

Introduction

This lesson briefly introduces the application we will be designing in this chapter. Every lesson covers a part of the application. After every lesson, you'll achieve a milestone.

WE'LL COVER THE FOLLOWING



- Introducing Project UrlShortener

In this chapter, we will develop a complete program: *goto*, a URLShortener web application, because the web is all-ubiquitous, and we don't want to type long URLs. The example is taken from the [excellent lecture](#) from **Andrew Gerrand** at **FOSDEM 2011**. We will do this in **3** stages; each stage has more functionalities and shows progressively more features of the Go language. We will draw heavily on what we have learned about web applications in [Chapter 13](#).

- **Version 1:** a map and a struct are used, together with a `Mutex` from the `sync` package and a struct factory.
- **Version 2:** the data is made persistent because it is written to a file in gob-format.
- **Version 3:** the application is rewritten with goroutines and channels.
- **Version 4:** what has to change if we want a JSON-version.
- **Version 5:** a distributed version is made with the `rpc` protocol.



Introducing Project UrlShortener

You know that some addresses in the browser (called URLs) are (very) long and/or complex and that there are services on the web which turn these into a nice short URL, to be used instead. Our project is like that. It is a web service with two functionalities:

- **Add:** given a long URL, it returns a short version, e.g.,
http://maps.google.com/maps?f=q&source=s_q&hl=en&geocode=&q=tokyo&sll=37.0625,-95.677068&sspn=68.684234,65.566406&ie=UTF8&hq=&hnear=Tokyo,+Japan&t=h&z=9
(link A) becomes <http://goto/UrcGq> (link B) and stores this pair of data (all our short URL's start with <http://goto/>).
 - **Redirect:** whenever a shortened URL is requested, it redirects the user to the original, long URL. So, if you type (B) in a browser, it redirects you to the page of (A). For example, <http://goto/a> redirects to <http://google.com/> if it was shortened to <http://goto/a>.
-

Now that we know what the application is supposed to do, let's look at the data structures we will use for it in the next lesson.