[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 0 - Getting Started](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Getting Started](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Blockchain 101](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Blockchain History](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Public and Managed Blockchains](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Consensus in Distributed Networks](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Cryptography](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Self-Assessment Quiz](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Go Introduction - First Steps](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Go Basics](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Go Interfaces](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Control Structures in Go](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Arrays and Slices in Go](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Standard Packages in Go](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Concurrency in Go](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Docker Introduction](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Introduction to the Interchain](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[The Interchain Ecosystem](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Getting ATOM and Staking It](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[A Blockchain App Architecture](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Accounts](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Transactions](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Messages](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Modules](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Protobuf](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Multistore and Keepers](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[BaseApp](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Queries](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Events](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Context](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Testing](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Relaying with IBC](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Interchain Security](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Bridges](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Migrations](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 1 Quiz](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 2 - First Steps](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[First Steps](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Setup Your Work Environment](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Run a Node, API, and CLI](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Ignite CLI](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Store Object](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Create Custom Messages](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Create and Save a Game Properly](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Add a Way to Make a Move](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Emit Game Information](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Record the Game Winner](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 2 Exercise](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[What is IBC?](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[IBC Token Transfer](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[IBC Tooling](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[What is CosmJS?](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Your First CosmJS Actions](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Compose Complex Transactions](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Learn to Integrate Keplr](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Put Your Games in Order](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Auto-Expiring Games](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Let Players Set a Wager](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Handle wager payments](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Integration tests](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Incentivize Players](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Help Find a Correct Move](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Understand IBC Denoms](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Go Relayer](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Hermes Relayer](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[CosmJS Advanced](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Create Custom Objects](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Create Custom Messages](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Get an External GUI](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Backend Script for Game Indexing](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[IBC Deep Dive](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[IBC Application Developer Introduction](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Create a Leaderboard Chain](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Run in Production](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Prepare the Software to Run](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Prepare a Validator and Keys](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Prepare Where the Node Starts](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Prepare and Do Migrations](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Simulate Production in Docker](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Tally Player Info After Production](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Simulate a Migration in Docker](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Final Exam](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[What's Next?](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

[Continue Your Interchain Journey](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html)

Docs Version Switcher

On this page

[Transfer packet flow](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#transfer-packet-flow)

[Sending a transfer packet](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#sending-a-transfer-packet)

[Receiving a transfer packet](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#receiving-a-transfer-packet)

[Acknowledging or timing out packets](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#acknowledging-or-timing-out-packets)

[#Copy link](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#ibc-token-transfer) **IBC Token Transfer**



Transferring tokens between chains is both a common requirement and a significant technical challenge when two chains are incompatible. A convenient solution for moving tokens between chains is essential.   
  
In this section, you will explore how a fungible token transfer can be done with IBC.

Having looked at IBC's transport, authentication, and ordering layer (IBC/TAO), you can now take a look at [ICS-20 (opens new window)↗](https://github.com/cosmos/ibc/blob/master/spec/app/ics-020-fungible-token-transfer/README.md). ICS-20 describes **fungible token transfers**.



Fungibility refers to an instance in which a token is interchangeable with other instances of that token or not. Fungible tokens can be exchanged and replaced.

There are many use cases involving token transfers on blockchains, like the tokenization of assets holding value or initial coin offerings (ICOs) to finance blockchain projects. IBC makes it possible to transfer tokens and other digital assets between (sovereign) chains, both fungible and non-fungible tokens. For example, fungible token transfers allow you to build applications relying on cross-chain payments and token exchanges. Therefore, IBC frees up great potential for cross-chain Decentralized Finance (DeFi) applications by offering a technically reliable cross-chain interoperability protocol that is compatible with digital assets on multiple networks.

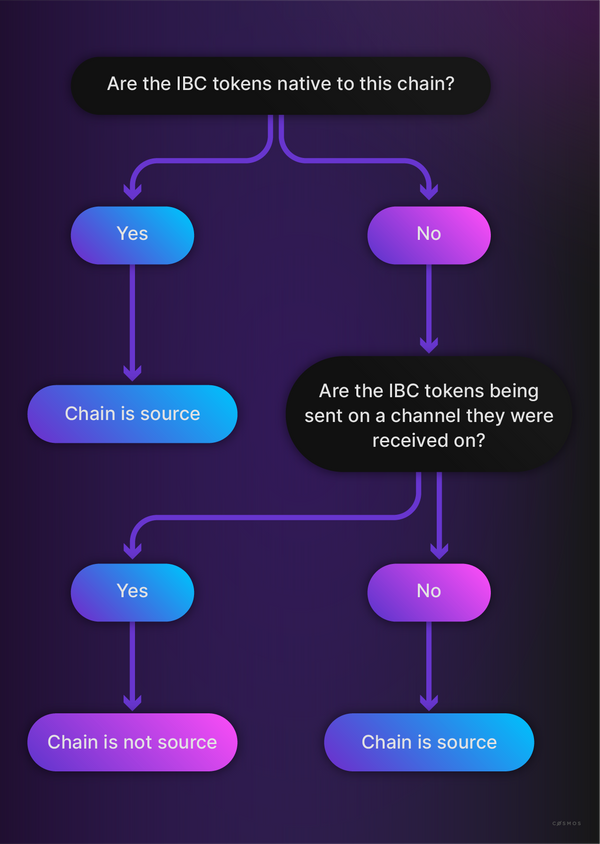
The corresponding [implementation (opens new window)↗](https://github.com/cosmos/ibc-go/tree/main/modules/apps/transfer) is a module on the application level.



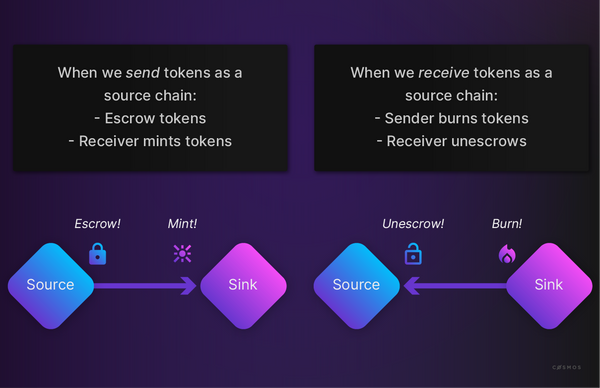
Look at the picture above. You can see two chains, A and B. You also see there is a channel connecting both chains.

How can tokens be transferred between chains and channels?

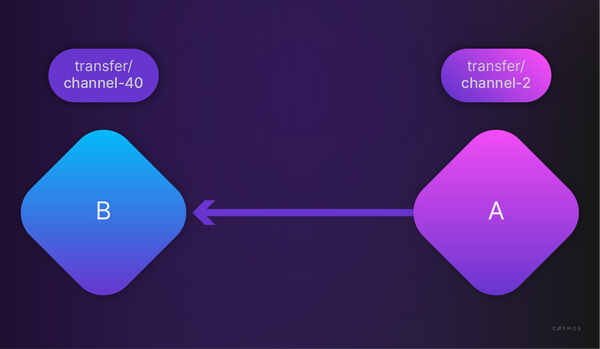
To understand the application logic for a token transfer, first, you have to determine the **source** chain:



Then the application logic can be summarized:



Shortly you will see the corresponding code. Now again have a look at a transfer from **source** to **sink**:

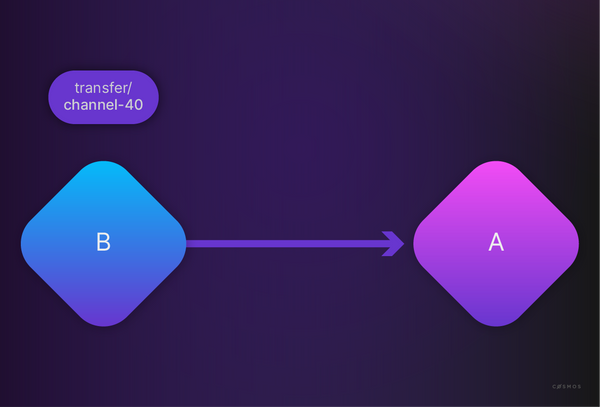


Above the **source** is chain A. The source channel is **channel-2** and the destination channel is **channel-40**. The token denominations are represented as {Port}/{Channel}/{denom} (or rather their [IBC denom representation](https://ida.interchain.io/tutorials/6-ibc-dev/) on chain). The prefixed port and channel pair indicate which channel the funds were previously sent through. You see **transfer/channel-...** because the transfer module will bind to a port, which is named transfer. If chain A sends 100 ATOM tokens, chain B will receive 100 ATOM tokens and append the destination prefix **port/channel-id**. So chain B will mint those 100 ATOM tokens as **ibc/<hash of transfer/channel-40/uatom>**. The **channel-id** will be increased sequentially per channel on a given connection.



We can send assets (or their IBC voucher representation) in multiple *hops* across multiple chains. Every single time the path will be prepended with the *port/channel-id/...* prefix.   
  
When sending this IBC denom (having had multiple hops) back to its source chain, for every hop back one *port/channel-id/...* prefix will be taken off. This results in a return to the original denom if all the hops are reversed.

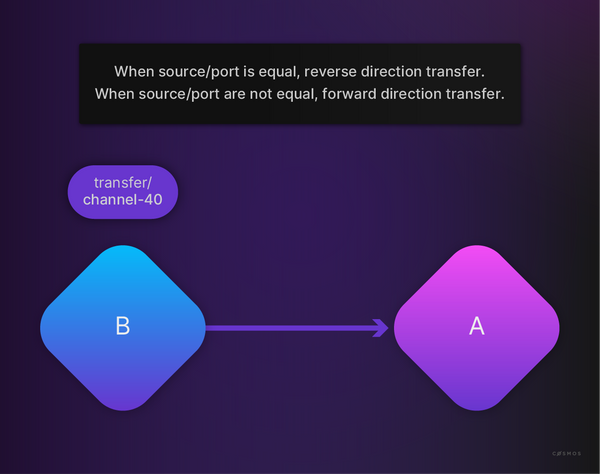
If the tokens are sent back from the **same channel** as they were received:



Chain A will "un-escrow" 100 **ATOM tokens**, thus, the prefix will be removed. Chain B will burn **transfer/channel-40/atoms**.



The prefix determines the **source** chain. If the module sends the token from another channel, chain B is the source chain and chain A mints new tokens with a prefix instead of un-escrowing ATOM tokens. You can have different channels between two chains, but you cannot transfer the same token across different channels back and forth. If {denom} contains /, then it must also follow the ICS-20 form, which indicates that this token has a multi-hop record. This requires that the character / is prohibited in non-IBC token denomination names.



You already know that an application needs to implement the [IBC Module Interface (opens new window)↗](https://github.com/cosmos/ibc-go/blob/v5.1.0/modules/core/05-port/types/module.go), so have a look at the [implementation for the token transfer (opens new window)↗](https://github.com/cosmos/ibc-go/blob/v5.1.0/modules/apps/transfer/ibc_module.go), e.g. for OnChanOpenInit:



Copy

// OnChanOpenInit implements the IBCModule interface

func (im IBCModule) OnChanOpenInit(

ctx sdk.Context,

order channeltypes.Order,

connectionHops []string,

portID string,

channelID string,

chanCap \*capabilitytypes.Capability,

counterparty channeltypes.Counterparty,

version string,

) error {

if err := ValidateTransferChannelParams(ctx, im.keeper, order, portID, channelID); err != nil {

return err

}

if version != types.Version {

return sdkerrors.Wrapf(types.ErrInvalidVersion, "got %s, expected %s", version, types.Version)

}

// Claim channel capability passed back by IBC module

if err := im.keeper.ClaimCapability(ctx, chanCap, host.ChannelCapabilityPath(portID, channelID)); err != nil {

return err

}

return nil

}

OnChanOpenAck, OnChanOpenConfirm, OnChanCloseInit, and OnChanCloseConfirm will do (almost) no checks.

[#Copy link](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#transfer-packet-flow) Transfer packet flow

You have seen an introduction to the application packet flow in [the section on channels](https://ida.interchain.io/academy/3-ibc/3-channels.html#application-packet-flow). This section will analyze this packet flow for the specific case of the *transfer* module.

[#Copy link](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#sending-a-transfer-packet) Sending a transfer packet

After a channel is established, the module can start sending and receiving packets.

So where does the module send a token? Take a look at the [msg\_server.go (opens new window)↗](https://github.com/cosmos/ibc-go/blob/v5.1.0/modules/apps/transfer/keeper/msg_server.go) of the token transfer module:



Copy

// Transfer defines an rpc handler method for MsgTransfer.

func (k Keeper) Transfer(goCtx context.Context, msg \*types.MsgTransfer) (\*types.MsgTransferResponse, error) {

...

if err := k.SendTransfer(

ctx, msg.SourcePort, msg.SourceChannel, msg.Token, sender, msg.Receiver, msg.TimeoutHeight, msg.TimeoutTimestamp,

); err != nil {

return nil, err

}

...

}

There you see SendTransfer, which implements the application logic after [checking if the sender is a source or sink chain (opens new window)↗](https://github.com/cosmos/ibc-go/blob/v5.1.0/modules/apps/transfer/types/coin.go):



Copy

func (k Keeper) SendTransfer(

ctx sdk.Context,

sourcePort,

sourceChannel string,

token sdk.Coin,

sender sdk.AccAddress,

receiver string,

timeoutHeight clienttypes.Height,

timeoutTimestamp uint64,

) {

...

// deconstruct the token denomination into the denomination trace info

// to determine if the sender is the source chain

if strings.HasPrefix(token.Denom, "ibc/") {

fullDenomPath, err = k.DenomPathFromHash(ctx, token.Denom)

if err != nil {

return err

}

}

...

// NOTE: SendTransfer simply sends the denomination as it exists on its own

// chain inside the packet data. The receiving chain will perform denom

// prefixing as necessary.

if types.SenderChainIsSource(sourcePort, sourceChannel, fullDenomPath) {

...

// create the escrow address for the tokens

escrowAddress := types.GetEscrowAddress(sourcePort, sourceChannel)

// escrow source tokens. It fails if balance insufficient.

if err := k.bankKeeper.SendCoins(...) {

} else {

...

if err := k.bankKeeper.SendCoinsFromAccountToModule(...);

...

if err := k.bankKeeper.BurnCoins(...);

...

}

packetData := types.NewFungibleTokenPacketData(

fullDenomPath, token.Amount.String(), sender.String(), receiver,

)

...

}

}

Expand



Take a look at the [type definition of a token packet (opens new window)↗](https://github.com/cosmos/ibc-go/blob/v5.1.0/proto/ibc/applications/transfer/v2/packet.proto) before diving further into the code:



Copy

syntax = "proto3";

package ibc.applications.transfer.v2;

option go\_package = "github.com/cosmos/ibc-go/v3/modules/apps/transfer/types";

// FungibleTokenPacketData defines a struct for the packet payload

// See FungibleTokenPacketData spec:

// https://github.com/cosmos/ibc/tree/master/spec/app/ics-020-fungible-token-transfer#data-structures

message FungibleTokenPacketData {

// the token denomination to be transferred

string denom = 1;

// the token amount to be transferred

string amount = 2;

// the sender address

string sender = 3;

// the recipient address on the destination chain

string receiver = 4;

// optional memo

string memo = 5;

}



An optional *memo* field was recently added to the packet definition. More details on the motivation, use cases, and consequences can be found in the [accompanying blog post (opens new window)↗](https://medium.com/the-interchain-foundation/moving-beyond-simple-token-transfers-d42b2b1dc29b).

[#Copy link](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#receiving-a-transfer-packet) Receiving a transfer packet

A relayer will then pick up the SendPacket event and submit a MsgRecvPacket on the destination chain.

This will trigger an OnRecvPacket callback that will decode a packet and apply the transfer token application logic:



Copy

// OnRecvPacket implements the IBCModule interface. A successful acknowledgement

// is returned if the packet data is successfully decoded and the receive application

// logic returns without error.

func (im IBCModule) OnRecvPacket(

ctx sdk.Context,

packet channeltypes.Packet,

relayer sdk.AccAddress,

) ibcexported.Acknowledgement {

ack := channeltypes.NewResultAcknowledgement([]byte{byte(1)})

var data types.FungibleTokenPacketData

var ackErr error

if err := types.ModuleCdc.UnmarshalJSON(packet.GetData(), &data); err != nil {

ackErr = sdkerrors.Wrapf(sdkerrors.ErrInvalidType, "cannot unmarshal ICS-20 transfer packet data")

ack = channeltypes.NewErrorAcknowledgement(ackErr)

}

// only attempt the application logic if the packet data

// was successfully decoded

if ack.Success() {

err := im.keeper.OnRecvPacket(ctx, packet, data)

if err != nil {

ack = channeltypes.NewErrorAcknowledgement(err)

ackErr = err

}

}

eventAttributes := []sdk.Attribute{

sdk.NewAttribute(sdk.AttributeKeyModule, types.ModuleName),

sdk.NewAttribute(sdk.AttributeKeySender, data.Sender),

sdk.NewAttribute(types.AttributeKeyReceiver, data.Receiver),

sdk.NewAttribute(types.AttributeKeyDenom, data.Denom),

sdk.NewAttribute(types.AttributeKeyAmount, data.Amount),

sdk.NewAttribute(types.AttributeKeyMemo, data.Memo),

sdk.NewAttribute(types.AttributeKeyAckSuccess, fmt.Sprintf("%t", ack.Success())),

}

if ackErr != nil {

eventAttributes = append(eventAttributes, sdk.NewAttribute(types.AttributeKeyAckError, ackErr.Error()))

}

ctx.EventManager().EmitEvent(

sdk.NewEvent(

types.EventTypePacket,

eventAttributes...,

),

)

// NOTE: acknowledgement will be written synchronously during IBC handler execution.

return ack

}

Expand





Observe in the previous example how we redirect to the module keeper's OnRecvPacket method and are constructing the acknowledgement to be sent back.

[#Copy link](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#acknowledging-or-timing-out-packets) Acknowledging or timing out packets

A useful exercise is to try to find and analyze the code corresponding to this. The place to start is the packet callbacks, usually defined in a file like module\_ibc.go or ibc\_module.go.



When a packet times out, the submission of a MsgTimeout is essential to get the locked funds unlocked again. As an exercise, try to find the code where this is executed.

synopsis

To summarize, this section has explored:

* How IBC provides a reliable solution to the technical challenge of transferring fungible and non-fungible tokens between two different blockchains, freeing up great potential for cross-chain Decentralized Finance (DeFi) applications.
* How the process for transferring value differs based on whether or not the IBC tokens are native to the source chain, or whether or not they are being sent on a channel they were previously received on.

previous

[](https://ida.interchain.io/academy/3-ibc/4-clients.html)

**[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/4-clients.html)**

up next

**[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/academy/3-ibc/8-ica.html)**

[[](https://ida.interchain.io/academy/3-ibc/8-ica.html)](https://ida.interchain.io/academy/3-ibc/8-ica.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

On this page

[Transfer packet flow](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#transfer-packet-flow)

[Sending a transfer packet](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#sending-a-transfer-packet)

[Receiving a transfer packet](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#receiving-a-transfer-packet)

[Acknowledging or timing out packets](https://ida.interchain.io/academy/3-ibc/7-token-transfer.html#acknowledging-or-timing-out-packets)

#### **Get Cosmos updates**

Unsubscribe at any time. [Privacy Policy↗](https://v1.cosmos.network/privacy)

     Next

Documentation

[Cosmos SDK](https://docs.cosmos.network/)[Cosmos Hub](https://hub.cosmos.network/)[CometBFT](https://docs.cometbft.com/)[IBC Protocol](https://ibc.cosmos.network/)

Community

[Interchain blog](https://blog.cosmos.network/)[Forum](https://forum.cosmos.network/)[Discord](https://discord.gg/cosmosnetwork)

Contributing

[Source code on GitHub](https://github.com/cosmos/sdk-tutorials)

[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)

**[](https://blog.cosmos.network/)[](https://twitter.com/cosmos)[](https://discord.gg/cosmosnetwork)[](https://www.linkedin.com/company/interchain-foundation/about/)[](https://reddit.com/r/cosmosnetwork)[](https://t.me/cosmosproject)[](https://www.youtube.com/c/CosmosProject)**



Dark mode

† This website is maintained by the Interchain Foundation (ICF). The contents and opinions of this website are those of the ICF. The ICF provides links to cryptocurrency exchanges as a service to the public. The ICF does not warrant that the information provided by these websites is correct, complete, and up-to-date. The ICF is not responsible for their content and expressly rejects any liability for damages of any kind resulting from the use, reference to, or reliance on any information contained within these websites.

Cosmos is a registered trademark of the [Interchain Foundation.](https://interchain.io/)[Privacy](https://v1.cosmos.network/privacy)