

Creating Accounts

You might want to create an account from a contract for many reasons. One example: You want to [progressively onboard](#) users, hiding the whole concept of NEAR from them at the beginning, and automatically create accounts for them (these could be sub-accounts of your main contract, such as `user123.some-cool-game.near`).

Since an account with no balance is almost unusable, you probably want to combine this with the token transfer from [the last page](#). You will also need to give the account an access key. Here's a way to do it:

```
Promise :: new ( "subaccount.example.near" . parse ( ) . unwrap ( ) ) . create_account ( ) . add_full_access_key ( env :: signer_account_pk ( ) ) . transfer ( 250_000_000_000_000_000_000 ) ;
```

// 2.5e23yN, 0.25N In the context of a full contract:

```
use

near_sdk :: { env , near_bindgen ,
AccountId ,
Balance ,
Promise } ;

const
INITIAL_BALANCE :
Balance
=
250_000_000_000_000_000_000 ;
// 2.5e23yN, 0.25N
```

[near_bindgen]

```
pub
struct
Contract
{ }
```

[near_bindgen]

```
impl
Contract
{
```

[private]

```
pub
fn
create_subaccount ( prefix :
AccountId )
->
Promise
```

```
{ let subaccount_id =
```

```
AccountId :: new_unchecked ( format! ( "{}.{}", prefix ,
```

```
env :: current_account_id ( ) ) ; Promise :: new ( subaccount_id ) . create_account ( ) . add_full_access_key ( env ::  
signer_account_pk ( ) ) . transfer ( INITIAL_BALANCE ) } } Things to note:
```

- add_full_access_key
- – This example passes in the public key of the human or app that signed the original transaction that resulted in this function call ([signer_account_pk](#)
-). You could also use [add_access_key](#)
- to add a Function Call access key that only permits the account to make calls to a predefined set of contract functions.

• **[private]**

- – if you have a function that spends your contract's funds, you probably want to protect it in some way. This example does so with a perhaps-too-simple [#\[private\]](#)
- macro.
- INITIAL_BALANCE
- uses the [Balance](#)
- type from near-sdk-rs. Today this is a simple alias for u128
- , but in the future may be expanded to have additional functionality, similar to recent [changes to the Gas type](#)
- . [Edit this page](#) Last updated on Aug 24, 2022 by [Damián Parrino](#) Was this page helpful? Yes No

[Previous](#) [Sending NEAR](#) [Next](#) [Deploying Contracts](#)