Master Thesis Summary: *Development, Test and Application of a framework for cloud serverless services*

Andrea Santu [s251579]

November 21, 2020

1 Introduction

The overview of services for the creation of web applications is focusing more and more towards a micro services oriented approach, moving away from monolithic structures. The maximum representation of this is with the serverless paradigm, which has found an implementation in the cloud model Function as a Service, a model that uses plain simple functions as its main resources. Serverless Framework has emerged as one of the major framework that allows the usage of the homonym paradigm in a simple way. Despite the functionalities introduced by Serverless, the developer must take charge of various operations concerning indirectly the business logic of the application. The Restlessness framework was born with the goal of improving the user experience of Serverless, providing a standard project and testing structure, a Command Line Interface and a local Web Interface through which is possible to completely manage the project, and with the further goal of minimizing all operations that do not concern directly the application's business logic.

2 Development

Part of the development has been spent on creating a productive development workflow, with tools for Continuous Integration and Continuous Delivery, provided by the CircleCi platform, and a version control system, provided by Github. Its development has then been focused on the main components *@restlessness/cli* and *@restlessness/core*, both available on Npm, the Node.js package manager. Users interact with the framework mainly using *@restlessness/cli*, which provides commands for: creating a project, developing the project locally, and deploying it on the cloud provider platform. The Cli package depends on the *@restlessness/core* package to provide its functionalities, mainly resources creation and management. The core resources that can be managed are: Endpoints and Schedules, which are functions executed in response to http and periodic or programmed events respectively; Authorizers, functions that perform authorization operation, granting or denying access to functions or other resources. The framework provides a system to extend its functionalities through addon packages. It has been developed addon packages for common patterns, such as database access and authentication.

3 Application

During the framework development, it has been possible to test it on real applications, thus allowing to find and correct critical issues. In particular the main improvements have been made during the development of the project *Spazio alla Scuola*, with the main ones being: Cold start handling, use of the non relational database mongodb, and limitations on the applications structure proposed at the beginning of the framework development. Addressing those problems led the Restlessness framework to be production ready, proving the benefits of the serverless paradigm in conjunction with the Function as a Service model, in terms of scalability and ease of development.