# Publish a Package

Before you can call functions in a Move package (beyond an emulated Sui execution scenario), that package must be available on the Sui network. When you publish a package, you are actually creating an immutable Sui object on the network that anyone can access.

To publish your package to the Sui network, use the publish CLI command in the root of your package. Use the --gas-budget flag to set a value for the maximum amount of gas the transaction can cost. If the cost of the transaction is more than the budget you set, the transaction fails and your package doesn't publish.

Beginning with the Sui v1.24.1 release , the --gas-budget option is no longer required for CLI commands.

If the publish transaction is successful, your terminal or console responds with the details of the publish transaction separated into sections, including transaction data, transaction effects, transaction block events, object changes, and balance changes.

In the Object Changes table, you can find the information about the package you just published in the Published Objects section. Your response has the actual PackageID that identifies the package (instead of ) in the form 0x123...ABC .

Your currently active address now has three objects (or more, if you had objects prior to this example). Assuming you are using a new address, running the sui objects command reveals what those objects are.

The objectId field is the unique identifier of each object.

Now that the package is on chain, you can call its functions to interact with the package. You can use the sui client call command to make individual calls to package functions, or you can construct more advanced blocks of transactions using the sui client ptb command. The ptb part of the command stands for programmable transaction blocks . In basic terms, PTBs allow you to group commands together in a single transaction for more efficient and cost-effective network activity.

For example, you can create a new Sword object defined in the package by calling the new_sword function in the my_module package, and then transfer the Sword object to any address:

You can pass literal addresses and objects IDs by prefixing them with '@'. This is needed to distinguish a hexadecimal value from an address in some situations.

For addresses that are in your local wallet, you can use their alias instead (passing them without '@', for example, --transfer-objects my_alias).

Depending on your shell and operating system, you might need to pass some values with quotes ( " ), for example: --assign "forge @" .

Make sure to replace , , and with the actual objectId of the Forge object, the address of the recipient (your address in this case), and the packageID of the package, respectively.

After the transaction executes, you can check the status of the Sword object by using the sui client objects command again. Provided you used your address as the , you should now see a total of four objects:

Congratulations! You have successfully published a package to the Sui network and modified the blockchain state by using a programmable transaction block.

## Related links