

# RESTHEART



## Ready to use Backend for the Modern Web.

Declarative Security, SDK and Instant REST, GraphQL and WebSocket Data APIs for MongoDB.

designed for **speed** and **simplicity**



to cut time and costs  
and make **development pleasant**



downloaded > 1 million times  
by startups and enterprises

Made with  by SoftInstigate

<https://restheart.org>

[ask@restheart.org](mailto:ask@restheart.org)





## Fast development

RESTHeart is a Runtime for microservices that provides developers with 80% of the most common features and a modern framework to develop the remaining 20%

- Declarative Authentication and Authorization
- Instant REST, GraphQL and WebSocket API for MongoDB
- GraalVM powered Polyglot Framework to develop Services and Interceptors
- Fully documented at <https://restheart.org/docs>



## Deploy at rest

RESTHeart is tailored for the JVM, GraalVM, Docker or Kubernetes, crafted from best of breed libraries and standards, designed to radically simplify microservices development and deployment.

- Ready-to-run Runtime
- Available as a standalone JAR file, native binary or Docker image
- Deploy it on Cloud and On-Premises



## Dual licensed

RESTHeart is dual-licensed under the AGPL and a Business Friendly Enterprise License

- Enjoy the free AGPL distribution without feature restrictions
- Purchase the Enterprise License for production-grade support and to use RESTHeart in closed source products or services

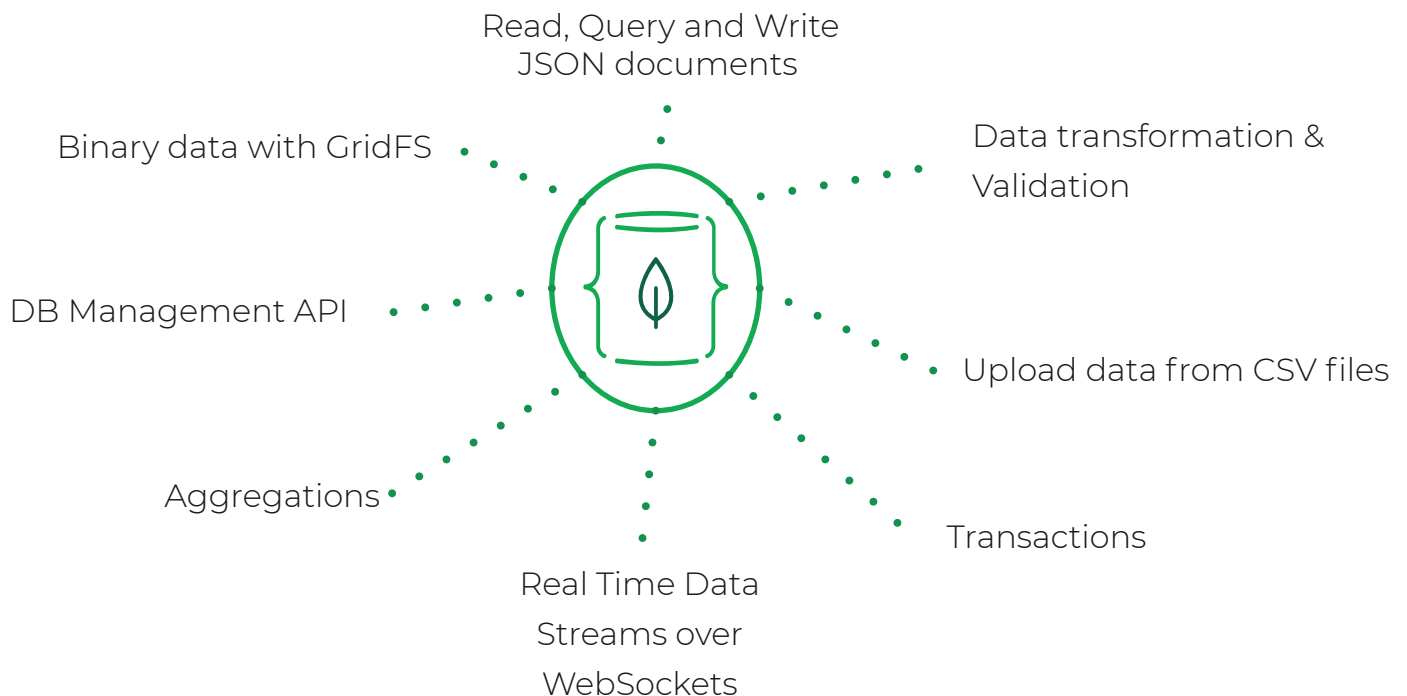




## The perfect MongoDB's companion

RESTHeart automatically connects to **MongoDB** (or AWS DocumentDB and Azure CosmosDB)

- It exposes **full** database's capabilities via **REST**, **GraphQL** and **WebSocket** APIs
- Developers don't need to write a single line of code to serve JSON data and binary content to Mobile Apps, Web Apps or Integration Middleware
- The Instant Data API **cuts development complexity and costs** up to 90%



Example: query data with JavaScript from the Browser

[Click me](#) to run the example in your browser

```
const url =
  encodeURIComponent('https://beta.mrest.io/demo/messages?
  filter={"from":"Bob"}');

fetch(url, { headers: { 'key': 'demo' } })
  .then(response => response.json())
  .then(json => JSON.stringify(json, null, 2))
  .then(data => console.log(data));
```

```
[
  {
    "_id": { "$oid": "5db0067615f384eda89e5fab"},
    "message": "hi there!",
    "from": "Bob",
    "timestamp": { "$date": 1571817078280 }
  }
]
```





## Modular and extensible

**RESTHeart** is modular and can be easily customized.

- The Core Runtime is the foundation
- Each functionality, e.g. the GraphQL API, is implemented in its own module
- Manage modules by adding or removing them from the “plugin” directory

### Security

Authenticate clients via multiple schemas and role-based authorization policies, via simple configuration.

The security layer can integrate external providers and implement fine-grained security policies.

### Data API

RESTHeart leverages MongoDB to provide all data functionalities required by modern applications.

Supports all the features of MongoDB including queries, aggregations, change streams and transactions!

## Core Runtime

RESTHeart Core Runtime deploys and runs modules in a lightweight, multi-threading, secure execution context.

It is built for stability, coded on solid Java foundations and covered by hundreds of tests.

### Dev Framework

Develop *Web Services* and *Interceptors* to execute logic on different stages of the request/response life-cycle.

Supports Java, Kotlin, JavaScript, TypeScript and leverages GraalVM language interoperability (e.g. call Java core methods from JavaScript).

### API gateway

Use RESTHeart as the *Ingress* for your microservices to handle them under the same security context.

This makes possible proxying external microservices, made with Node.js, Spring Boot, AWS Lambda, etc





## Extreme Performances

### Hundreds of thousands TPS

RESTHeart parallel architecture provides superior performances.

More at <https://restheart.org/docs/performances>

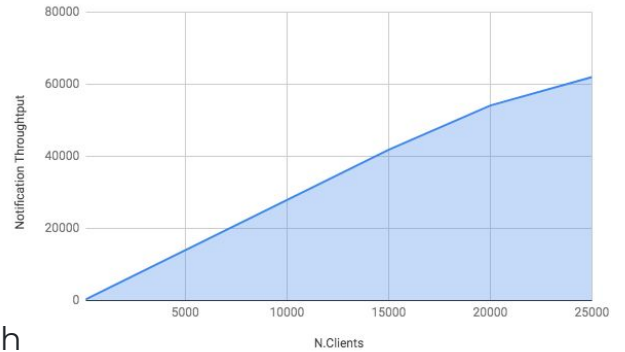
### Horizontal scaling

RESTHeart is fully stateless and allows clustering, to reach demanding requirements (only supported with the Enterprise License)

More at <https://restheart.org/docs/clustering>

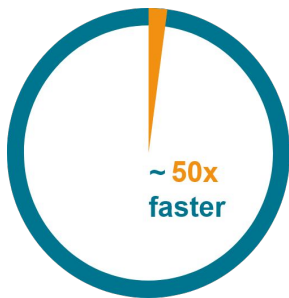
### Even faster on GraalVM

RESTHeart for GraalVM (supported with the Enterprise License) provides a native solution with instant startup time and smaller memory footprint. This is perfect, for example, when deploying to Kubernetes clusters, where regular Java applications usually consume too many resources.

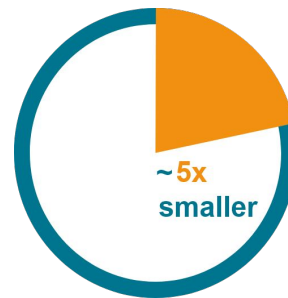


The chart shows nearly-realtime data streams in a test scenario with 25.000 concurrent clients

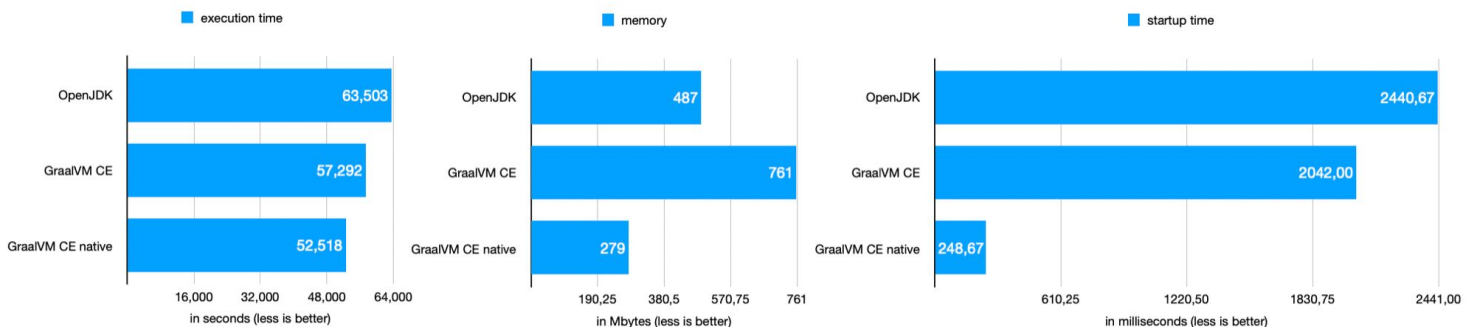
### Startup time

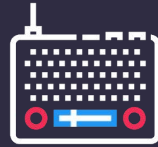


### Memory Footprint



The following table compares RESTHeart on OpenJDK, GraalVM and as native image





Runtime for Microservices



Contact us

[ask@restheart.org](mailto:ask@restheart.org)




Get more information

<https://restheart.org>

Fork RESTHeart on Github

<https://github.com/SoftInstigate/restheart>

Made with  by SoftInstigate  
<https://restheart.org>  
[ask@restheart.org](mailto:ask@restheart.org)

