

Open Source Metering and Usage Based Billing

www.getlago.com

AGPL-3.0 license

1.1k stars 40 forks


☆ Star

▼

🔔 Notifications

- <> Code
- 🕒 Issues 31
- 🔗 Pull requests 1
- ▶ Actions
- 📁 Projects 1
- 📖 Wiki
- 🛡 Security
- 📈 Insights


🔗 main ▼

 jdenquin misc(docker): fix letsencrypt config (#139) ...

4 days ago ⌚ 196

[View code](#)

☰ README.md

 **Lago**

Open source metering & usage-based billing

📧 Slack

📈 Billable metrics

📅 Plans

⚙ Add-ons

🎫 Coupons

👤 Customers

Billable metrics

Metric name
<div>📈 Active Users</div> <div>active_users</div>
<div>📈 Max Storage</div> <div>storage</div>

Lago

The best alternative to Chargebee, Recurly or Stripe Billing.
For usage-based, subscription-based, and all the nuances of pricing in between.

[Slack](#) · [Website](#) · [Issues](#) · [Roadmap](#)

Lago Slack Community [lago.slack.com](#)

stars **1.1k**

Hacker News **777**

license **AGPLv3**

follow [@getlago](#) 241

Backed by [Y Combinator](#)

The programmable API for usage-based billing

🔍 _lago-product.mov ▾

0:00

The problem: Billing systems are still a nightmare for engineers



- **Building our billing system internally**



- **Working with dates**
- **Pro-ratas**
- **Upgrades & Downgrades**
- **Pay in advance or in arrears**
- **Dunning & Retries**
- **Usage based**
- **Taxes & Currencies**

Engineers be like...

Read more first-hand experiences from Qonto, Algolia, Pleo, Segment, or the 350+. Hackernews comments [here](#).

The Solution: Lago, the open-source billing API for product-led SaaS

- Event-based: if you can track it, you can charge for it;
- Built for product-led growth companies;
- Hybrid pricing: subscription and usage;
- Hybrid go-to-market motion: self-serve and sales-led.

Open-source, open architecture:

- Composable: connect Lago to any of your internal systems or tools (i.e. any payment gateway, CRM, CPQ, accounting software);
- Pricing: we're not rent seekers, we're not asking for a % of your revenue. Our self-hosted version is free. Our cloud version is priced like a SaaS;
- Privacy: your data never has to leave your infrastructure.

Features

- **Usage metering**: Lago's event-based architecture provides a solid foundation for building a fair pricing model that scales with your business.
- **Price plans**: Lago supports all pricing models. Create pay-as-you-go and hybrid plans in no time with our intuitive user interface or API.
- **Coupons**: Create engaging marketing campaigns and increase conversion with coupons that customers can redeem to get a discount.
- **Add-ons**: Why wait until the end of the billing cycle to get paid? Lago allows you to create one-time charges that are invoiced on the fly.
- **Invoicing**: Depending on the configuration of your plans, Lago automatically calculates what each customer owes you and generates invoices.
- **Prepaid credits**: Unlock recurring revenue opportunities for pay-as-you-go pricing models with Lago's prepaid credit features.

Stay up to date

Lago launched its v0.1 on June 2nd, 2022. Lots of new features are coming, and are generally released on a bi-weekly basis. Watch updates of this repository to be notified of future updates. Take a look at our [Public Roadmap](#)!

License

Distributed under the AGPLv3 License. Read more [here](#).

Deploy locally

Requirements

1. Install Docker on your machine;
2. Make sure Docker Compose is installed and available (it should be the case if you have chosen to install Docker via Docker Desktop); and

3. Make sure Git is installed on your machine.

Run the app

To start using Lago, run the following commands in a shell:

```
# Get the code
git clone https://github.com/getlago/lago.git

# Go to Lago folder
cd lago

# Set up environment configuration
echo "LAGO_RSA_PRIVATE_KEY=\"`openssl genrsa 2048 | base64`\"" >> .env
source .env

# Start
docker-compose up
```

You can now open your browser and go to <http://localhost> to connect to the application. Lago's API is exposed at <http://localhost:3000>.

Find your API key

Your API Key can be found directly in the UI:

1. Access the **Developer** section from the sidebar;
2. The first tab of this section is related to your **API keys**; and
3. Click the **Copy** button to copy it to clipboard.

Version, environment variables and components

Docker images are always updated to the last stable version in the docker-compose.yml file. You can use a different tag if needed by checking the releases list.

Lago uses the following environment variables to configure the components of the application. You can override them to customise your setup. Take a closer look at our [documentation](#).

Use the fully hosted Lago Cloud

Get started with our free Lago Cloud version, request access: hello@getlago.com.

Getting the most out of Lago

- See the [documentation](#) to learn more about all the features;
- Use our [templates](#) to get inspiration and learn how to reproduce Algolia's, Segment's and Klaviyo's pricing models;
- Join our [Slack community](#) if you need help, or want to chat, we're here to help;
- Contribute on GitHub: read our [guidelines](#);

- Follow us on [Twitter](#) for the latest news;
- You can email us as well: hello@getlago.com.



Contributing & Development Environment

If you want to have a nice Lago Development Environment, you can follow this [guide](#) on how to make it works on your machine. This guide is intended for people willing to contribute to Lago. If you want to try Lago on your local system, we encourage your to look at the Lago public documentation.

You can contribute by following our [guidelines](#).



Philosophy

B2B SaaS has evolved, but billing has not yet.

1- We're not in the “subscription economy” anymore. And we won't go “full usage-based pricing” quite yet

Pricings are now mostly hybrid: they include a usage-based component (i.e. “if you use more you pay more”) and a subscription component (i.e. a recurring fee for basic usage).

Not all software companies will go full “usage-based” like Snowflake for instance. This model is the new standard for cloud infrastructure products. However, in other areas of SaaS, users want to know beforehand how much they will pay to control their spending and software companies want to be able to predict recurring revenues.

2- Go-to-market is not either bottom-up or top-down anymore

SaaS used to be either self-service (SMBs) or sales-led (Enterprises). Go-to-market now mixes the self-service (all customers access the same price plans) and sales-led (customers get a custom quote from a sales representative) motions. A typical journey involves an individual contributor in a company who tests a new tool, puts their corporate credit card in, and starts spreading the use of the tool within the organization. At that point, the VP or head of department might want to upgrade to a custom plan tailored to the needs of the whole organization. As a result, billing needs to be flexible, automated, and transparent enough to embrace this hybrid go-to-market motion as well.

3- The “rent seeker” pricing of current billing solutions needs to stop

Why do payment companies take a cut on revenues? Because the higher the amount, the higher the risk for them (e.g. fraud, disputes, etc.).

Why did billing companies adopt the same pricing structure? We're not able to provide an answer that makes sense. It's been said on the internet that they did this because they could (read more [here](#)).

One last thing...

Lago is agnostic and we aim at being as transparent as possible, so we won't nudge or lock you into using a specific tool in exchange for using our billing API ([learn more](#)).

Releases

 27 tags

Contributors 9



Languages

 Shell 100.0%