AmqplibQueueOptions

August 18, 2026

AmqplibQueueOptions

Kind: Interface

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

Part of: Microservices

`AmqplibQueueOptions` defines RabbitMQ queue configuration options used when declaring or managing queues through the AMQP integration. It covers queue lifecycle behavior, message retention, dead-letter routing, capacity limits, and priority support.

Properties

Property	Type
`durable`	`boolean`
`autoDelete`	`boolean`
`arguments`	`any`
`messageTtl`	`number`
`expires`	`number`
`deadLetterExchange`	`string`
`deadLetterRoutingKey`	`string`
`maxLength`	`number`
`maxPriority`	`number`

Diagram

```
graph LR
  A[AmqplibQueueOptions] --> B[Queue Lifecycle]
  A --> C[Message Retention]
  A --> D[Dead-Letter Routing]
```

```
  A --> E[Capacity and Priority]

  B --> B1[durable]
  B --> B2[autoDelete]
  B --> B3[expires]

  C --> C1[messageTtl]

  D --> D1[deadLetterExchange]
  D --> D2[deadLetterRoutingKey]

  E --> E1[maxLength]
  E --> E2[maxPriority]
  A --> F[arguments]
```

Usage

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

const queueOptions: AmqplibQueueOptions = {
  durable: true,
  autoDelete: false,
  arguments: {
    'x-queue-type': 'classic',
  },
  messageTtl: 60_000,
  expires: 86_400_000,
  deadLetterExchange: 'orders.dlx',
  deadLetterRoutingKey: 'orders.failed',
  maxLength: 10_000,
  maxPriority: 10,
};

// Map these options to the AMQP queue declaration options as needed.
await channel.assertQueue('orders.created', {
  durable: queueOptions.durable,
  autoDelete: queueOptions.autoDelete,
  arguments: {
    ...queueOptions.arguments,
    'x-message-ttl': queueOptions.messageTtl,
    'x-expires': queueOptions.expires,
    'x-dead-letter-exchange': queueOptions.deadLetterExchange,
    'x-dead-letter-routing-key': queueOptions.deadLetterRoutingKey,
    'x-max-length': queueOptions.maxLength,
    'x-max-priority': queueOptions.maxPriority,
  },
});
```

AI Coding Instructions

- Preserve RabbitMQ argument names such as `x-message-ttl`, `x-dead-letter-exchange`, and `x-max-priority` when mapping these fields to AMQP options.
- Treat `messageTtl` and `expires` as millisecond durations; avoid passing values intended to be seconds.
- Configure `deadLetterExchange` and `deadLetterRoutingKey` together, and ensure the referenced dead-letter exchange exists.
- Use `durable: true` for queues that must survive broker restarts, and avoid `autoDelete: true` for long-lived production queues.
- Keep custom broker-specific configuration in `arguments` without overwriting explicitly mapped queue settings.