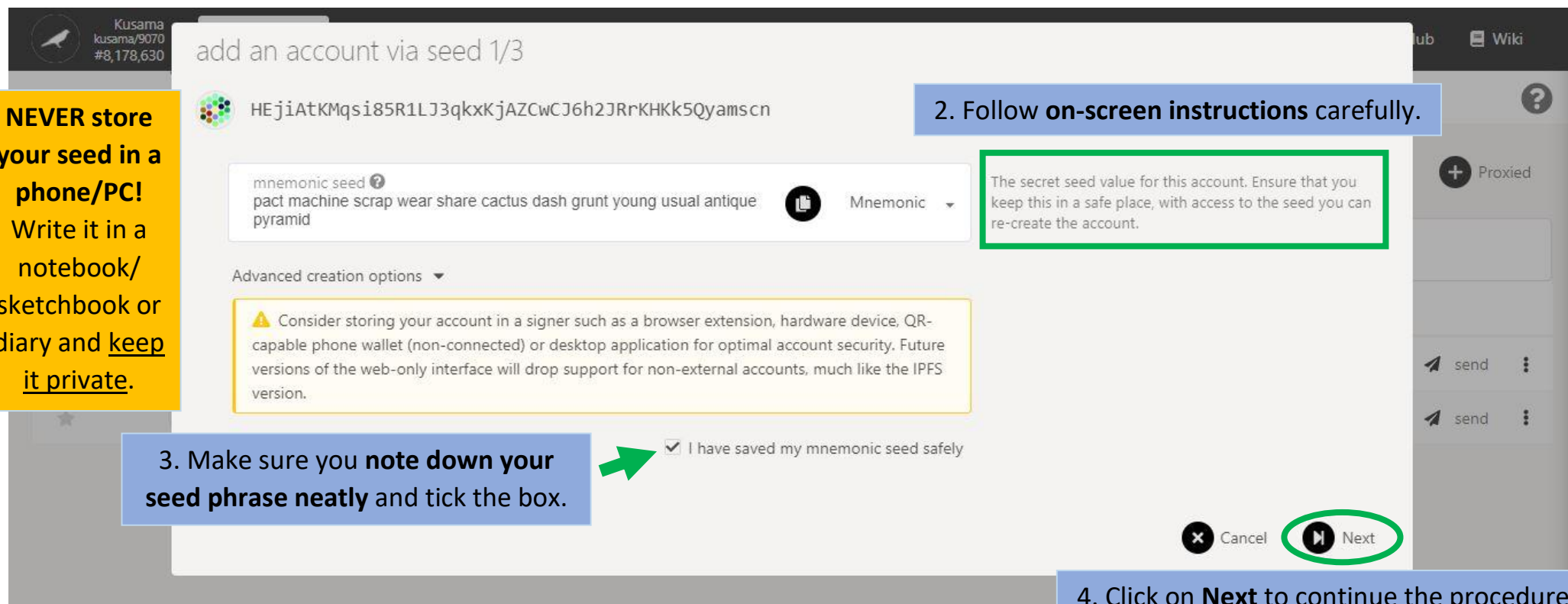
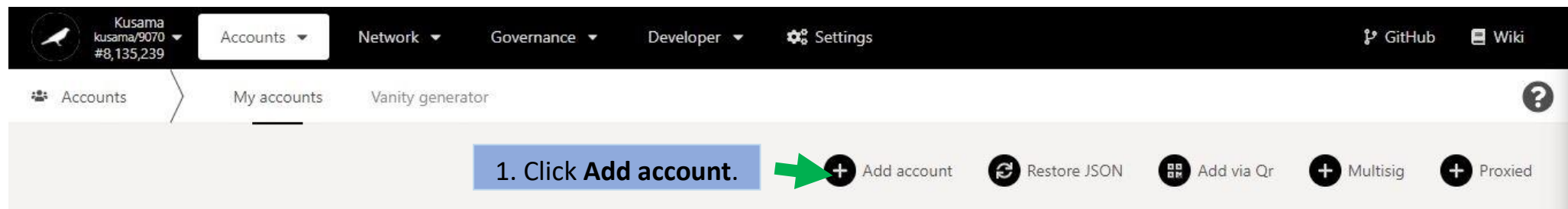
Overview of account's **KSM balances**.

The screenshot shows the 'Accounts' section of the Polkadot-JS interface. The top navigation bar includes 'Kusama', 'kusama/9070', '#8,135,239', and tabs for 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. On the right, there are links for 'GitHub' and 'Wiki'. Below the navigation bar, the 'Accounts' section is active, showing 'My accounts' 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 with 'injected' type and 'no tags'. To the right of the table, there is a 'balances' section showing three KSM balances: 10.3509 KSM, 1.0376 KSM, and 11.3886 KSM. Annotations with green arrows point to various parts of the interface: one points to the 'Wiki' link, another to the 'Add account' button, a third to the 'type' column header, a fourth to the 'tags' column header, a fifth to the account names, and a sixth to the 'balances' section.

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

balances	
10.3509 KSM	send
1.0376 KSM	send
11.3886 KSM	

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



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:

- Yellow box:** Avoid using the same password for all your accounts.
- Blue box:** 5. Follow the new instructions carefully.
- 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.
- Blue box:** 6. Click on **Next** to continue the procedure.

Kusama
kusama/9070
#8,178,651

Accounts

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 2.0

Star your most important accounts.

filter by name or tags

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

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.**

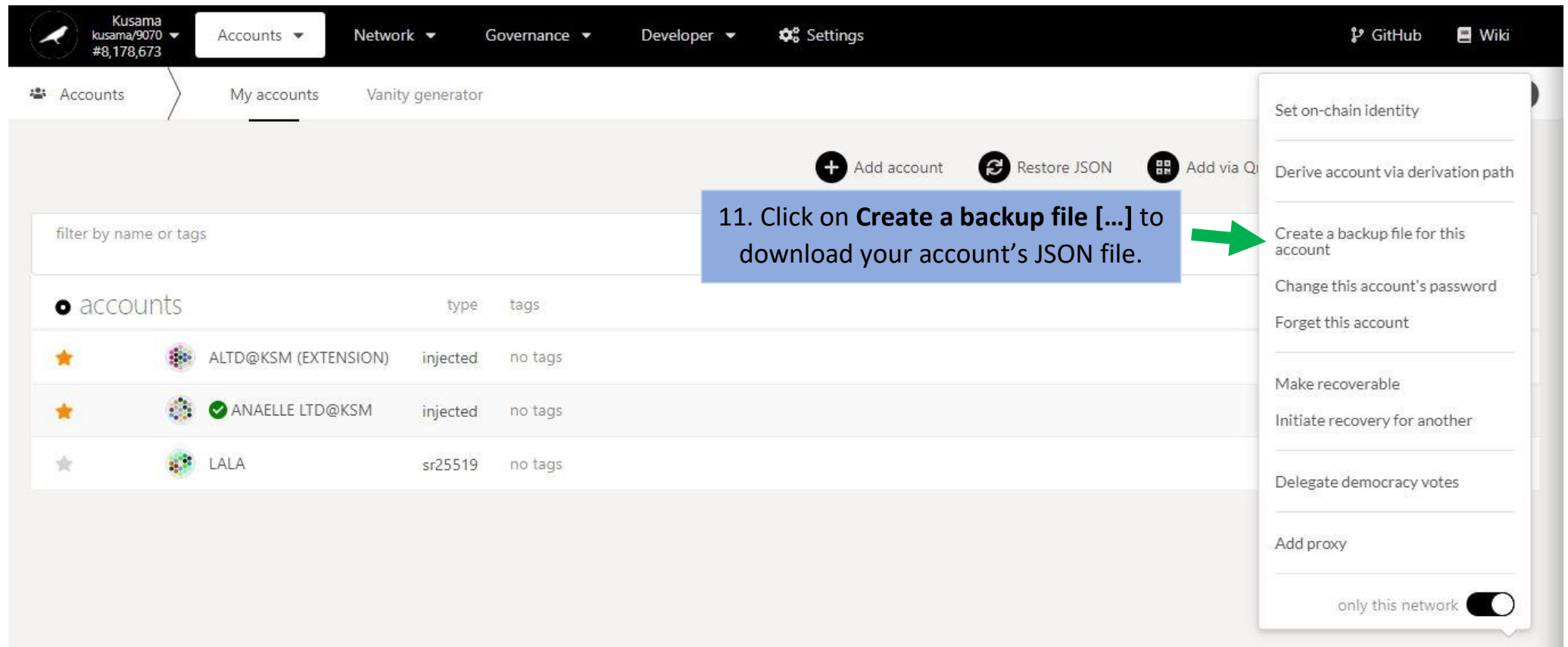
A JSON (backup) file has also been created for your account! It should be stored in a device that is not connected to the internet.

HEjiAtKMqsi85R1....json

Show all ×

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

Version 2.0



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, 'ALTD@KSM (EXTENSION)', is selected, and a context menu is open over it. The menu options are: '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', and a toggle for 'only this network'.

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 2.0

The screenshot shows the Polkadot-JS Accounts interface. A modal dialog titled "backup account" is open, displaying instructions for creating a backup. The dialog includes a text box for a password and two buttons: "Cancel" and "Download". The "Download" button is circled in green. A blue callout box highlights the "Download" button with the instruction: "13. Click on **Download** to complete the procedure." Another blue callout box at the top of the dialog reads: "12. Follow on-screen instructions carefully." The background shows a list of accounts, including "ALTD@K", "ANAELLE LTD@KSM", and "LALA".

backup account

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.

password ?

Cancel Download

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

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

The image shows two screenshots of the Polkadot-JS Accounts interface. The top screenshot shows the main Accounts page with a blue callout box that says "1. Click Restore JSON." with a green arrow pointing to the "Restore JSON" button. The bottom screenshot shows the "add via backup file" dialog with a blue callout box that says "2. Follow on-screen instructions carefully." and a green box highlighting the instructions for supplying a backed-up JSON file and the password.

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 2.0

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 account icon and name. It contains two input fields: 'backup file' with a file path and size, and 'password' with masked characters. A yellow warning box is present, and at the bottom are 'Cancel' and 'Restore' buttons. A green arrow points to the warning box, and a green circle highlights the 'Restore' button.

add via backup file

LALA
HEjiAt...yamscn

backup file ?
HEjiAtKMqsi85R1U3qkxKjAZCwCJ6h2JRrKHKk5Qyamscn.json (600 bytes)

password ?
.....

Supply a backed-up JSON file, encrypted with your account-specific password.

The password previously used to encrypt this account.

⚠ 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

3. Double-check
warning
messages.

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

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

Version 2.0

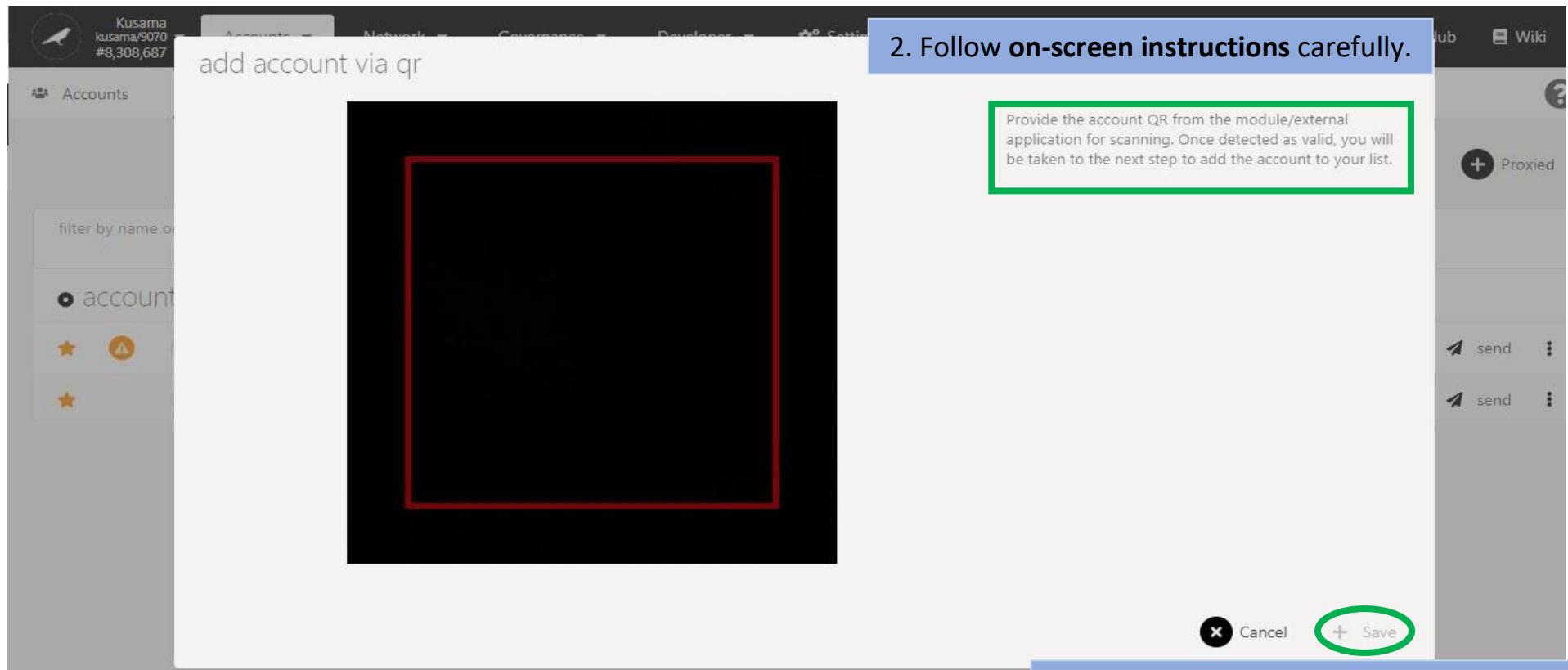
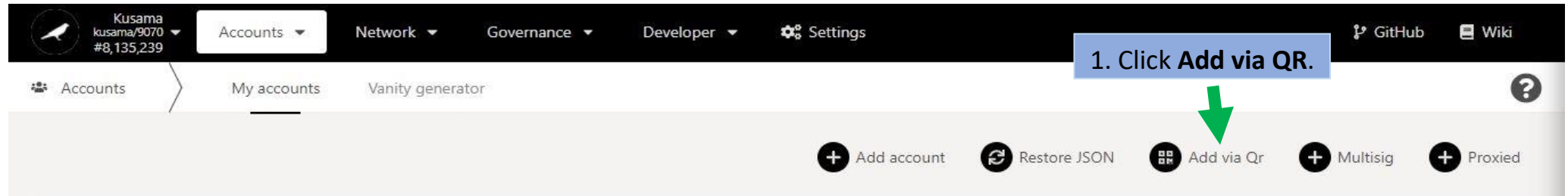
The screenshot shows the Polkadot-JS Accounts interface. At the top, there's a dark header with the Kusama logo, account name 'kusama/9070', and balance '#8,212,268'. Below this are tabs for 'Accounts', 'My accounts', and 'Vanity generator'. A green notification banner at the top right says 'restore LALA account restored'. Below the tabs, there are buttons for '+ Add account', 'Restore JSON', 'Add via Qr', '+ Multisig', and '+ Proxied'. A search bar labeled 'filter by name or tags' is present. The main section is titled 'accounts' and contains a table with columns: 'type', 'tags', and 'balances'. The table lists three accounts: '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, the total balance is shown as 11.4144 KSM.

	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 square area for scanning a QR code, outlined with a red border. To the right of this area, 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 code area, 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".

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

[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 two entries: 'ALTD@KSM (EXTENSION)' and 'ANAELLE LTD@KSM'. Overlaid on the screenshot are four blue callout boxes with instructions: 1. Click the Polkadot-JS extension icon. 2. Click the Polkadot-JS extension settings icon. 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. A green box with the text 'Possible solution: 1. Make sure that your Polkadot-JS extension settings allow QR Camera Access.' is also present. The Polkadot-JS extension settings menu is open, showing the 'Allow QR Camera Access' checkbox checked.

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.

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

[Troubleshooting 3/3]

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

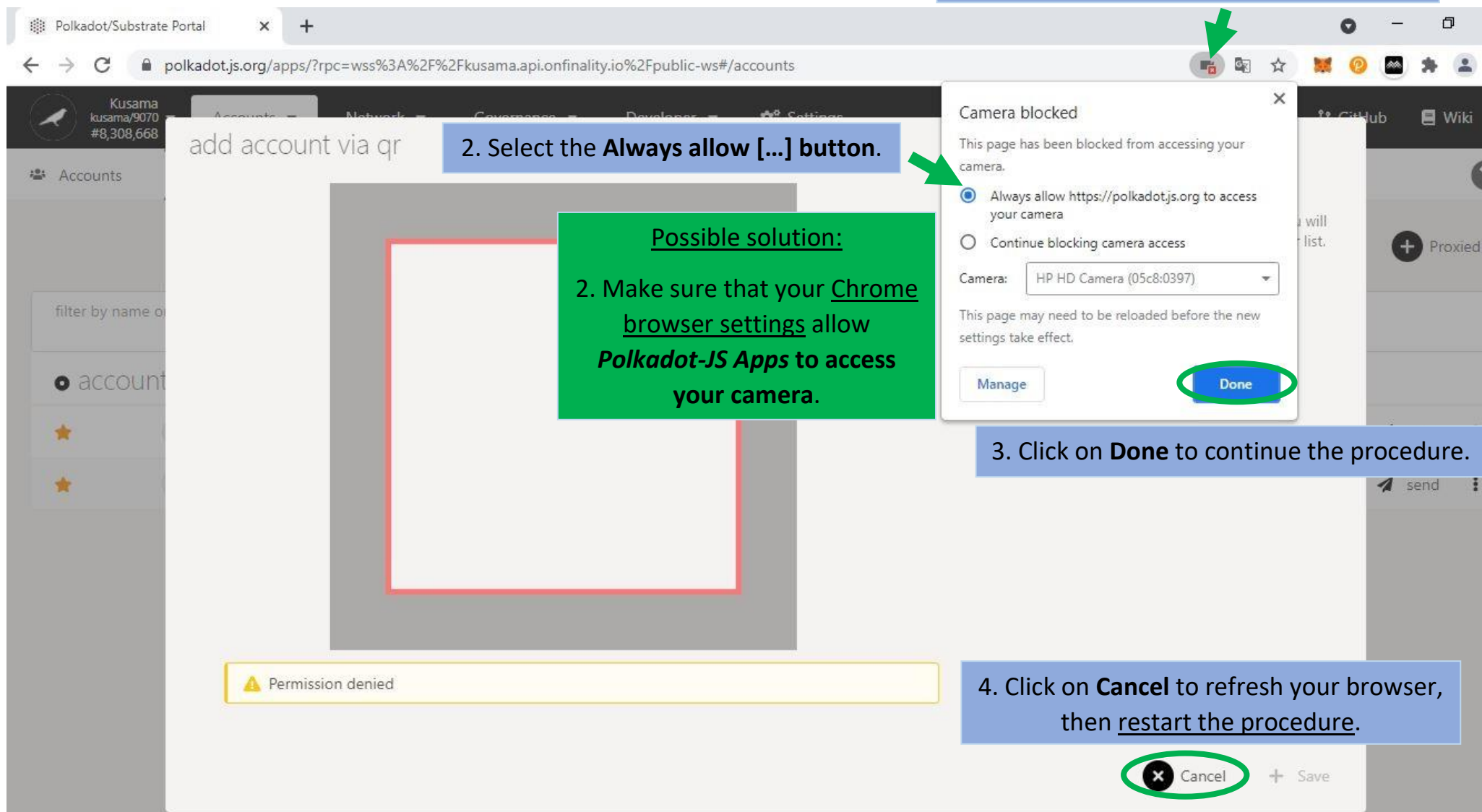
2. Select the **Always allow [...]** button.

Possible solution:

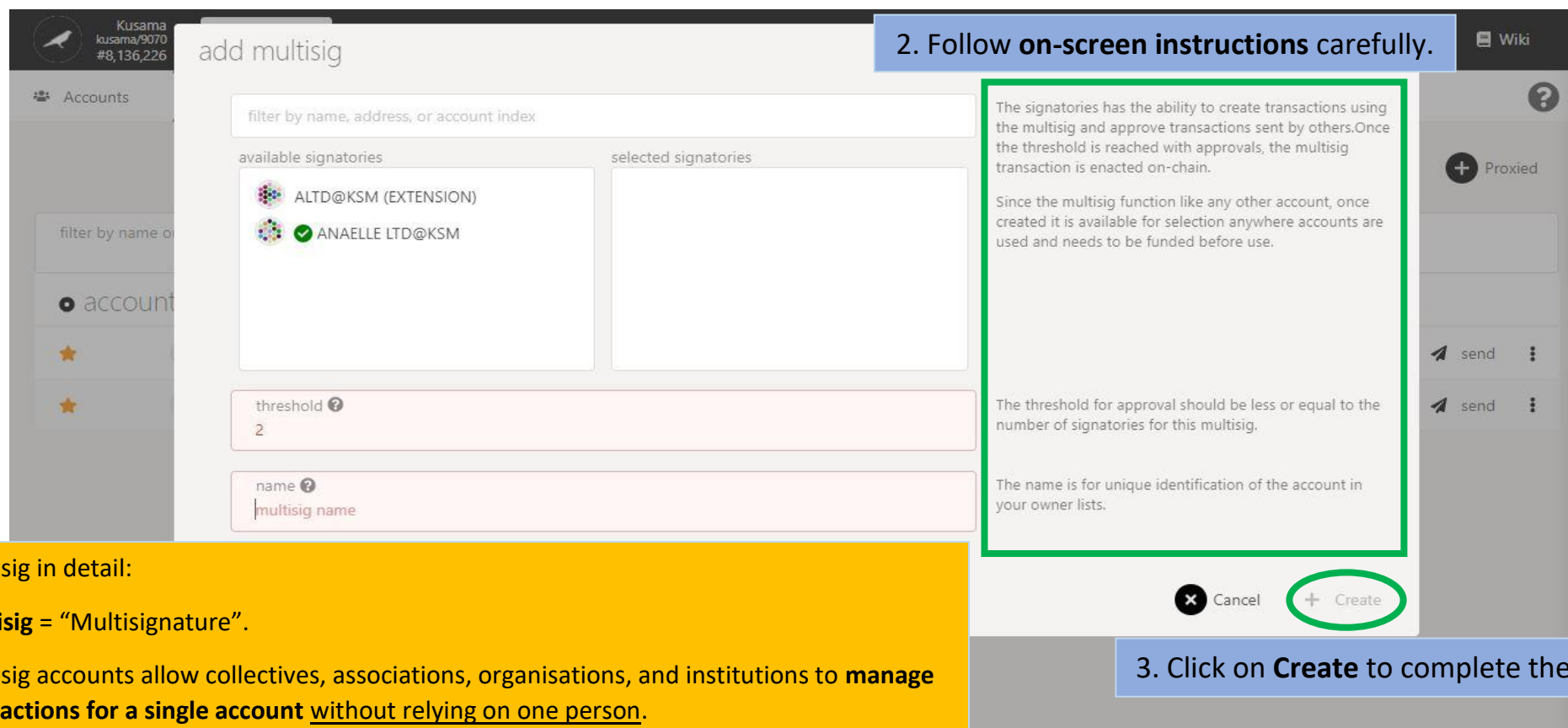
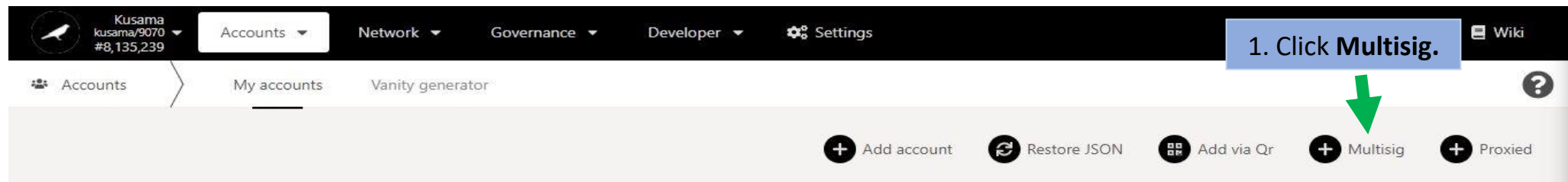
2. Make sure that your Chrome browser settings allow **Polkadot-JS Apps** to access your camera.

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.



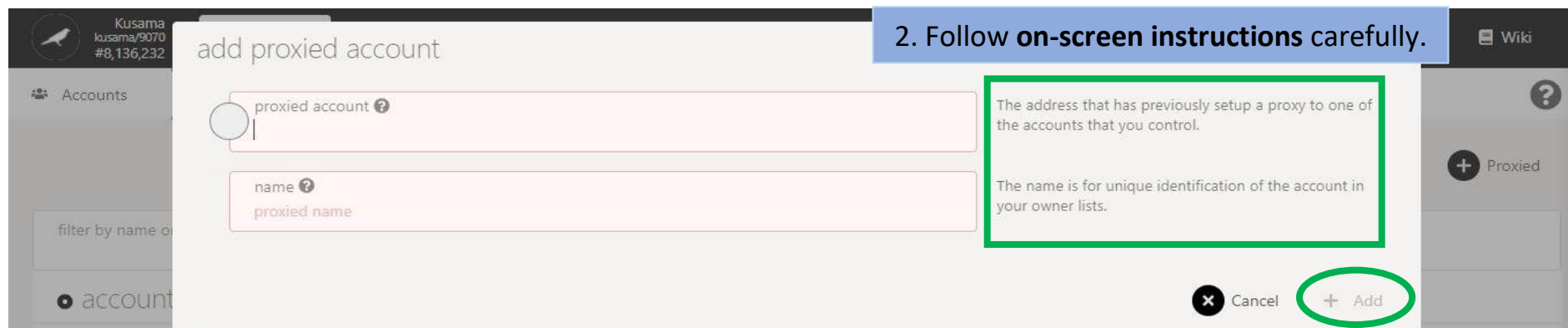
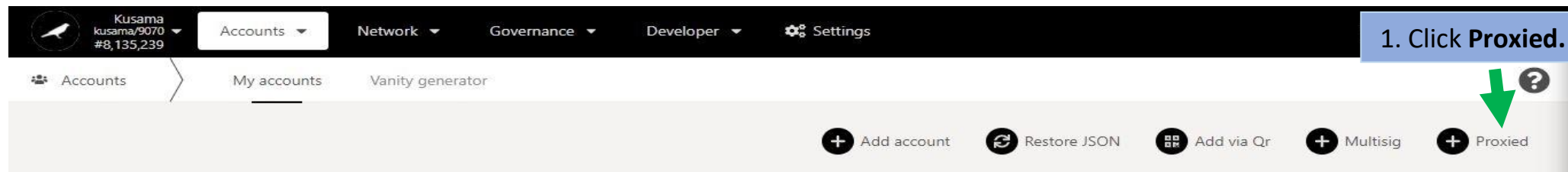
Multisig in detail:

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.

e) Set up a Polkadot-JS Proxied account.



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

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 'Accounts', 'Network', 'Governance', 'Developer', and 'Settings'. Below this, there's a sub-header with 'Accounts', 'My accounts', and 'Vanity generator'. The main area displays a list of accounts. 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 'ALTD@KSM (EXTENSION)')
- 3. Click on the **dropdown arrow** to view your detailed KSM balances. (Points to the dropdown arrow next to '10.3509 KSM')

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 2.0

Kusama
kusama/9070
#8,136,085

Accounts

Network

Governance

Developer

Settings

GitHub

Wiki

Accounts

My accounts

Vanity 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

g) Generate a vanity address.

The screenshot shows the 'Accounts' section of the Polkadot.js interface. The 'Vanity generator' tab is selected, indicated by a green arrow and a blue callout box: '1. Click Vanity generator.' The form contains three input fields: 'Search for' with 'Kusama' entered, 'keypair crypto type' with 'Schnorrkel (sr25519, recommended)' selected, and 'case sensitive' with 'Yes' selected. These fields are enclosed in a green rectangular box, with a blue callout box below it stating: '2. Fill-in or select the information required.' A 'Start generation' button, featuring a play icon, is circled in green. A blue callout box next to it says: '3. Click on Start generation to complete the procedure.'

Kusama
kusama/9070
#8,135,263

Accounts ▾ Network ▾ Governance ▾ Developer ▾ Settings

Accounts My accounts Vanity generator ?

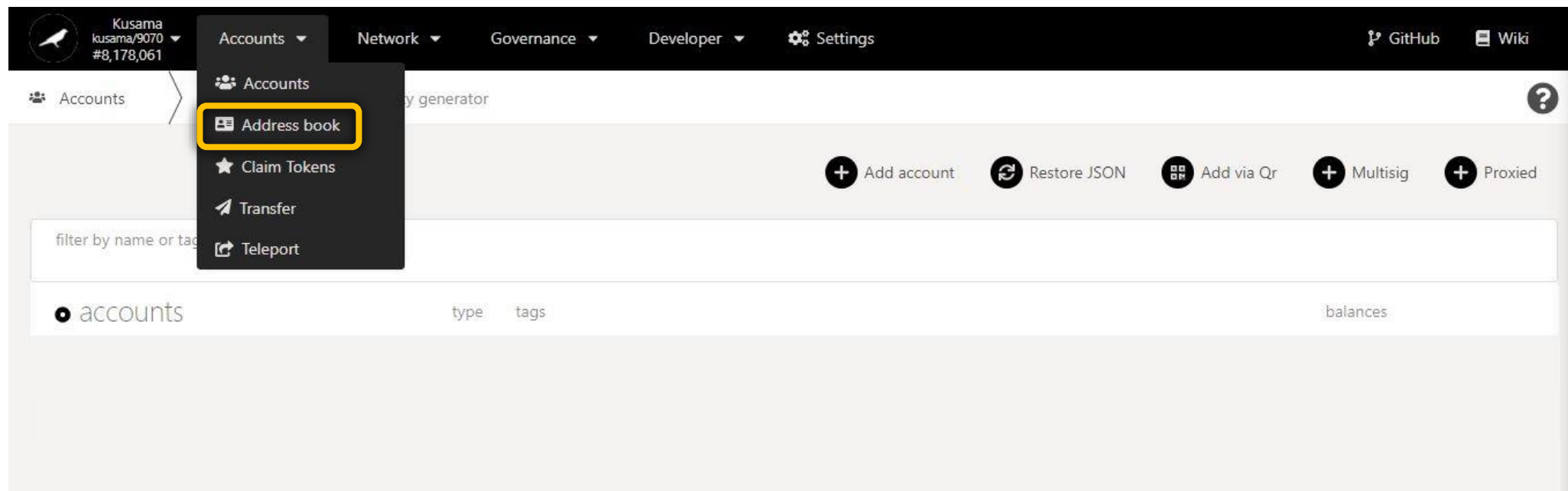
Search for ?
Kusama

keypair crypto type ?
Schnorrkel (sr25519, recommended)

case sensitive ?
Yes

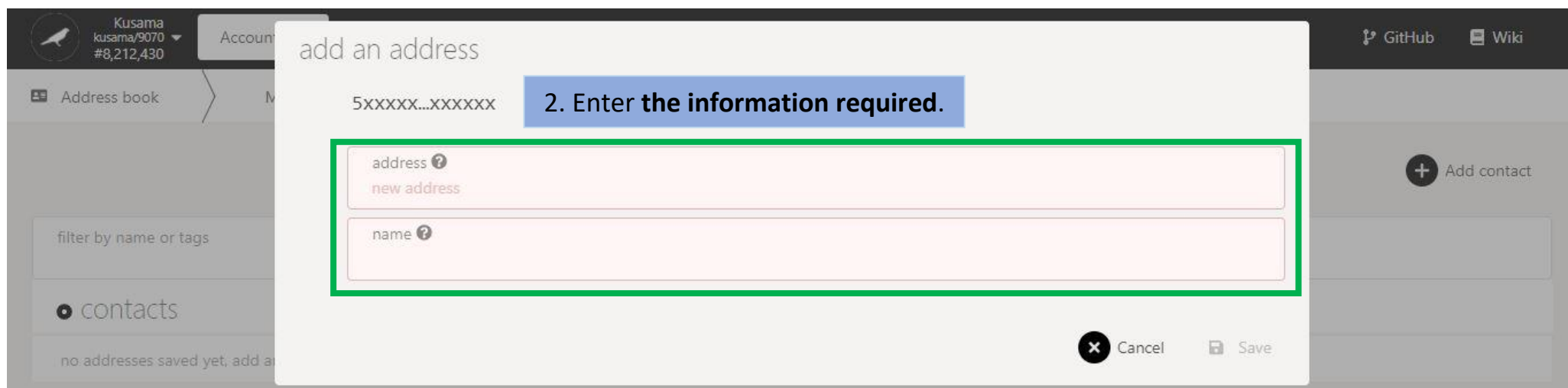
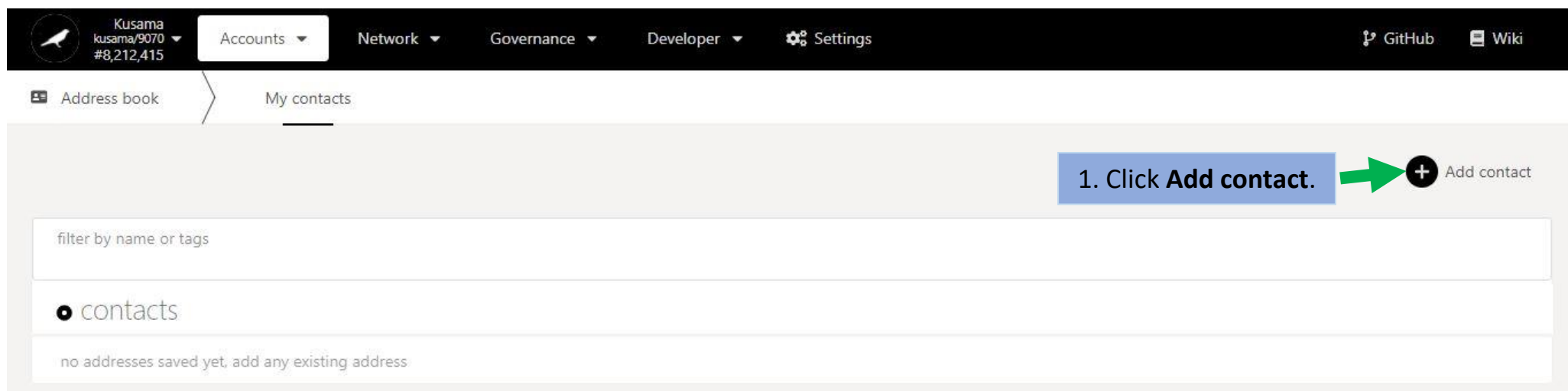
Start generation

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



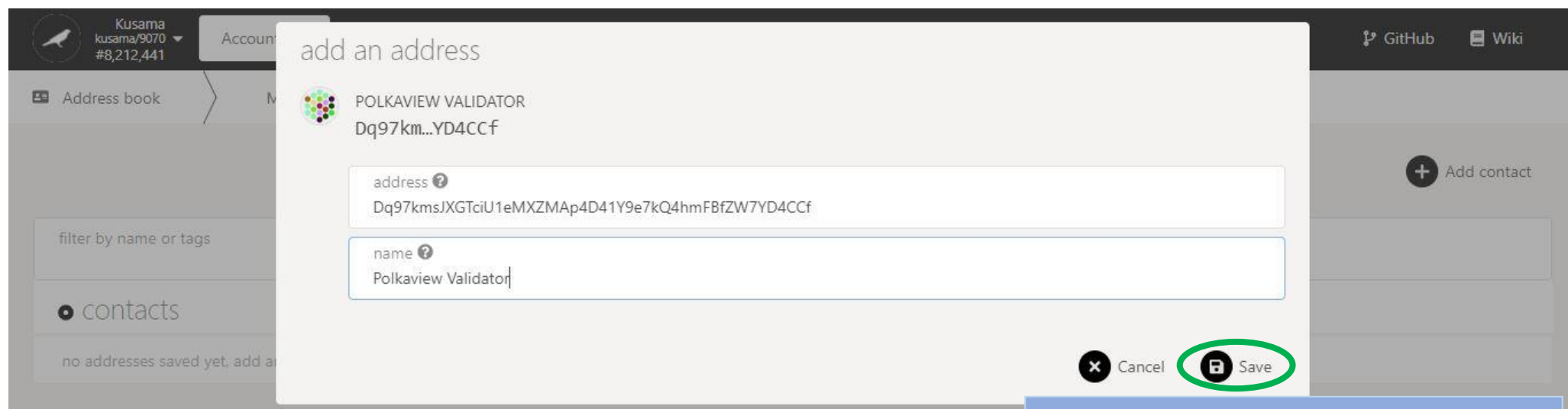
GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 2.0

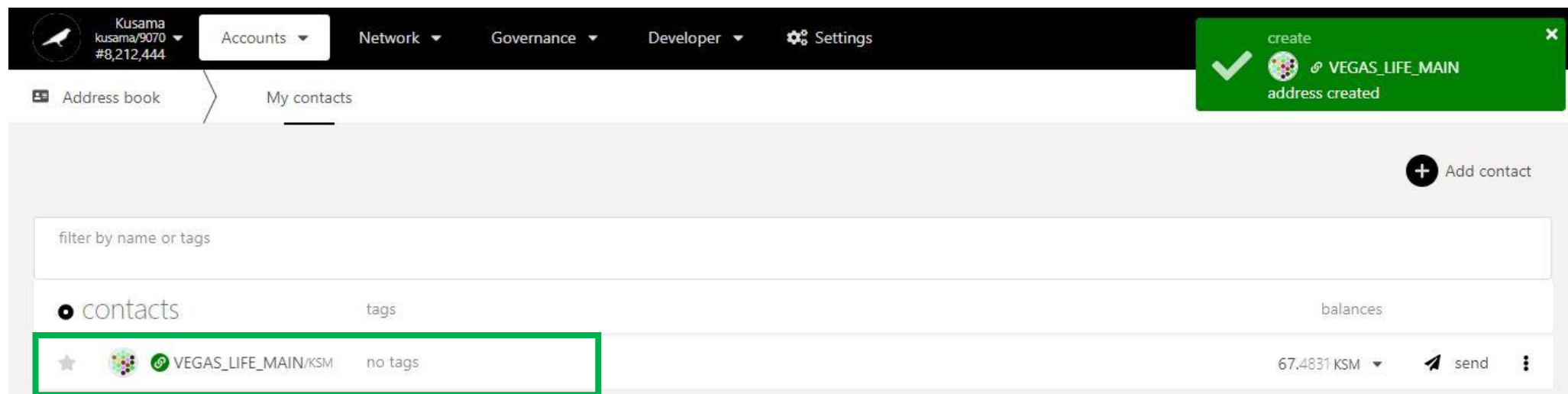


GUIDE TO POLKADOT-JS – PART I: Accounts (Basic)

Version 2.0



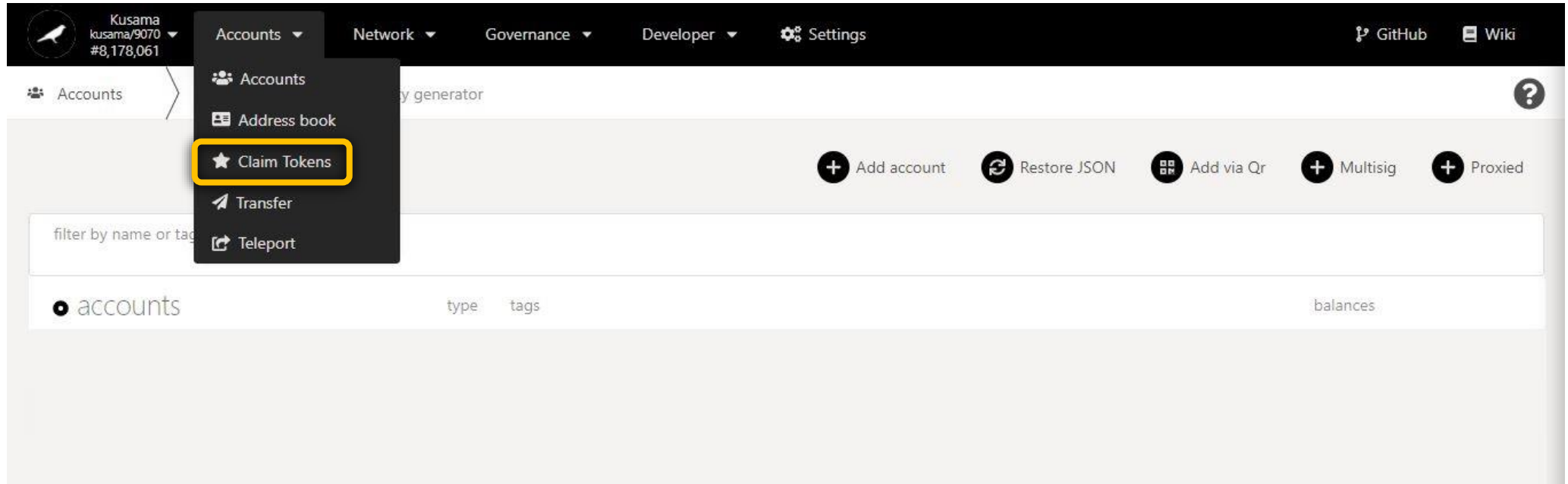
3. Click on **Save** to complete the procedure.



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 2.0




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.

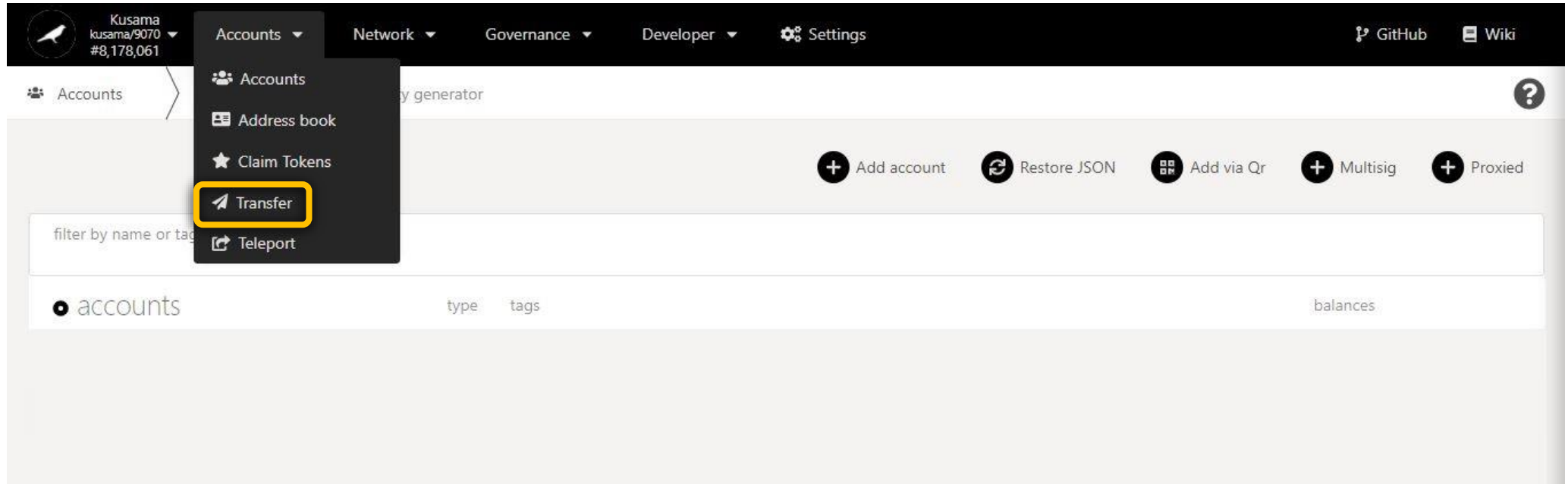
Pre-sale ethereum address ?

➔ 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 can be pasted manually or selected from your **My Accounts** or **My contacts** list.

The screenshot shows the 'send funds' interface in Polkadot.js. A green arrow points from the yellow instruction box to the 'send to address' field, which contains 'ALTD@KSM (EXTENSION)'. Another green arrow points from the yellow instruction box to the 'Transfer with account keep-alive checks' toggle, which is currently turned on. A green box highlights the explanatory text for the 'keep-alive' option. At the bottom right, the 'Make Transfer' button is circled in green. A blue box at the top right contains the instruction '1. Follow on-screen instructions carefully.' and another blue box at the bottom right contains '2. Click on **Make Transfer** to complete the procedure.'

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.

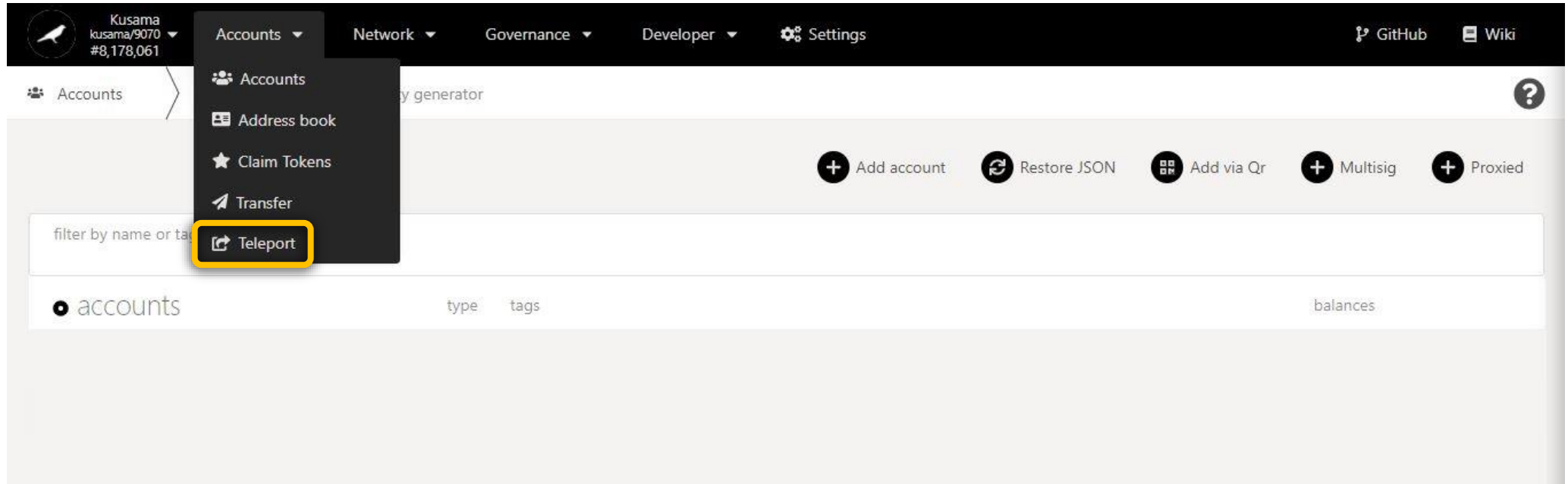
If the recipient account is new, the balance needs to be more than the existential deposit. Likewise if the sending account balance drops below the same value, the account will be removed from the state.

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

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.

The screenshot shows the 'teleport assets' dialog box in Polkadot-JS. It contains the following fields and information:

- send from account:** ANAELLE LTD@KSM (with a green checkmark icon). The account ID is HjcErRijmpoiBiKEHT3edPXM3NFycJogVwDPuByN...
- transferrable:** 1.0043 KSM
- destination chain:** Statemine (with a dropdown arrow).
- send to address:** ALTD@KSM (EXTENSION). The address is FegWwJq7AwgkYguRA796uvNVYburnURpBz7wZ9k...
- amount:** 1 (with a dropdown arrow set to KSM).
- destination transfer fee:** 7.9999 (with a dropdown arrow set to micro).
- destination existential deposit:** 3.3333 (with a dropdown arrow set to micro).

Annotations include:

- A green arrow pointing to the 'destination chain' field.
- A green arrow pointing to the 'destination existential deposit' field.
- A green box around the right side of the dialog containing explanatory text.
- A green circle around the 'Teleport' button at the bottom right.

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.

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

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