

Создание веб-приложений с помощью языка программирования Go

Ленкин Алексей Викторович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Цель исследования продемонстрировать работу Go в качестве аналога серверного языка для веб-приложений. Методами исследования являются поиска теоретических источников по теме и практическое применение для создания прототипа. Итогом является выделение особенностей использования Go в качестве замены серверных языков.

Ключевые слова: go, серверный язык, веб-приложения

Creating web applications using the Go programming language

Lenkin Aleksei Viktorovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The aim of the study is to demonstrate the work of Go as an analogue of the server language for web applications. The methods of research are the search for theoretical sources on the topic and practical application to create a prototype. The result is the identification of features of using Go as a substitute for server languages.

Keywords: go, server language, web applications

В основе любого сайта на данный момент есть серверная часть, которая отвечает за обработку всей информации, а также обеспечивает сайт требуемым функционалом. Но при начале разработки такого сайта разработчик, как правило, задается вопросом, какой язык программирования использовать для создания серверной части веб-приложения.

Для этих целей может быть применён также и язык программирования Go. Являясь достаточно молодым языком он обладает необходимым функционалом необходимым для создания веб-приложений любой сложности.

Цель исследования продемонстрировать работу Go в качестве аналога серверного языка для веб-приложений.

Исследованиями в данной теме занимались следующие авторы. В.Н. Зимин и П.В. Федосов «Разработали веб-приложение на основе технологии AJAX в среде Microsoft Visual Studio» [1]. «Серверный JavaScript на примере библиотеки node.js» показали П.С. Зайцев и В.И. Воловач [2]. Е.М.

Юртанова написала статью о «Разработке Web-приложений с использованием языка PHP» [3]. А.Ю. Мелихов и Р.Ф. Юхимук «Создали Web-приложения для генерации отчетов на базе каркасной системы управления контентом "Django"» [4].

Покажем на примере, как создать базовый HTTP сервер в Go. Сначала давайте сказать о том, на что должен быть способен HTTP сервер. Базовый HTTP сервер имеет несколько ключевых задач, о которых нужно позаботиться:

- Обработка динамических запросов: Обрабатывать входящие запросы от пользователей, которые просматривают веб-сайт, заходят в свои аккаунты или размещают изображения.

- Обслуживать статические активы: Подавайте JavaScript, CSS и изображения браузерам, чтобы создать динамические возможности для пользователя.

- Принять соединения: HTTP-сервер должен прослушивать определенный порт, чтобы иметь возможность принимать соединения из интернета.

Пакет `net/http` содержит все утилиты, необходимые для приема запросов и их динамической обработки. Мы можем зарегистрировать новый обработчик с помощью функции `http.HandleFunc`. Первый параметр это путь, а второй - выполняемая функция [5].

Начнём создание простейшего веб-приложения. Сначала создадим html страницу с формой отправки своего имени (рис.1).

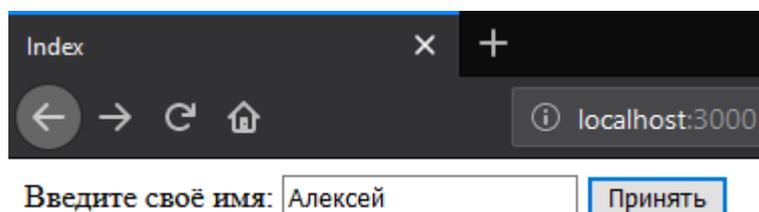


Рисунок 1. Форма для отправки имени

Код страницы «`index.html`»:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Index</title>
  </head>
  <body>
    <form method="POST" action="/">
      <label>Введите своё имя:</label>
      <input type="text" name="yourname"/>
      <input type="submit" value="Принять"/>
    </form>
  </body>
</html>
```

Далее на языке программирования Go создадим программу, которая будет выполнять следующие функции:

1. Запускает HTTP-сервер на порт 3000
2. Сканирует данный порт и принимает или отправляет запросы по протоколу TCP
3. При получении запроса типа GET или POST выдает сообщение вида «Привет, % yourname %», где % yourname % - введенное на html-форме имя.

Листинг программы представлен ниже:

```
package main

import (
    "fmt"
    "log"
    "net/http"
)

func getpost(w http.ResponseWriter, r *http.Request) {
    if r.URL.Path != "/" {
        http.Error(w, "Ошибка 404, страница не найдена.",
            http.StatusNotFound)
        return
    }
    switch r.Method {
    case "GET":
        http.ServeFile(w, r, "index.html")
    case "POST":
        if err := r.ParseForm(); err != nil {
            fmt.Fprintf(w, "Ошибка ", err)
            return
        }
        yourname := r.FormValue("yourname")
        fmt.Fprintf(w, "Привет, %s\n", yourname)
    default:
        fmt.Fprintf(w, "Неправильный запрос")
    }
}

func main() {
    http.HandleFunc("/", getpost)
    fmt.Printf("Listening...\n")
    if err := http.ListenAndServe(":3000", nil); err != nil {
        log.Fatal(err)
    }
}
```

В итоге при запуске программы по адресу «localhost:3000» действительно открывается форма для ввода имени и при нажатии кнопки «Принять» выводится правильное сообщение (рис.2).

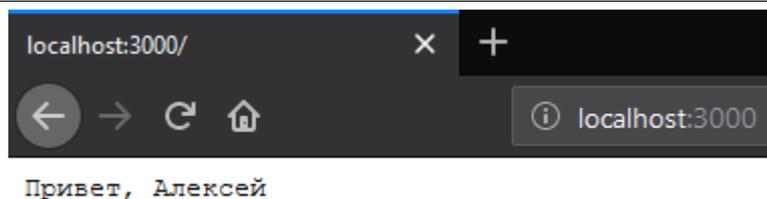


Рисунок 2. Итог отправки POST-запроса в форме

Язык Go в этом примере выступил как серверный, так как выполнял те функции, которые исполняются на стороне сервера, а не клиента. Но главным его преимуществом над другими такими же языками является отсутствие необходимости включенного веб-сервера, таких как Apache или nginx, так как запускает свой собственный встроенный сервер. Такой подход упрощает развертывание такого веб-приложения, ведь оно не использует никаких сторонних программ, которые могли бы влиять на его работоспособность.

Библиографический список

1. Зимин В.Н., Федосов П.В. Разработка веб-приложений на основе технологии AJAX в среде Microsoft Visual Studio // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2008. № 51. С. 103-107.
2. Зайцев П.С., Воловач В.И. Серверный JavaScript на примере библиотеки node.js // Наука - промышленности и сервису. 2010. № 5-2. С. 55-58.
3. Юртанова Е.М. Разработка Web-приложений с использованием языка PHP // Учебный эксперимент в образовании. 2011. № 1. С. 47-50.
4. Мелихов А.Ю., Юхимук Р.Ф. Создание Web-приложения для генерации отчетов на базе каркасной системы управления контентом "Django" // Вестник Югорского государственного университета. 2011. № 3 (22). С. 27-30.
5. HTTP Server // Go Web Examples URL: <https://gowebexamples.com/http-server/> (дата обращения: 05.09.2019).