

# {plumberizer} - Production-Grade Apps with Automatically Generated API Backends

*Anónimo*

**Abstract** The Shiny framework has significantly enhanced R's ability to interface R projects with the world wide web. Shiny is exceptionally useful for initial prototypes, minimum viable products, and even production-grade real SaaS startup projects. However, as applications grow and require greater scalability, Shiny's monolithic structure may become a limitation. To achieve better scalability, modern web design patterns recommend adopting a frontend-backend architecture, especially when cost-effectiveness is a concern. In R, this can be achieved by decoupling Shiny's backend functions using the {plumber} framework.

In this presentation, we introduce {plumberizer}, an innovative R package designed to streamline the creation of RESTful APIs for R packages. By executing a single function, `plumberize(package_name)`, {plumberizer} automatically generates the `plumber.R` file, establishing a distinct API endpoint for each exported function in an R package. This separation of concerns facilitates the development of scalable, maintainable, and high-performance R-based web applications.

We will showcase {plumberizer} in action and discuss the limitations of Shiny's monolithic structure, particularly in terms of scalability and cost. By adopting a frontend-backend architecture with Plumber APIs, we will show how these issues can be effectively resolved. This approach enables developers to create R applications that are not only robust but also optimized for large-scale production environments.

For further information on the {plumberizer} package, please visit <https://github.com/jcrodriguez1989/plumberizer>.

**Palabras clave:** Frontend-Backend Architecture - Scalability - Shiny Limitations - Cost-Effectiveness