ServiceConfig

The enabled features and service limits configured by the server.

```
type ServiceConfig {
  isEnabled(
    feature: Feature!
  ): Boolean!
  enabledFeatures: [Feature!]!
  maxQueryDepth: Int!
  maxQueryNodes: Int!
  maxOutputNodes: Int!
  maxDbQueryCost: Int!
  defaultPageSize: Int!
  maxPageSize: Int!
  mutationTimeoutMs: Int!
  requestTimeoutMs: Int!
  maxTransactionPayloadSize: Int!
  maxQueryPayloadSize: Int!
  maxTypeArgumentDepth: Int!
  maxTypeArgumentWidth: Int!
  maxTypeNodes: Int!
  maxMoveValueDepth: Int!
  maxTransactionIds: Int!
  maxMultiGetObjectsKeys: Int!
  maxScanLimit: Int!
}
```

Fields

ServiceConfig.**isEnabled** . `Boolean!` **non-null** **scalar**

Check whether `feature` is enabled on this GraphQL service.

ServiceConfig.isEnabled.**feature** . **Feature!** **non-null** **enum**

ServiceConfig.**enabledFeatures** . `[Feature!]!` **non-null** **enum**

List of all features that are enabled on this GraphQL service.

ServiceConfig.**maxQueryDepth** . `Int!` **non-null** **scalar**

The maximum depth a GraphQL query can be to be accepted by this service.

`ServiceConfig.`**`maxQueryNodes`**`.``Int!` **non-null** **scalar**

The maximum number of nodes (field names) the service will accept in a single query.

`ServiceConfig.`**`maxOutputNodes`**`.``Int!` **non-null** **scalar**

The maximum number of output nodes in a GraphQL response.

Non-connection nodes have a count of 1, while connection nodes are counted as the specified 'first' or 'last' number of items, or the default_page_size as set by the server if those arguments are not set.

Counts accumulate multiplicatively down the query tree. For example, if a query starts with a connection of first: 10 and has a field to a connection with last: 20, the count at the second level would be 200 nodes. This is then summed to the count of 10 nodes at the first level, for a total of 210 nodes.

`ServiceConfig.`**`maxDbQueryCost`**`.``Int!` **non-null** **scalar**

Maximum estimated cost of a database query used to serve a GraphQL request. This is measured in the same units that the database uses in EXPLAIN queries.

`ServiceConfig.`**`defaultPageSize`**`.``Int!` **non-null** **scalar**

Default number of elements allowed on a single page of a connection.

`ServiceConfig.`**`maxPageSize`**`.``Int!` **non-null** **scalar**

Maximum number of elements allowed on a single page of a connection.

`ServiceConfig.`**`mutationTimeoutMs`**`.``Int!` **non-null** **scalar**

Maximum time in milliseconds spent waiting for a response from fullnode after issuing a a transaction to execute. Note that the transaction may still succeed even in the case of a timeout. Transactions are idempotent, so a transaction that times out should be resubmitted until the network returns a definite response (success or failure, not timeout).

`ServiceConfig.`**`requestTimeoutMs`**`.``Int!` **non-null** **scalar**

Maximum time in milliseconds that will be spent to serve one query request.

`ServiceConfig.`**`maxTransactionPayloadSize`**`.``Int!` **non-null** **scalar**

The maximum bytes allowed for the `txBytes` and `signatures` fields of the GraphQL mutation `executeTransactionBlock` node, or for the `txBytes` of a `dryRunTransactionBlock`.

It is the value of the maximum transaction bytes (including the signatures) allowed by the protocol, plus the Base64 overhead (roughly 1/3 of the original string).

`ServiceConfig.maxQueryPayloadSize`.`Int!` **non-null** **scalar**

The maximum bytes allowed for the JSON object in the request body of a GraphQL query, for the read part of the query. In case of mutations or dryRunTransactionBlocks the txBytes and signatures are not included in this limit.

`ServiceConfig.maxTypeArgumentDepth`.`Int!` **non-null** **scalar**

Maximum nesting allowed in type arguments in Move Types resolved by this service.

`ServiceConfig.maxTypeArgumentWidth`.`Int!` **non-null** **scalar**

Maximum number of type arguments passed into a generic instantiation of a Move Type resolved by this service.

`ServiceConfig.maxTypeNodes`.`Int!` **non-null** **scalar**

Maximum number of structs that need to be processed when calculating the layout of a single Move Type.

`ServiceConfig.maxMoveValueDepth`.`Int!` **non-null** **scalar**

Maximum nesting allowed in struct fields when calculating the layout of a single Move Type.

`ServiceConfig.maxTransactionIds`.`Int!` **non-null** **scalar**

Maximum number of transaction ids that can be passed to a `TransactionBlockFilter`.

`ServiceConfig.maxMultiGetObjectsKeys`.`Int!` **non-null** **scalar**

Maximum number of keys that can be passed to a `multiGetObjects` query.

`ServiceConfig.maxScanLimit`.`Int!` **non-null** **scalar**

Maximum number of candidates to scan when gathering a page of results.

Returned By

`serviceConfig` **query**