

RabbitMQ: Getting started

All you need to know about RabbitMQ

Welcome on Stackhero's documentation!

Stackhero provides RabbitMQ instances that are ready for production in just 2 minutes!

Including TLS encryption (aka HTTPS), customizable domain name, full access to WEB UI, backups and updates in just a click.

Try our [managed RabbitMQ cloud](#) in just 2 minutes

- [What is RabbitMQ?](#)
- [Examples of RabbitMQ usages](#)
 - [Example of RabbitMQ usage in a ride-sharing or a food delivery app](#)
- [When to use RabbitMQ?](#)

What is RabbitMQ?

RabbitMQ is a "message broker" server.

It provides a way to exchange messages between different services, in an async or sync way.

The goal is to let your app delegate actions, in case they are time consuming or if you want to parallelize them.

Using such a message broker will increase your app speed, provide an easy way to scale and improve you app realibility.

RabbitMQ will help you to have a micro services app, without the bad side of such architectures.

The ecosystem around RabbitMQ is super rich and a lot of plugins exist to extend its fonctionnalities.

Also, multiple client libraries exist so you can connect to RabbitMQ from any language: Java, Spring, .NET, Ruby, Python, PHP, Node.js, Go, Rust and more.

Examples of RabbitMQ usages

Imagine you have an app which let a user downloading every past year invoices in just one action.

This app will generate multiple invoices as PDF files in one shot, which is really time consuming.

Using RabbitMQ, your app will send 12 messages with invoices informations to a `topic`.

Another system, dedicated to invoices generation, will be triggered by these messages. It will generate the PDF files and then will send back to RabbitMQ the files.

Your app will then receive the PDF files and send them back to your client.

Where this system becomes really powerful, is that you can run multiple scripts that will generate invoices in parallel!

With the same code, you will be able to generate these 12 invoices in a row!

Now imagine that you have 100 users, asking for the latest 12 invoices at the same time. No problem, your system is now able to scale and generate any amount of PDF files in parallel!

Now, imagine that you want to send the invoices per email.

Just create a script that takes an email address and a PDF file in input. Make it sends the PDF file to the email address.

Plug this email script to RabbitMQ.

Then, tell your invoice generator to send the PDF file to the `topic "emails"`.

That's it. Your app can now generate dozen of invoices and send them by email, in a scalable way.

Another benefit is the increase of reliability.

Imagine that for some reason, one of your email script crashes sometimes.

The email script will take the invoice file from RabbitMQ, try to send the email and crashes.

RabbitMQ will handle that crash and the invoice will not be loose! Another email script will take the file and send it, without your user noticing!

RabbitMQ increases reliability and consistency at the same time.

There is a ton of different use cases. Basically, if you have multiple systems or

apps that need to communicate together in a reliable way, or if you have some time consuming tasks that you want to delegate to other systems, RabbitMQ is probably the solution you need.

Example of RabbitMQ usage in a ride-sharing or a food delivery app

In an app like Uber or Lyft, when a user asks for a ride, multiple informations are displayed:

- The price of the ride
- The duration of the ride
- The time at which the use can be picked up by a driver
- The itinerary

This 4 parts can be split in 4 services and communicate through to RabbitMQ. When a user asks for a ride, its start position and destination can be send to the 4 services that will compute the result in parallel.

You can then display as soon as they are resolved to the user.

If the duration of the ride is long to be calculated, the 3 others informations will be displayed to the user, ensuring you are not loosing your customer because of one slow calculation.

In an app like GurbHub or Deliveroo, when the user is selecting his food, multiple informations are displayed:

- The availability of the ordered food
- The waiting time for the food to be delivered
- The price of the cart

As in the Uber/Lyft example, these 3 parts can be delegated to 3 services too, using RabbitMQ to exchanges messages between them.

When to use RabbitMQ?

RabbitMQ is perfect if you are in one of this use-case:

- you have some time consuming tasks
- you need to scale your app
- you want to increase reliability of your systems

If your app sends emails, generate PDF files, do long computations or if you want to split your code to execute multiple tasks in parallel, RabbitMQ is probably the system you need.

Other articles about RabbitMQ that might interest you

- [Configuration and examples](#)
How to start using RabbitMQ
- [Using with PHP](#)
How to connect to RabbitMQ with PHP and php-amqplib