

Backend Software Engineer: Criteria

Raileasy are looking for a backend developer, ideally with a .NET/C# background, to work with us on our retail API microservices. Our API powers some of the most established and recognisable brands in the rail retail space (particularly where split-ticketing is involved) such as TrainSplit, SplitMyFare, Railboard and others. We provide B2C and B2B services.

You will work collaboratively with the rest of our small engineering team which comprises a mix of frontend and backend developers, app engineers, an infrastructure/security engineer and our team lead. We offer a competitive salary, 33 days paid annual leave, a leisure rail travel allowance and other benefits.

Essential

- Professional experience with writing C#, Java or another OOP language
- Experience working with relational databases and ideally an ORM of some description (Entity Framework is what we primarily use, as well as GORM)
- Ability to become familiar with an existing codebase (although support with this will be provided), troubleshoot issues in code you may not be familiar with and reduce technical debt as appropriate
- Familiarity with writing technical documentation and automated tests (XUnit/NUnit)
- Willingness to work remotely with occasional travel to London, Leamington Spa or other mutually convenient locations (for e.g. hack days to collaborate in-person)
- Ability to take some responsibility for the running of the API microservices; responding to escalated technical issues, alongside other team members
- Experience with Google Cloud or another cloud provider (AWS, Azure, Oracle)
- Understanding of Git and Docker

Desirable

- An interest in public transport/rail
- Experience with Golang or other low-level languages suited for smaller microservices
- Experience maintaining CI(/CD) pipeline configurations (e.g. CircleCI, Travis, GitLab, Bitbucket Pipelines, GitHub Actions etc)
- Experience with Kubernetes*

Raileasy's engineering culture:

- Open/transparent, with the ability to shape company direction & have your voice heard
- We'll give you autonomy and respect your skillset
- Small, close-knit engineering team with a willingness to wear different hats, collaborate and support each other
- Flexible working hours

* We will organise training if necessary

Backend Software Engineer: Q&A

What can you tell me about the API's architecture?

Our API is mostly .NET 6 LTS (with a minority of services currently on .NET Core 3.1) and comprises a number of containerised microservices. It runs in Kubernetes on GCP (using GKE). We use Traefik as our ingress controller. Some newer microservices are written in Golang.

Clients communicate with the API using JSON (or XML) HTTP requests. The API is documented using OpenAPI and the Swagger UI is exposed to API clients.

What does the API do to book tickets?

When a client (our website, our Android/iOS app, external clients) performs a journey search, we communicate with the journey planning service to retrieve itineraries and fares. We allow the clients to add their journeys into a basket, request seat reservations using the fulfilment service and the system then works out the available delivery methods. When the customer pays (primarily using Braintree), we communicate with our fulfilment provider to generate a paper ticket collection reference (for Ticket on Departure) or to generate E-Ticket PDFs which we then send out via email.

How is the API tested?

We have a selection of unit, integration and regression tests which cover domain logic as well as ensuring the components of our solution work well together. As you'd expect, we have CI/CD pipelines in place for these tests and deployments.

Our in-house API clients also benefit from end-to-end (Cypress/Playwright/Espresso/XCUI/Test) tests which indirectly test the API in the staging environment.

How would I be working in this role?

We regularly hold prioritisation meetings to collaboratively determine where development effort should be concentrated. These meetings would be with technical staff, informed by a steer from the business.

As part of our usual working week, we hold a brief synchronous call within the engineering team on each Monday morning where we talk about our planned work for the week and what we achieved in the previous week. Sometimes what you plan to do in a week doesn't work out (perhaps your work is blocked by an external supplier or you get stuck on something, or something else takes priority) but that's not a problem. We take meeting minutes via a Google Doc and share these with the wider business.

We collaborate both virtually (by jumping on calls and pair-programming when it makes sense to do so) and occasionally in-person by arranging collaborative “hack days” at a mutually convenient location. Travel would of course be covered. These are a chance to better get to know the team and hopefully work on a minimum viable prototype of something cool that our customers will benefit from. Previous hack days have allowed us to ship expense receipts functionality and improved brand authentication for our web retail solution.

What if I’m unfamiliar with rail data / journey planning / fares?

We’re happy to give full training on rail domain matters, particularly given that much of the documentation relating to the rail industry systems we use is not in the public domain. Some knowledge of fares and ticketing would be useful, but we are happy to explain everything you’d need to know in this area.