

Entry points

Entry points, or handlers are where messages or queries are handled by the contract.

All three of the functions we will be talking about are explicitly flagged as entry points, and excluded from being bundled in the library:

```
[cfg_attr(not(feature =
```

```
"library" ), entry_point)] pub
```

```
fn
```

```
instantiate ( deps :
```

```
DepsMut , _env :
```

```
Env , _info :
```

```
MessageInfo , msg :
```

```
InstantiateMsg , )
```

```
->
```

```
Result < Response ,
```

```
StdError
```

```
{ // ...etc } These handlers are:
```

1. Instantiate messages, as defined by the `InstantiateMsg`
2. struct, are handled by `instantiate`
3. .
4. Messages, as defined by the `ExecuteMsg`
5. enum, are handled by the `execute`
6. function, using a
7. pattern-matching `match`
8. statement.
9. Queries, as defined by the `QueryMsg`
10. enum, are handled by the `query`
11. function, using a pattern-match.

`execute` and `query` must exhaustively match every variant in the enums they handle, while `instantiate` only has to deal with the struct it is passed.

Typically, `instantiate` and `execute` have the type `Result`, while `query` has `StdResult` due to the underlying Cosmos SDK `Querier`.
[Previous Complex State and Maps](#) [Next Query](#)