Busy Bee

Glossary

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 22/03/2020 | 1.0 | Document inception | Carla-Maria Rusu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Glossary 4

Glossary

# Introduction

This document provides an alphabetical list of the terms, acronyms and abbreviations used throughout the span of project documents and offers a definition for each of them.

# Glossary

|  |  |  |  |
| --- | --- | --- | --- |
| **Term** | **Definition and Information** | **Format** | **Validation Rules** |
| Axios | Axios is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser that also supports the ES6 Promise API. | Axios  Axios.js |  |
| Domain Driven Design | Domain-driven design (DDD) is an approach to software development for complex needs by connecting the implementation to an evolving model. | DDD  Domain Driven Design |  |
| Entity | An object that is not defined by its attributes, but rather by a thread of continuity and its identity.  Example: Most airlines distinguish each seat uniquely on every flight. Each seat is an entity in this context. However, Southwest Airlines, EasyJet and Ryanair do not distinguish between every seat; all seats are the same. In this context, a seat is actually a value object. | Entity |  |
| Factory | Methods for creating domain objects should delegate to a specialized Factory object such that alternative implementations may be easily interchanged. | Factory |  |
| Maven Project | Maven is a tool that can be used for building and managing any Java-based project. | Maven  Maven Project |  |
| Microservices | Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are   * Highly maintainable and testable * Loosely coupled * Independently deployable * Organized around business capabilities * Owned by a small team   The microservice architecture enables the rapid, frequent and reliable delivery of large, complex applications. It also enables an organization to evolve its technology stack. | Micro Services  Microservices  Microservice architecture |  |
| MySQL Workbench | MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. | MySQL Workbench |  |
| Repository | Methods for retrieving domain objects should delegate to a specialized Repository object such that alternative storage implementations may be easily interchanged. | Repository  Repo |  |
| RESTful Api | A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data. | REST API  RESTful Api |  |
| Service | When an operation does not conceptually belong to any object. Following the natural contours of the problem, you can implement these operations in services. It is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development. | Service |  |
| Spring Boot | Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications. | Spring Boot |  |
| Vue.js | Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries. | Vue.js |  |