



MARSHRUTLASH

Yusupov Mirsaid Abdulaziz o'g'li

Farg'ona Davlat Universiteti Amaliy matematika va informatika kafedrası

o'qituvchisi

mirsaidbeky@gmail.com

Qodirova Gulnora Akmaljon qizi

Farg'ona Davlat Universiteti, talaba

qodirova035@gmail.com

Anotatsiya: *Marshrutlash (routing) — bu foydalanuvchi so'rovlari va server javoblari o'rtasidagi yo'nalishlarni belgilash jarayoni bo'lib, zamonaviy web dasturlar va ilovalarda muhim o'rin egallaydi. Ushbu maqolada marshrutlash tushunchasi, uning asosiy prinsiplari va dasturlash texnologiyalarida qo'llanishi yoritib beriladi. Marshrutlash orqali web ilovalar ichida URL manzillarga javob berish, resurslarga yo'naltirish va foydalanuvchi interfeyslarini boshqarish masalalari ko'rib chiqiladi. Shuningdek, PHP, Node.js, ASP.NET va boshqa texnologiyalardagi marshrutlash misollari keltirilib, ularning afzallik va kamchiliklari tahlil qilinadi.*

Kalit so'zlar: *Marshrutlash, web ilova, URL manzil, HTTP so'rov, HTTP javob, server, mijoz, router, middleware, PHP routing, Node.js routing, ASP.NET routing, yo'naltirish, controller, REST API, dynamic routing, static routing, xavfsizlik, autentifikatsiya, authorization, web framework.*

Annotation: *Routing is the process of defining the paths between user requests and server responses, playing a key role in modern web applications and systems. This article explores the concept of routing, its main principles, and its implementation in programming technologies. The paper covers how routing helps handle URL addresses, redirects to resources, and manages user interfaces within*



web applications. Examples of routing in PHP, Node.js, ASP.NET, and other technologies are provided, analyzing their advantages and disadvantages.

Keywords: Routing, web application, URL address, HTTP request, HTTP response, server, client, router, middleware, PHP routing, Node.js routing, ASP.NET routing, redirection, controller, REST API, dynamic routing, static routing, security, authentication, authorization, web framework.

Аннотация: Маршрутизация — это процесс определения путей между запросами пользователей и ответами сервера, играющий важную роль в современных веб-приложениях и системах. В данной статье рассматривается понятие маршрутизации, её основные принципы и применение в технологиях программирования. Также описывается, как маршрутизация помогает обрабатывать URL-адреса, перенаправлять на ресурсы и управлять пользовательским интерфейсом внутри веб-приложений. Приводятся примеры маршрутизации в PHP, Node.js, ASP.NET и других технологиях с анализом их преимуществ и недостатков.

Ключевые слова: Маршрутизация, веб-приложение, URL-адрес, HTTP-запрос, HTTP-ответ, сервер, клиент, роутер, промежуточное ПО, маршрутизация PHP, маршрутизация Node.js, маршрутизация ASP.NET, перенаправление, контроллер, REST API, динамическая маршрутизация, статическая маршрутизация, безопасность, аутентификация, авторизация, веб-фреймворк.

Kirish

Marshrutlash (routing) bugungi kunda dasturlash texnologiyalarida muhim tushunchalardan biri hisoblanadi. Ayniqsa, web ilovalar va web xizmatlar rivojlanishi bilan marshrutlash jarayoni asosiy mexanizmlardan biriga aylandi. Har



bir foydalanuvchi tomonidan yuborilgan so‘rov ma’lum bir URL manzilga mos keluvchi javobga yo‘naltiriladi va ushbu jarayonni to‘g‘ri tashkil qilish web ilova samaradorligi va xavfsizligi uchun katta ahamiyatga ega.

Marshrutlash nafaqat URL va so‘rovlarni boshqarish, balki foydalanuvchi tajribasini yaxshilash, ma’lumotlar oqimini optimallashtirish va server resurslaridan to‘g‘ri foydalanishga ham xizmat qiladi. Turli dasturlash muhitlari va frameworklar marshrutlashni o‘ziga xos usullarda amalga oshiradi: PHP, Node.js, ASP.NET, Python Django va boshqa ko‘plab texnologiyalarda marshrutlashning o‘z qoidalari va imkoniyatlari mavjud.

Mazkur maqolada marshrutlash tushunchasi, uning dasturlash jarayonidagi o‘rni, asosiy prinsiplari va qo‘llanilish sohasi keng yoritiladi. Shuningdek, marshrutlashda qo‘llaniladigan texnologiyalar, ularning imkoniyatlari va amaliy jihatlari kod misollari bilan birgalikda tahlil qilinadi.

Asosiy qism

1. Marshrutlash tushunchasi va uning vazifalari

Marshrutlash (routing) – bu web serverga kelgan foydalanuvchi so‘rovlari qaysi resursga, qaysi controllerga yoki funksiya/metodga yo‘naltirilishini belgilab beruvchi jarayondir. Har bir URL manzil ma’lum bir logikaga ega bo‘lib, shu manzil orqali foydalanuvchiga kerakli ma’lumot yoki interfeys ko‘rsatiladi.

Masalan, <https://example.com/profile> manzili **foydalanuvchi profili** sahifasiga olib boradi, <https://example.com/admin> esa **admin panelga** yo‘naltiradi. Shu yo‘nalishlarni boshqarish **routing tizimi** orqali amalga oshiriladi. Marshrutlashning asosiy vazifalari quyidagilardan iborat:

- So‘rovlarni to‘g‘ri controllerga yo‘naltirish
- URL manzillarni boshqarish va standartlashtirish



- Xatoliklarni boshqarish (masalan, 404 page)
- Qo‘shimcha middleware larni ishga tushirish

2. Marshrutlash turlari

Marshrutlash ikki asosiy turga bo‘linadi:

• **Statik marshrutlash** – har bir URL aniq bir resursga bog‘lanadi, oldindan belgilangan.

• **Dinamik marshrutlash** – URL ichidagi parametrlar va shartlarga qarab yo‘nalish belgilanadi.

Masalan:

php

КопироватьРедактировать

// PHP statik marshrut

```
if ($_SERVER['REQUEST_URI'] == '/home') {  
    include 'home.php';  
}
```

php

КопироватьРедактировать

// PHP dinamik marshrut

```
if (preg_match('/\/user\/(\d+)/', $_SERVER['REQUEST_URI'], $matches)) {  
    $userId = $matches[1];  
    include 'user_profile.php';  
}
```

Statik marshrutlar oddiy saytlar uchun qulay, dinamik esa murakkab web ilovalar uchun muhimdir.



3. PHP da marshrutlash

PHP da oddiy marshrutlash `$_SERVER['REQUEST_URI']` orqali amalga oshiriladi. Lekin zamonaviy PHP frameworklari (Laravel, Symfony) o'zlarining qulay **router** tizimlariga ega. Masalan, Laravel frameworkida routing quyidagicha yoziladi:

php

КопироватьРедактировать

```
Route::get('/home', function () {  
    return view('home');  
});  
  
Route::get('/user/{id}', function ($id) {  
    return 'User ID: ' . $id;  
});
```

Bu kod `https://site.com/home` so'rovini `home.blade.php` sahifasiga yo'naltiradi va `https://site.com/user/1` so'rovini foydalanuvchi ID bilan ko'rsatadi.

PHP routing orqali URL larni **REST API** tamoyillari asosida boshqarish, **middleware** larni qo'shish va xavfsizlikni ta'minlash imkonini beradi.

4. Node.js (Express.js) da marshrutlash

Node.js da mashhur frameworklardan biri – **Express.js** bo'lib, unda routing juda qulay amalga oshiriladi. Quyidagi kod misoliga e'tibor bering:

javascript

КопироватьРедактировать

```
const express = require('express');
```



```
const app = express();

app.get('/home', (req, res) => {
  res.send('Welcome to Home Page');
});

app.get('/user/:id', (req, res) => {
  res.send(`User ID: ${req.params.id}`);
});

app.listen(3000, () => console.log('Server running on port 3000'));
```

Bu yerda /home va /user/:id marshrutlari belgilandi. Express.js da middleware lar orqali har bir marshrutga qo'shimcha tekshiruvlar, autentifikatsiya va logging qo'shish mumkin.

5. ASP.NET da marshrutlash

ASP.NET MVC yoki ASP.NET Core frameworkida marshrutlash quyidagicha amalga oshiriladi:

```
csharp
КопироватьРедактировать
app.UseRouting();

app.UseEndpoints(endpoints =>
{
  endpoints.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
});
```



Bu kod URL ni controller/action/id formatida ajratadi va mos controller va action methodga yo'naltiradi. ASP.NET marshrutlash tizimi kuchli va o'ta moslashuvchan bo'lib, korporativ web ilovalar uchun qulaylik yaratadi.

6. Marshrutlashda xavfsizlik va autentifikatsiya

Marshrutlashda xavfsizlik masalasi juda muhim. Chunki foydalanuvchi so'rovlari noto'g'ri yo'naltirilsa, tizimga zarar yetishi mumkin. Quyidagi xavfsizlik choralariga e'tibor qaratish lozim:

- Har bir marshrut uchun autentifikatsiya va authorization talab qilish
- Kirish mumkin bo'lmagan marshrutlar uchun 403 Forbidden javobini yuborish
- URL parametrlarini tozalash va tekshirish (SQL Injection, XSS dan himoya)
- HTTPS protokolini ishlatish

Masalan, Laravel da middleware orqali autentifikatsiyani qo'shish:

php

КопироватьРедактировать

```
Route::get('/dashboard', function () {  
    return view('dashboard');  
})->middleware('auth');
```

Node.js Express.js da esa:

javascript

КопироватьРедактировать

```
app.get('/dashboard', isAuthenticated, (req, res) => {  
    res.send('Welcome to Dashboard');  
});
```



7. Qo'shimcha texnologiyalar

Marshrutlashda **middleware**, **REST API** va **dynamic routing** kabi qo'shimcha texnologiyalar keng qo'llaniladi. Middleware yordamida har bir marshrutga qo'shimcha funksiyalar qo'shish mumkin: logging, autentifikatsiya, exception handling. REST API esa marshrutlarni CRUD operatsiyalar (Create, Read, Update, Delete) asosida boshqarishga imkon beradi.

Xulosa

Marshrutlash – zamonaviy dasturlash texnologiyalarida asosiy komponentlardan biri bo'lib, web ilovalar va xizmatlar ish faoliyatida muhim o'rin egallaydi. Ushbu maqolada marshrutlash tushunchasi, uning asosiy turlari, PHP, Node.js, ASP.NET kabi texnologiyalarda qo'llanilishi va ularning texnik jihatlari keng yoritildi.

Marshrutlash orqali foydalanuvchidan kelgan har bir HTTP so'rov to'g'ri manzilga yetkazilib, mos javob qaytarilishi ta'minlanadi. Shuningdek, marshrutlash jarayonida xavfsizlik, autentifikatsiya va URL parametrlarining tekshirilishi juda muhim. Har bir dasturlash muhiti va framework marshrutlash tizimini o'ziga xos usullar bilan tashkil etadi va bu esa dasturchilarga loyihalarni samarali boshqarish imkoniyatini beradi.

Kelajakda marshrutlash tizimlari yanada rivojlanib, dinamik va moslashuvchan marshrutlarni boshqarish, shuningdek, mikroservis arxitekturalarida marshrutlashni avtomatlashtirish kabi imkoniyatlar kengayadi. Shu sababli, marshrutlashni chuqur o'rganish va uni to'g'ri qo'llash har bir dasturchining muhim ko'nikmalaridan biridir.



Foydalanilgan adabiyotlar

1. Brad Dayley, Brendan Dayley, Caleb Dayley – *Node.js, MongoDB, and Angular Web Development*. Addison-Wesley, 2017.
2. Andrew Mead – *The Complete Node.js Developer Course*. Packt Publishing, 2019.
3. Jeffrey Way – *Laravel: Up & Running: A Framework for Building Modern PHP Apps*. O'Reilly Media, 2019.
4. Microsoft Docs – *ASP.NET Core Routing Documentation*. Microsoft, 2024.
5. David Flanagan – *JavaScript: The Definitive Guide*. O'Reilly Media, 2020.
6. Rick Viscomi, Andy Davies, Marcel Duran – *Using WebPageTest: Web Performance Testing for Novices and Power Users*. O'Reilly Media, 2017.
7. Matthew MacDonald – *Beginning ASP.NET 4.5 in C#*. Apress, 2012.
8. PHP Manual – *PHP Official Documentation*. php.net, 2024.
9. Eric Freeman, Elisabeth Robson – *Head First Design Patterns*. O'Reilly Media, 2021.
10. Jon Duckett – *PHP & MySQL: Server-side Web Development*. Wiley, 2022.
11. Nicholas C. Zakas – *Professional JavaScript for Web Developers*. Wrox, 2020.
12. Adam Freeman – *Pro ASP.NET Core MVC 2*. Apress, 2017.
13. Robin Nixon – *Learning PHP, MySQL & JavaScript*. O'Reilly Media, 2021.
14. Ethan Brown – *Web Development with Node and Express*. O'Reilly Media, 2019.
15. Frank M. Carrano – *Data Abstraction and Problem Solving with C++*. Pearson, 2017.