



## Tutorial B4: Acquiring network connection details

---


Estimated time: **10 minutes**

You may recall from tutorial [A5: Invoking a smart contract from an external application](#) that in order to connect to a Hyperledger Fabric network you need both a connection profile and a wallet. The connection profile allows client applications to specify *where* to connect; the wallet contains identities that allow client applications to specify *who* is connecting.

In this tutorial we will:

- Use the IBM Blockchain Platform web console to download your connection profile
- Use the IBM Blockchain Platform web console to download your identity
- Create a wallet from your identity

In order to successfully complete this tutorial, you must have first completed tutorial [B3: Enrolling with the network](#) in your web browser.

 **B4.1:** Expand the first section below to get started.

---

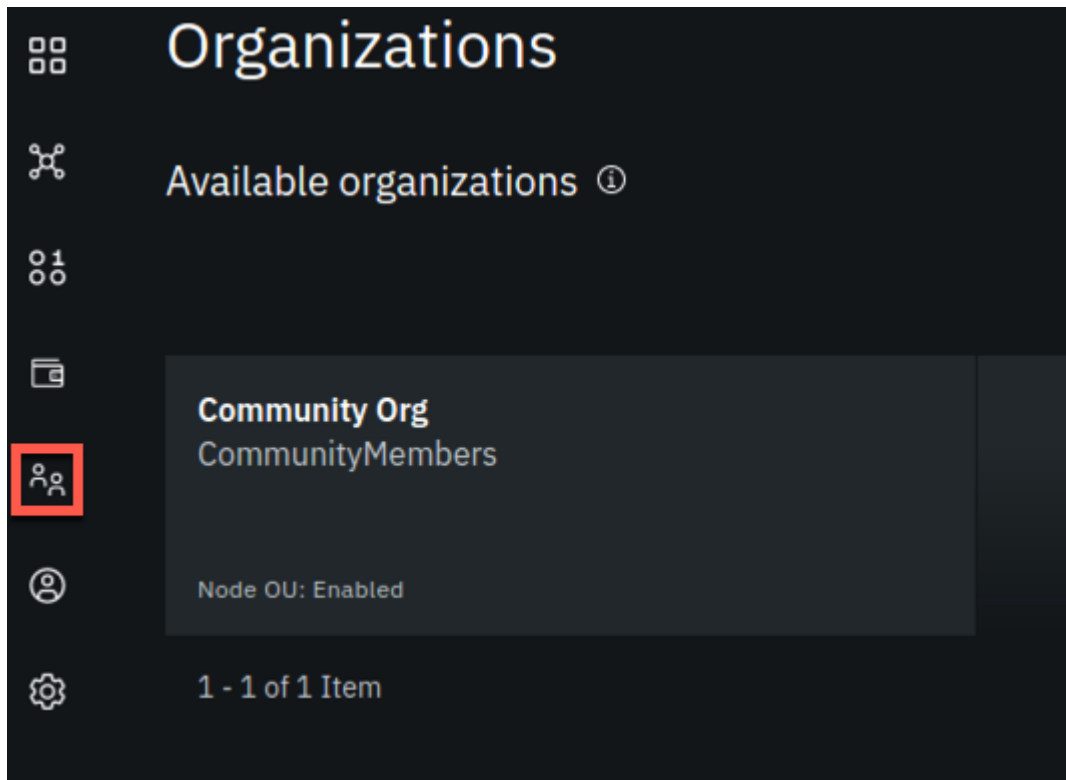
### ► Download the DriveNet connection profile

Now that we are fully enrolled on the DriveNet network, we will now download these resources using the IBM Blockchain Platform web console. This will allow Hyperledger Fabric client applications, including the IBM Blockchain platform VS Code extension, to connect and use it.

You were given a connection profile when you first registered with DriveNet in tutorial [B2: Discovering the network](#). This is a useful convenience, but you will not always be given such a file when you register for a new network. We'll therefore now go through the process for downloading one from the IBM Blockchain Platform web console.

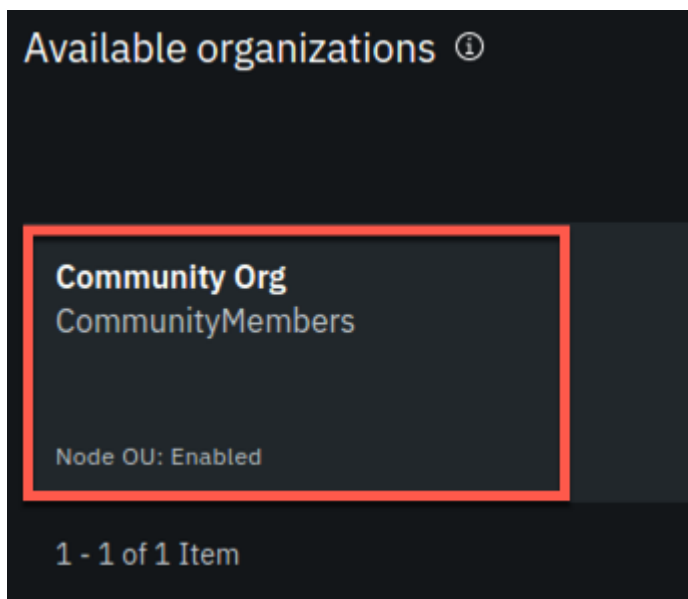
To do this, we need to browse to the Organizations view in the web console.

 **B4.2:** Click the 'Organizations' icon in the icon bar.



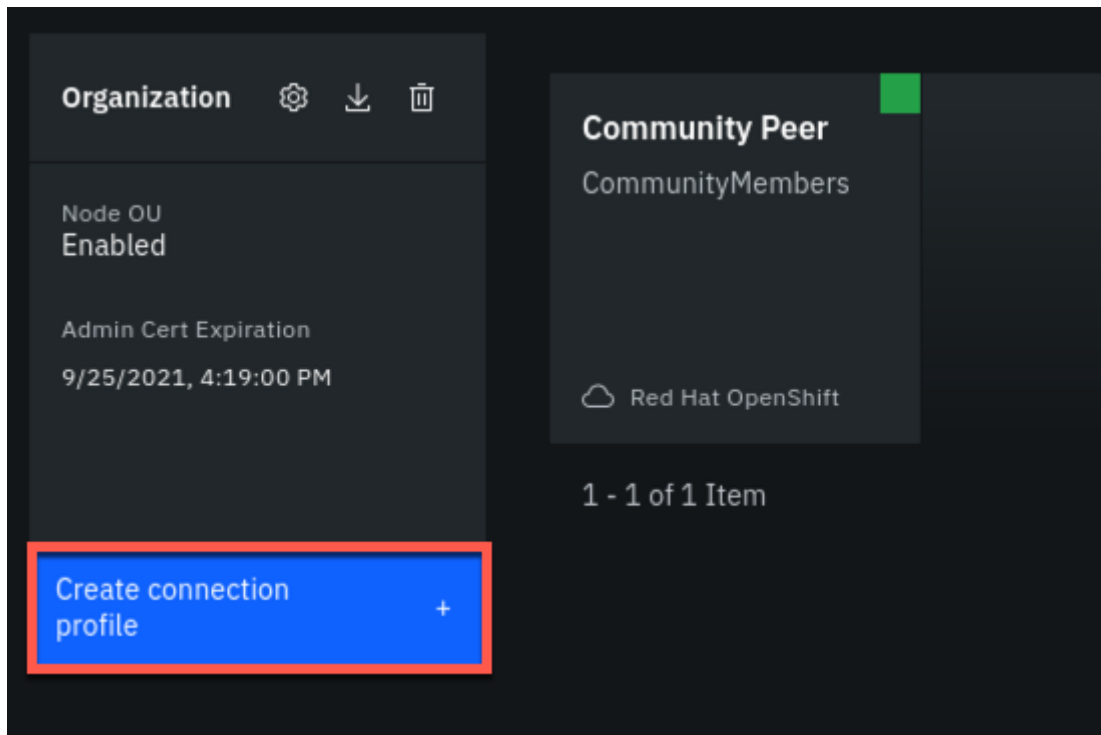
The Organizations view shows you the organizations for which you have access, and is where you can download your connection profile.

 **B4.3:** Click the 'Community Org' tile.



The screen will now show the peers that Community Org is managing, and also give us the ability to generate our connection profile.

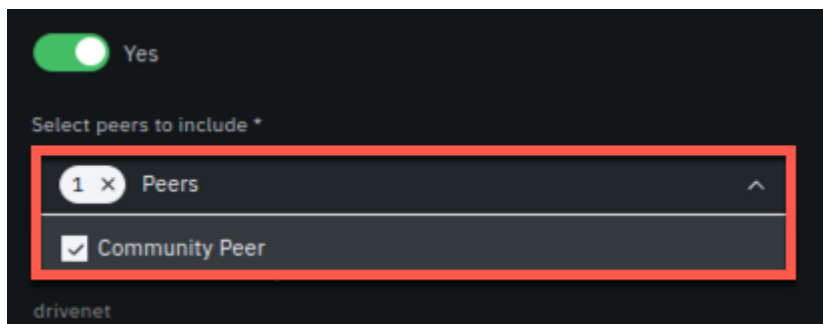
 **B4.4:** Click 'Create connection profile'.



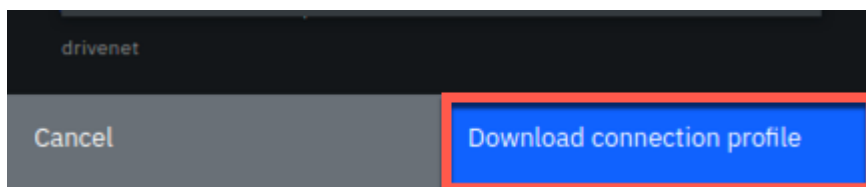
This reveals a side panel that allows us to customize the connection profile. We need to select the peer(s) that our client applications will be accessing.

**B4.5:** Under the 'Select peers to include section', click 'Peers' and select 'Community Peer'.

Notice that as you select Community Peer, the text underneath the peer confirms that this will allow you to connect to the 'DriveNet' network.

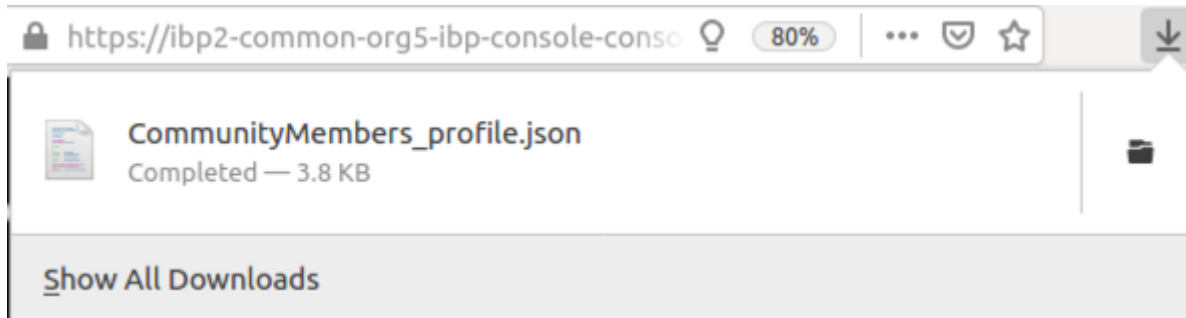


**B4.6:** Click the 'Download connection profile' button.



The connection profile will be downloaded. The file will be called 'CommunityMembers\_profile.json'.

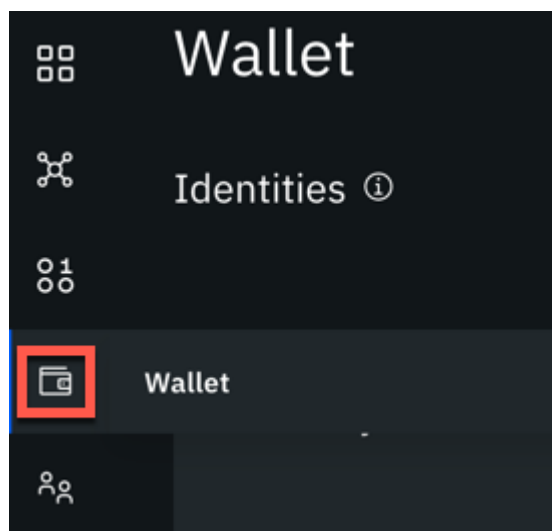
Keep this file safe; we will use it in later tutorials.



### ► Download your identity file

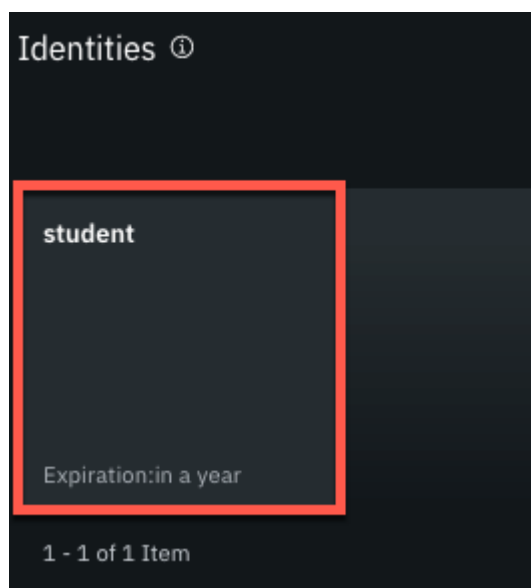
In tutorial [B3: Enrolling with the network](#) we enrolled our Fabric user and stored the certificate in a wallet in the web browser's storage. The Wallet view allows us to extract this certificate so we can import it into other wallets for use by other applications.

**B4.7:** Click the 'Wallet' icon in the icon bar to show the Wallet view.



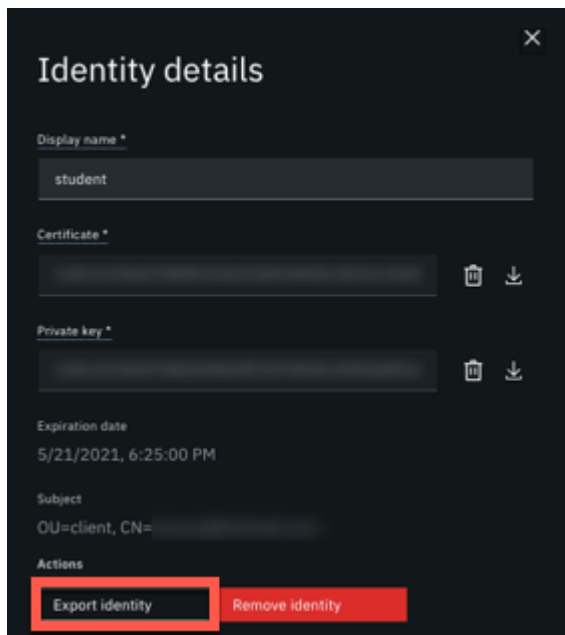
There will only be one identity listed in this view, and is represented by the name you assigned earlier.

**B4.8:** Click on the 'student' tile.

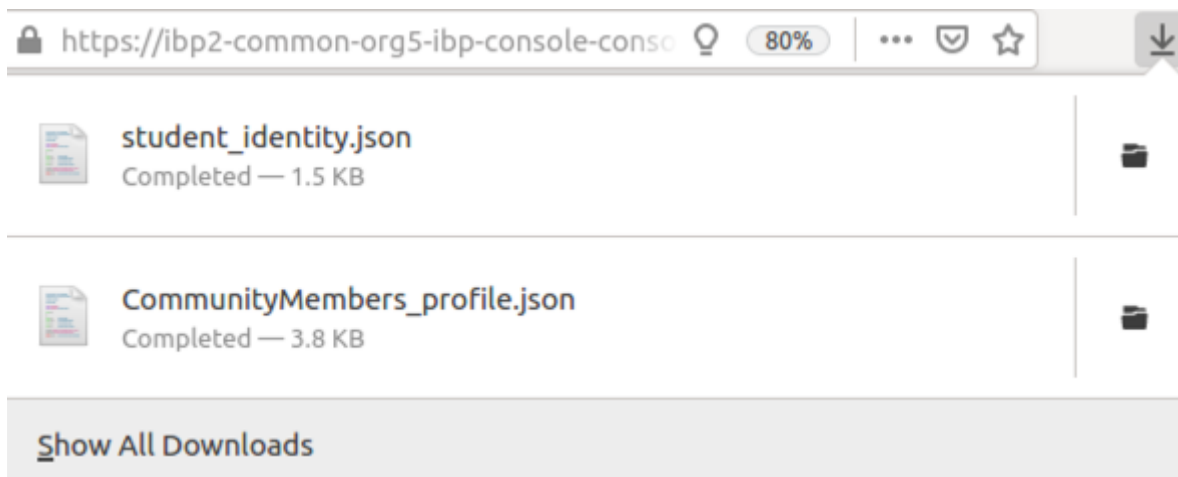


This will reveal a side panel that shows information on the identity we selected.


 **B4.9:** Click 'Export identity'.



A file called 'student\_identity.json' will be downloaded.



Keep this file safe along with the connection profile; we will use them both throughout these tutorials.

 **B4.10:** Close the sidebar and expand the next section to continue.

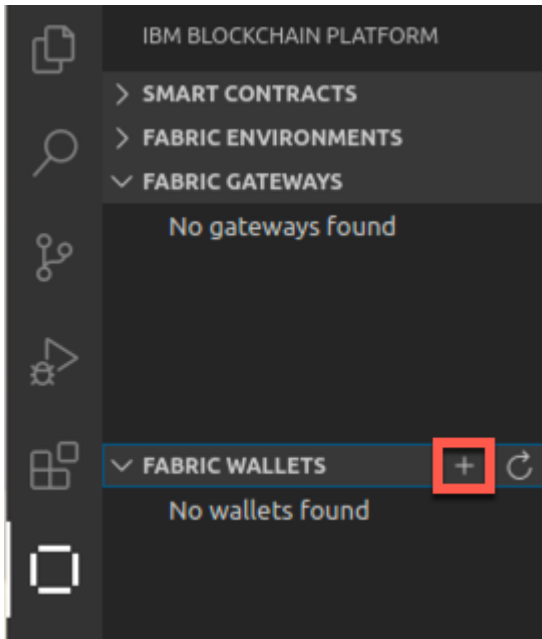
---

### ► Create a wallet from your identity

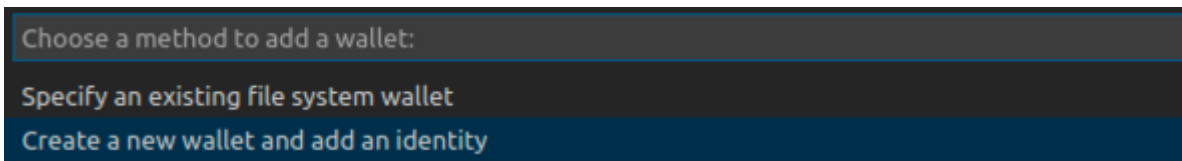
Now we have downloaded our identity, we need to store it in a wallet. Wallets are containers of user identities, and can take different forms depending on how they are going to be used. They can be stored on the file system, in memory or in a database. They can be backed by a hardware security module that provides secure storage of the private keys associated with an identity.

For our purposes we will use a simple file system wallet, and create it using the IBM Blockchain Platform VS Code extension. As we saw in tutorial [A5: Invoking a smart contract from an external application](#), wallets created by the VS Code extension can be exported for use in any client application.

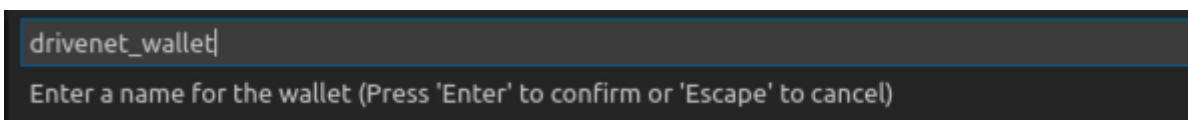
- B4.11: Switch to the VS Code application and if necessary, click the IBM Blockchain Platform icon to show the Fabric Wallets view in the side bar.
- B4.12: Hover over the Fabric Wallets view to reveal a '+' icon, and click it.



- B4.13: Click 'Create a new wallet and add an identity'.

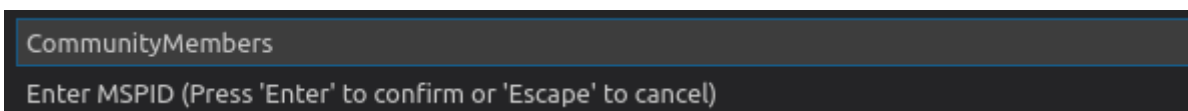


- B4.14: Enter the name `drivenet_wallet` and press Enter.

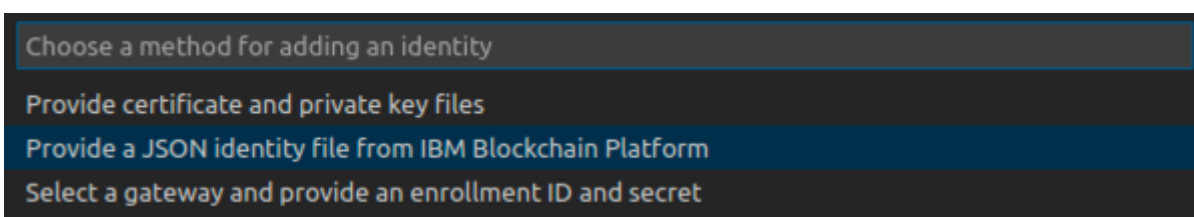


You will now be prompted to enter the MSPID of the identity. As you will recall from tutorial [B3: Enrolling with the network](#), our MSPID is called 'CommunityMembers'.

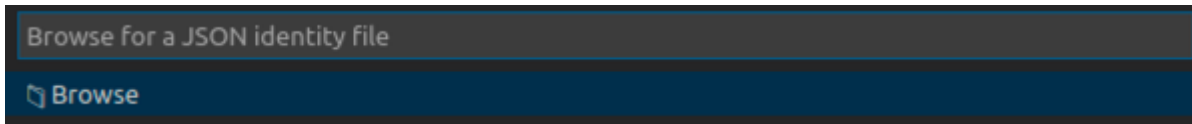
- B4.15: Enter `CommunityMembers` as the MSPID and press Enter. (This is case sensitive).



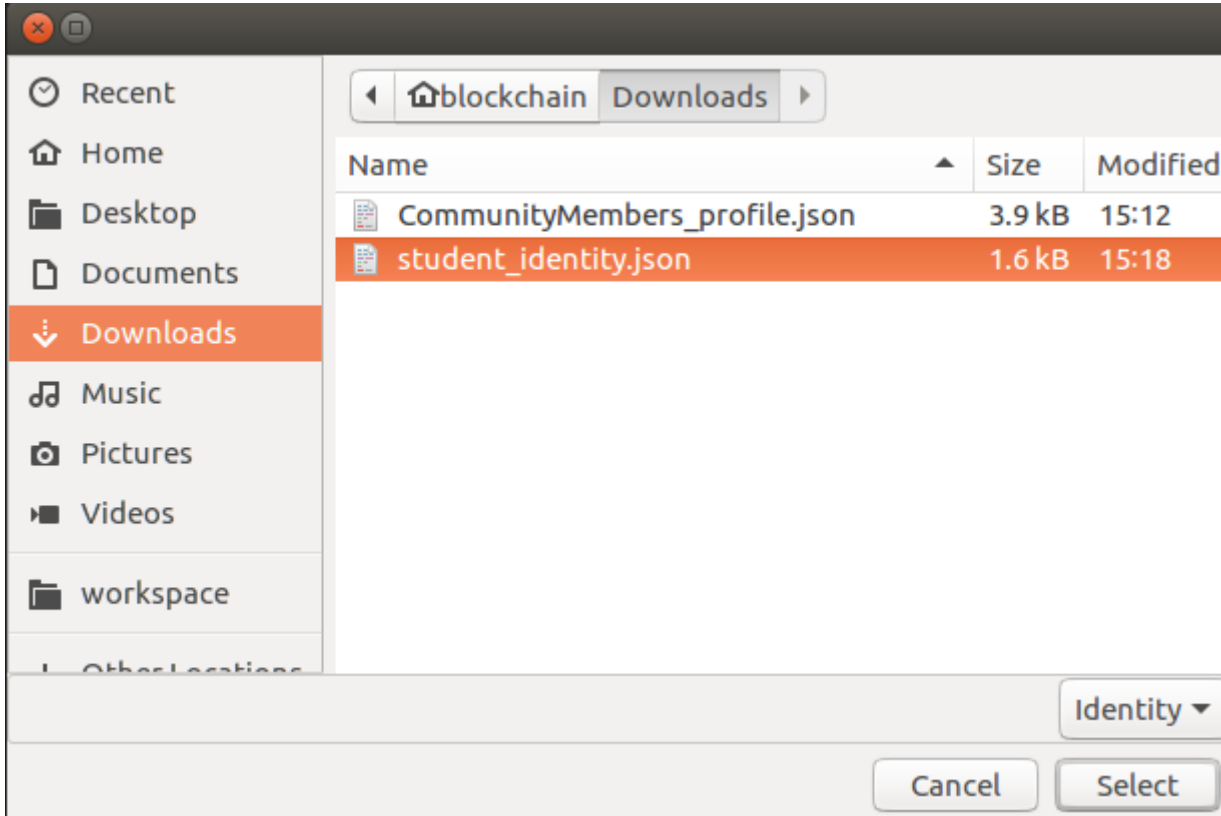
- B4.16: Click 'Provide a JSON identity file from IBM Blockchain Platform'.



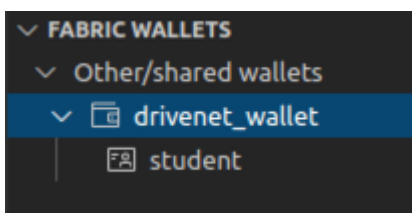
 B4.17: Click 'Browse'.



 B4.18: Locate the *student\_identity.json* file you downloaded earlier and click 'Select'.



The wallet will now be created, and displayed in the Fabric Wallets view. It will contain your identity.



Your wallet is now ready for use.

## Summary

In this tutorial we used the IBM Blockchain Platform web console to download our connection profile and identity, and imported the identity into a new wallet within the IBM Blockchain Platform VS Code extension.

We are now ready to use the connection profile and wallet to connect to DriveNet and submit transactions. We'll do that in the next tutorial; at the same time we'll use the IBM Blockchain Platform web console to see in more detail what's happening when we do this.

---

→ **B5: Invoking transactions on the network**