

# ProtocolConfigs

Constants that control how the chain operates.

These can only change during protocol upgrades which happen on epoch boundaries.

```
type ProtocolConfigs {  
  protocolVersion: UInt53!  
  featureFlags: [ProtocolConfigFeatureFlag!]!  
  configs: [ProtocolConfigAttr!]!  
  config(  
    key: String!  
  ): ProtocolConfigAttr  
  featureFlag(  
    key: String!  
  ): ProtocolConfigFeatureFlag  
}
```

## Fields

**ProtocolConfigs.protocolVersion** • **UInt53!** **non-null** **scalar**

The protocol is not required to change on every epoch boundary, so the protocol version tracks which change to the protocol these configs are from.

**ProtocolConfigs.featureFlags** • **[ProtocolConfigFeatureFlag!]!** **non-null** **object**

List all available feature flags and their values. Feature flags are a form of boolean configuration that are usually used to gate features while they are in development. Once a flag has been enabled, it is rare for it to be disabled.

**ProtocolConfigs.configs** • **[ProtocolConfigAttr!]!** **non-null** **object**

List all available configurations and their values. These configurations can take any value (but they will all be represented in string form), and do not include feature flags.

**ProtocolConfigs.config** • **ProtocolConfigAttr** **object**

Query for the value of the configuration with name **key**.

`ProtocolConfigs.config.key`. `String!` `non-null` `scalar`

`ProtocolConfigs.featureFlag`. `ProtocolConfigFeatureFlag`

`object`

Query for the state of the feature flag with name `key`.

`ProtocolConfigs.featureFlag.key`. `String!` `non-null` `scalar`

## Returned By

`protocolConfig` `query`

## Member Of

`Epoch` `object`

[✎ Edit this page](#)



© 2025 SUI FOUNDATION | DOCUMENTATION DISTRIBUTED UNDER CC BY 4.0