

**Building Applications in Go**

**Course Number:** GO-114  
**Duration:** 4 days

**Overview**

This Building Applications in Go training course teaches how to develop applications using the Golang programming language. Attendees learn how to use Go's standard library to overcome common programming challenges, build RESTful services, create a Go web application Docker image, and more.

**Prerequisites**

All attendees must be familiar with any modern programming language.

**Materials**

All Golang training students receive comprehensive courseware.

**Software Needed on Each Student PC**

* The latest distribution of Go for your operating system (Windows, Mac, or Linux)
* A Go-compatible IDE such as JetBrains GoLand
* Related lab files that Accelebrate provides

**Objectives**

* Understand Golang’s language constructs and unique features
* Write and call functions for proper code reuse
* Use the standard library to solve common problems
* Create concurrent applications in Go
* Construct RESTful services
* Build and deploy Docker containers for Go applications

**Outline**

* Introduction
  + Overview of the Go language
  + Comparison with other languages
  + Unique features of Go
* Building Blocks
  + Packages
  + Identifiers, variables, and constants
  + Operators
  + Data types
  + Control flow constructs
  + Pointers
* Functions in Go
  + Go functions
  + Function parameters
  + Higher order functions
  + Error handling
  + Returning results
  + Deferring function calls
* Error Handling
  + Errors as values
  + Handling errors
  + Creating custom errors
  + Panic and recovery
* Composite Types
  + Array type
  + Slice type
  + Map type
  + Struct type
* OO in Go
  + Objects in Go
  + Go methods
  + Inheritance using type embedding
  + Type aliasing
* Abstractions in Go
  + Interfaces in Go
  + Empty interfaces @ any
  + Type assertion
* Modularity
  + Modules
  + Using 3rd party modules
  + Packages and nested packages
  + Dependency injection using “wire”
* Concurrency
  + Goroutines
  + Channels
  + Passing data in and out of channels
  + Handling race conditions
  + Parallelism in Go
  + Sync package
  + Using context
  + Implementing worker pool
* IO in Go
  + Readers and writers
  + Exploring the io package
  + Working with files
  + Buffered IO
  + Encoding and decoding data
* Database Programming
  + Creating a database connection
  + Making database query
  + Adding, deleting, and updating records with Golang
* JSON and XML Binding with Golang
  + Serializing and deserializing JSON
  + Serializing and deserializing XML
* Web Programming
  + HTTP package
  + Building a web server
  + Using Gorilla Mux Router
  + Handling requests and responses
  + Serving static files
  + Server JSON responses
  + Cross-cutting concerns using middleware
  + Structured logging
* Testing
  + Organizing test code
  + Performing assertions
  + Detecting unreachable code
  + Mocking dependencies using GoMock
* Go and Docker
  + Building a Go Docker image
  + Running the Docker container
  + Building a Go web application docker image
* Conclusion