RmqUrl

August 18, 2026

RmqUrl

Kind: Interface

Source: `packages/microservices/external/rmq-url.interface.ts`

Part of: Microservices

`RmqUrl` defines the connection components required to build or represent a RabbitMQ broker URL. It captures broker location, authentication, virtual host, and AMQP connection tuning values used by the microservices transport layer.

Properties

Property	Type
`protocol`	`string`
`hostname`	`string`
`port`	`number`
`username`	`string`
`password`	`string`
`locale`	`string`
`frameMax`	`number`
`heartbeat`	`number`
`vhost`	`string`

Diagram

```
graph LR
  RmqUrl["RmqUrl"]
  RmqUrl --> Protocol["protocol: string"]
  RmqUrl --> Hostname["hostname: string"]
```

```
  RmqUrl --> Port["port: number"]
  RmqUrl --> Credentials["username / password"]
  RmqUrl --> VHost["vhost: string"]
  RmqUrl --> Locale["locale: string"]
  RmqUrl --> Tuning["frameMax / heartbeat"]

  Credentials --> Connection["RabbitMQ Connection"]
  VHost --> Connection
  Tuning --> Connection
  Protocol --> Connection
  Hostname --> Connection
  Port --> Connection
```

Usage

```ts
import type { RmqUrl } from './rmq-url.interface';

const rabbitMqConfig: RmqUrl = {
  protocol: 'amqp',
  hostname: 'localhost',
  port: 5672,
  username: 'guest',
  password: 'guest',
  locale: 'en_US',
  frameMax: 0,
  heartbeat: 60,
  vhost: '/',
};

const connectionUrl =
  `${rabbitMqConfig.protocol}://` +
  `${encodeURIComponent(rabbitMqConfig.username)}:` +
  `${encodeURIComponent(rabbitMqConfig.password)}@` +
  `${rabbitMqConfig.hostname}:${rabbitMqConfig.port}` +
  `${rabbitMqConfig.vhost}`;
```

AI Coding Instructions

- Keep all fields populated when constructing an `RmqUrl`; this interface does not mark connection settings as optional.
- Use `amqp` or `amqps` for `protocol` depending on whether the RabbitMQ broker requires TLS.
- Encode `username` and `password` before interpolating them into a connection URI, especially when they contain reserved URL characters.

- Preserve `vhost`, `heartbeat`, and `frameMax` values when passing configuration into RabbitMQ transport or connection setup code.
- Avoid hardcoding credentials in source files; populate `RmqUrl` from validated environment or configuration values.

How it works

- `RmqUrl` is an exported, public TypeScript interface for the object form of an RMQ connection URL. It declares only optional properties and has no methods or runtime implementation. [packages/microservices/external/rmq-url.interface.ts:4-17]
- Its optional fields are:
  - `protocol`, `hostname`, `username`, `password`, `locale`, and `vhost` as strings. [packages/microservices/external/rmq-url.interface.ts:8-13,16]
  - `port`, `frameMax`, and `heartbeat` as numbers. [packages/microservices/external/rmq-url.interface.ts:10,14-15]
- `RmqOptions.options.urls` accepts either an array of URL strings or an array of `RmqUrl` objects; the adjacent documentation says these are connection URLs tried in order. [packages/microservices/interfaces/microservice-configuration.interface.ts:222-228]
- `ServerRMQ` stores this setting as `string[] | RmqUrl[]`. Its constructor reads `options.urls` and falls back to an array containing the RMQ default URL when the setting is falsy. [packages/microservices/server/server-rmq.ts:67,77-85]
- When creating the RMQ client, `ServerRMQ` passes the stored URL array directly as the first argument to `amqp-connection-manager`'s `connect` call. [packages/microservices/server/server-rmq.ts:169-176]
- No validation, normalization, field-level parsing, errors, or side effects are declared by `RmqUrl` itself. In the visible `ServerRMQ` path, individual `RmqUrl` fields are not inspected before the array is passed to `connect`. [packages/microservices/external/rmq-url.interface.ts:7-17] [packages/microservices/server/server-rmq.ts:169-176]