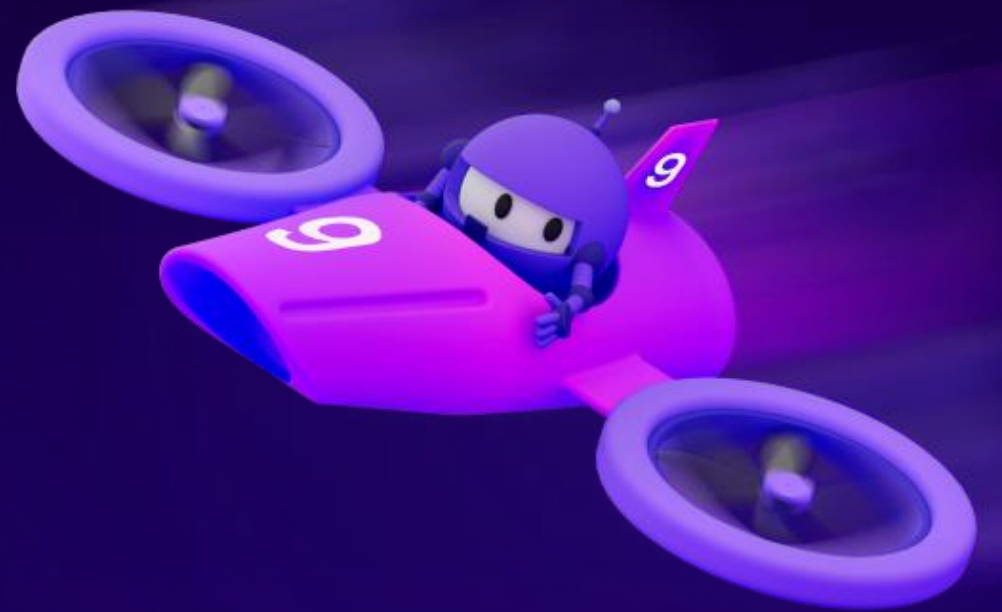


Что нового в ASP.NET Core 9.0

Георгий Гуцол
.NET Разработчик
ИТ Контакт



Содержание

- Оптимизация доставки ресурсов
- Blazor
 - Расширение работы с рендерингом
 - Улучшение переподключения
 - Новый шаблон
 - Инъекция через конструктор
 - Улучшение компонентов
- ASP.NET
 - Страница исключения
 - Улучшение отладки словарей
- OpenAPI
- HybridCache



Оптимизация доставки ресурсов

Name	×	Headers	Preview	Response	Initiator	Timing
<input checked="" type="checkbox"/> bootstrap.min.46ein0sx1k.css	▼ General					
<input checked="" type="checkbox"/> app.khy4lop6wu.css	Request URL: https://localhost:7081/app.khy4lop6wu.css					
<input checked="" type="checkbox"/> BlazorApp1.Idd75up0b5.styles....	Request Method: GET					
<input checked="" type="checkbox"/> Microsoft.AspNetCore.Compo...	Status Code: 200 OK					
	Remote Address: [::1]:7081					
	Referrer Policy: strict-origin-when-cross-origin					

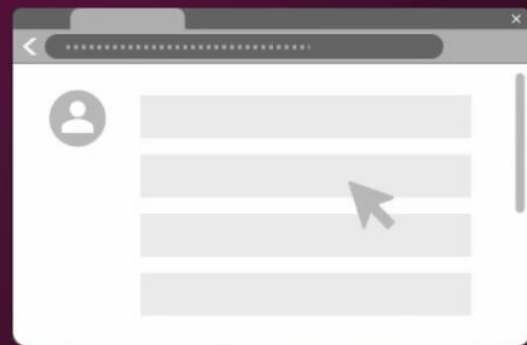


Оптимизация доставки ресурсов

```
Program.cs      — □ ×  
  
//app.UseStaticFiles();  
app.MapStaticAssets();
```



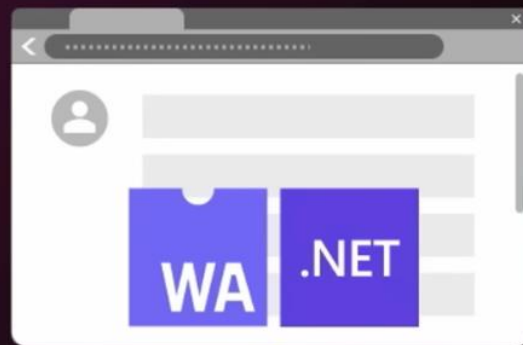
Рендеринг



WebSocket (SignalR)

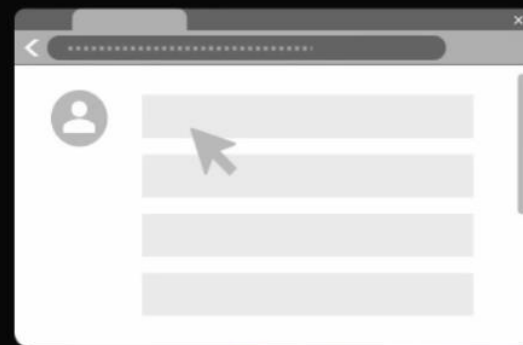


Server



WebAssembly

Auto



HTTP

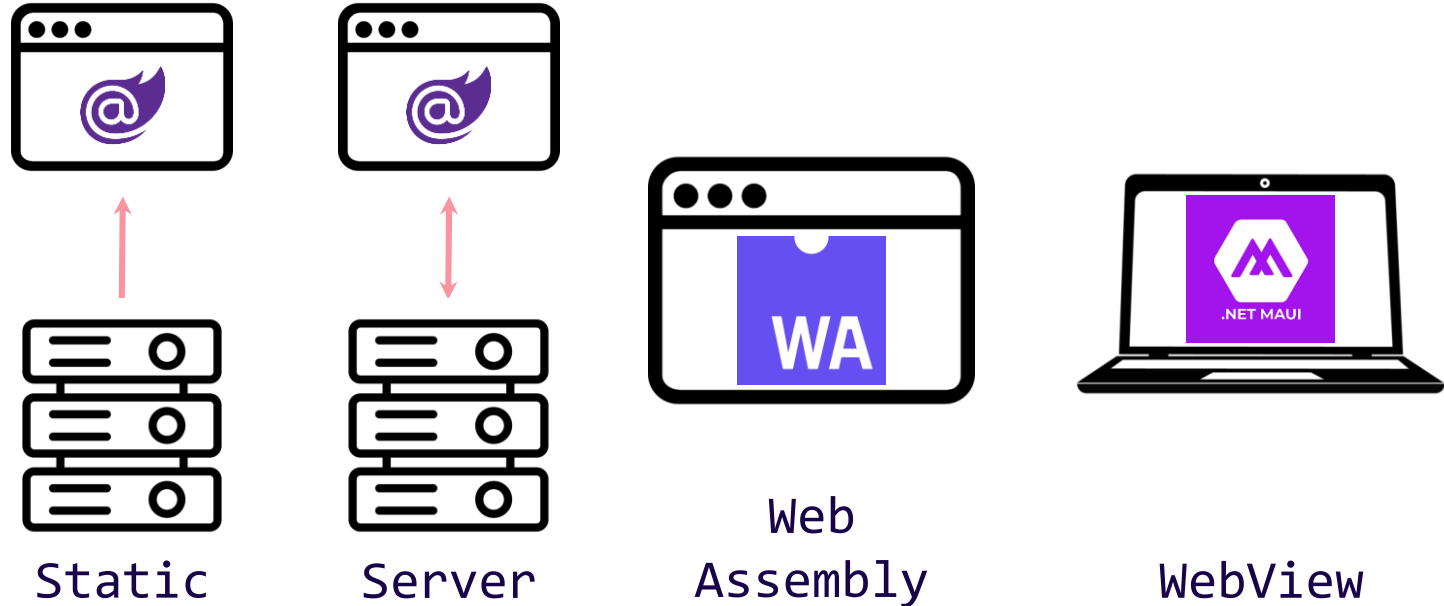
HTML



NEW Static Server-side Rendering (SSR)

Рендеринг

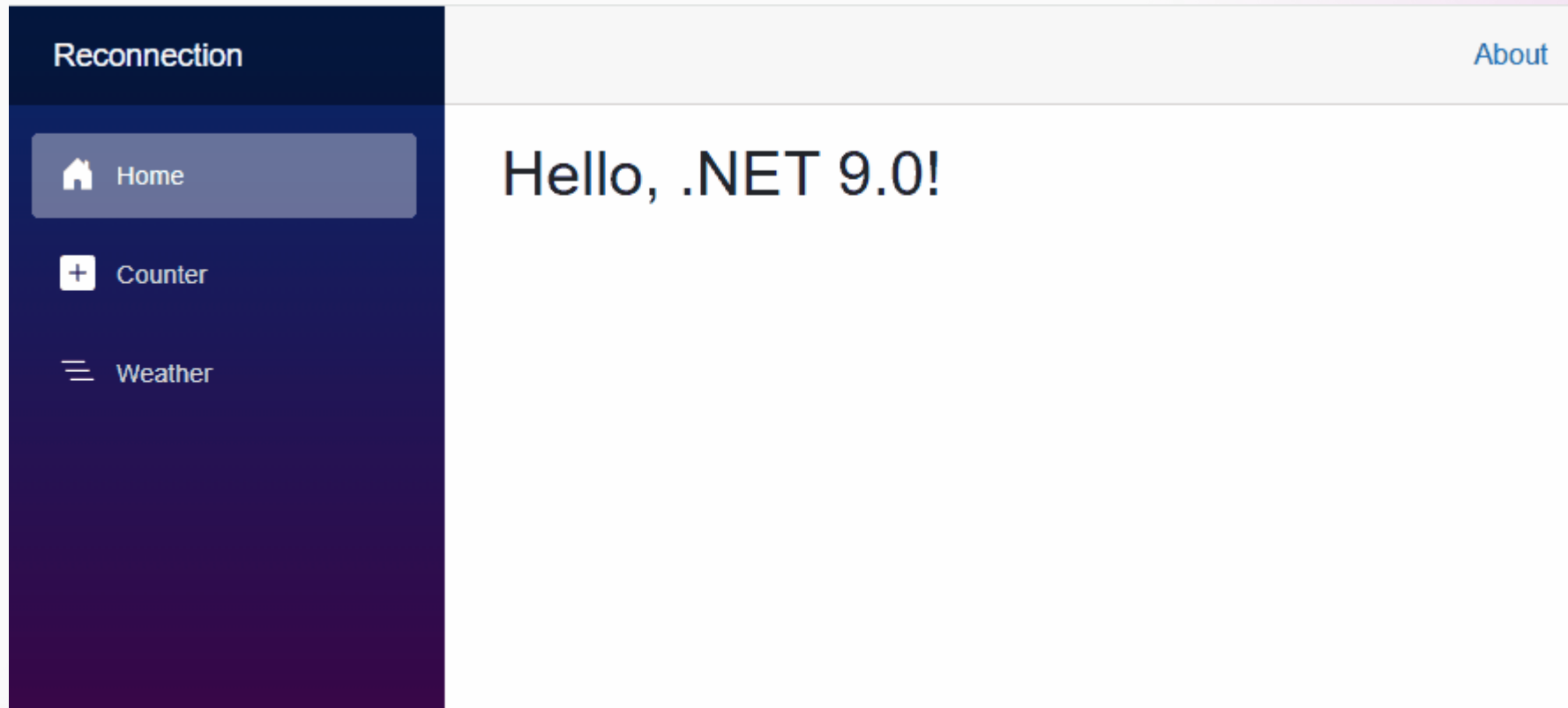
RenderInfo.Name



RenderInfo.IsInteractive Компонент интерактивен?

`IComponentRenderMode` AssignedRenderMode { get; }

Улучшение переподключения



Rejoining the server...

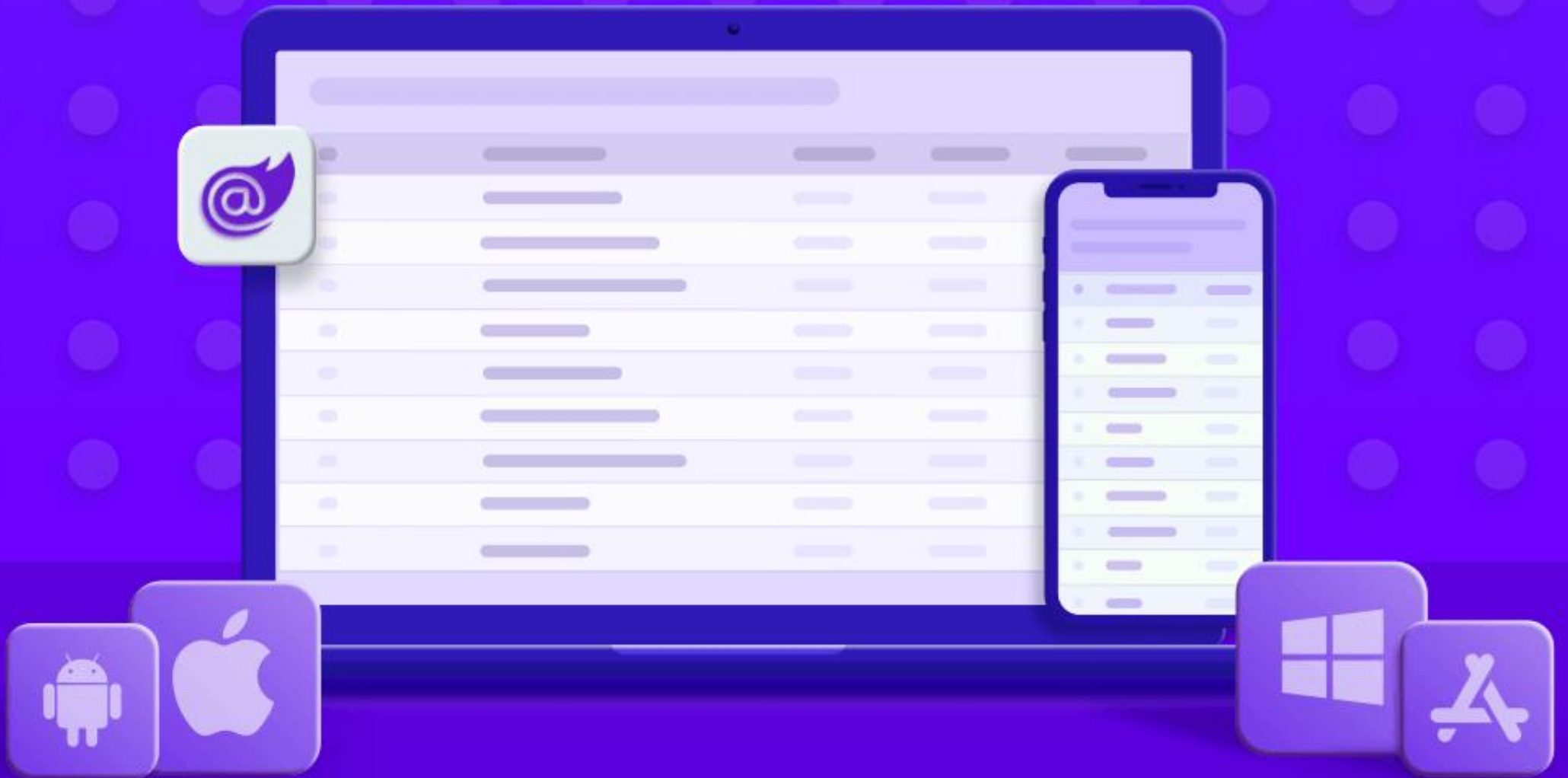


Rejoin failed... trying again in 5 seconds



Rejoin failed... trying again in 30 seconds

.NET MAUI Blazor Hybrid and Web App



Инъекция в конструктор

ConstructorInjection.razor

```
@page "/constructor-injection"
```

```
<button @onclick="HandleClick">
```

```
    Take me to the Counter component
```

```
</button>
```

ConstructorInjection.razor.cs

```
using Microsoft.AspNetCore.Components;
```

```
public partial class ConstructorInjection
```

```
(NavigationManager navigation)
```

```
{
```

```
    private void HandleClick()
```

```
    {
```

```
        navigation.NavigateTo("/counter");
```

```
    }
```

```
}
```

PropertyInjection.razor

```
@page "/property-injection"
```

```
@inject NavigationManager Navigation
```

```
<button @onclick="@(() => Navigation.NavigateTo("/counter"))">
```

```
    Take me to the Counter component
```

```
</button>
```

Обновления компонентов

QuickGrid.razor



```
<QuickGrid Virtualize="true" OverscanCount="4">  
    ...  
</QuickGrid>
```

Etravirine	HUMAN PRESCRIPTION DRUG	06/14/2021
Eucerin Advance Hydration Sunscreen Spf 30	HUMAN OTC DRUG	12/01/2021
Eucerin Advanced Hydration Spf 50 Sunscreen	HUMAN OTC DRUG	11/03/2021
Eucerin Advanced Hydration Spf 50 Sunscreen	HUMAN OTC DRUG	11/03/2021
Eucerin Advanced Hydration Sunscreen Spf 50	HUMAN OTC DRUG	12/01/2021
Eucerin Age Defense Spf 50 Sunscreen	HUMAN OTC DRUG	11/03/2021
Eucerin Baby Sensitive Mineral Spf 50 Sunscreen	HUMAN OTC DRUG	11/03/2021
Eucerin Eczema Relief Body	HUMAN OTC DRUG	08/15/2021
Eucerin Oil-Control Spf 50 Sunscreen	HUMAN OTC DRUG	11/03/2021
Eucerin Sensitive Mineral Spf 50 Sunscreen	HUMAN OTC DRUG	11/03/2021
Eulexin	HUMAN PRESCRIPTION DRUG	11/12/2021
Eulexin	HUMAN PRESCRIPTION DRUG	11/12/2021

InputNumber.razor



```
<InputNumber @bind-Value="currentInput"  
    max="10" min="1"  
    step="1" type="range" />
```

Input Number demo

- Home
- Input Number
- Weather

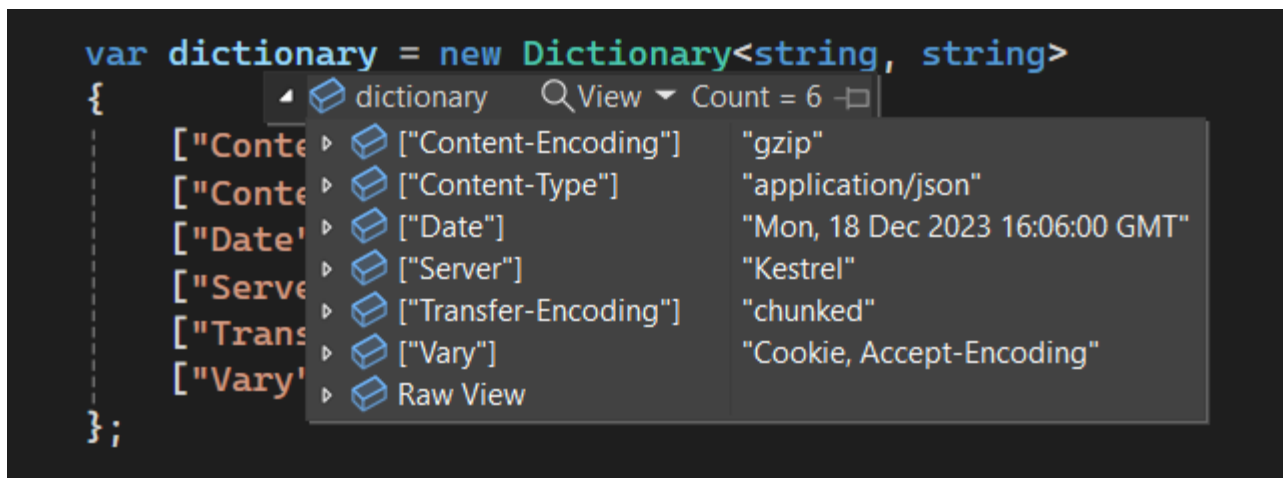
Input number

Current input: 5

Улучшения для разработчиков

Отладка словарей

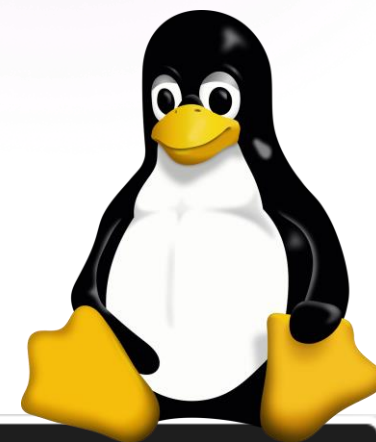
```
var dictionary = new Dictionary<string, string>
{
    ["Content-Encoding"] = "gzip"
    ["Content-Type"] = "application/json"
    ["Date"] = "Mon, 18 Dec 2023 16:06:00 GMT"
    ["Server"] = "Kestrel"
    ["Transfer-Encoding"] = "chunked"
    ["Vary"] = "Cookie, Accept-Encoding"
};
```



HTTPS сертификат разработчика в Linux



```
dotnet dev-certs https --trust
```



Страница исключения

- Улучшили внешний вид
- Добавили подраздел
Endpoint Metadata

An unhandled exception occurred while processing the request.

InvalidOperationException: A test error

Program+<>c__DisplayClass0_0.<<Main>>\$>b__0() in Program.cs, line 37

Stack Query Cookies Headers Routing

Endpoint

Name	Value
Display Name	HTTP: GET /weatherforecast
Route Pattern	/weatherforecast
Route Order	0
Route HTTP Method	GET

Endpoint Metadata

Type	Detail
RuntimeMethodInfo	WeatherForecast[] <<Main>>\$>b__0()
HttpMethodMetadata	HttpMethods: GET, Cors: False
ProducesResponseTypeMetadata	Produces StatusCode: 200, ContentTypes: application/json, Type: WeatherForecast[]
EndpointNameMetadata	EndpointName: GetWeatherForecast
RouteNameMetadata	RouteName: GetWeatherForecast
OpenApiOperation	Microsoft.OpenApi.Models.OpenApiOperation
RouteDiagnosticsMetadata	Route: /weatherforecast

Route Values

No route values.

OpenApi

Microsoft.AspNetCore.OpenApi



Генерация на этапе сборки и в рантайме



Native AOT



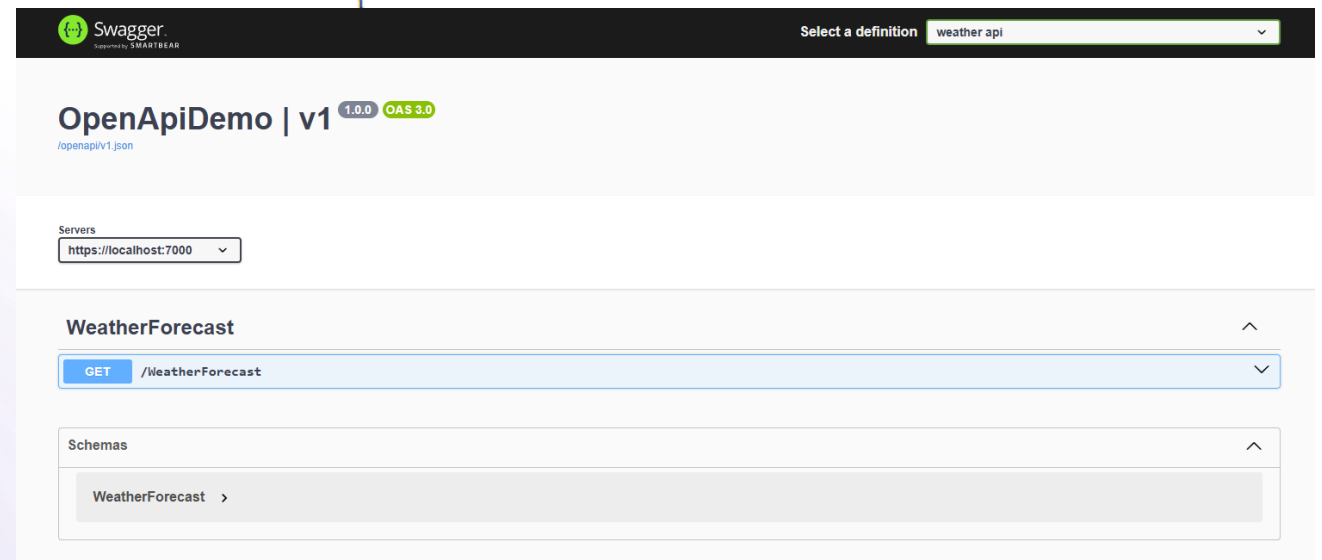
Кастомизируемый



Поддержка существующих UI

```
//...
builder.Services.AddOpenApi();
var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.MapOpenApi();
    app.UseSwaggerUI(options =>
        options.SwaggerEndpoint
            ("/openapi/v1.json",
            "weather api"));
}
```



Hybrid Cache

Зачем?

Существующие API



Тяжело использовать хорошо



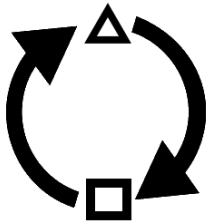
Много сопутствующего кода

Цель



Упростить работу с кэшем

IMemoryCache



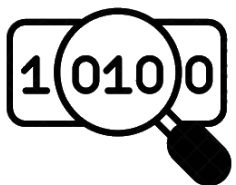
Нет иммутабельности



Привязан к
единственному
экземпляру приложения

```
builder.Services.AddMemoryCache();  
//...  
var key = $"/somepath/{id}";  
if (!(memoryCache.TryGetValue(key, out object? untyped)  
    && untyped is SomeBackendData value))  
{  
    value = await UnderlyingDataFetchAsync(id, cancellationTokens);  
    memoryCache.Set(key, value, Expiration);  
}  
  
return value; //изменяемый объект
```


IDistributedCache



Сериализация в byte[]



Write/read задержка



Риск недоступности

```
builder.Services.AddStackExchangeRedisCache(...);  
//...  
var key = $"/somepath/{id}";  
//запрашиваем данные  
var arr = await distributedCache.GetAsync(key, cancellationToken);  
  
SomeBackendData value;  
if (arr is null)  
{  
    //Не попали в кэш. Грузим данные из источника  
    value = await UnderlyingDataFetchAsync(id, cancellationToken);  
  
    //Сериализуем и кладем в кэш  
    arr = SomeSerializer.Serialize(value);  
    await distributedCache.SetAsync(key, arr,  
        OptionsWithExpiration, cancellationToken);  
}  
else  
{  
    //Есть попадание. Десериализуем.  
    value = SomeSerializer.Deserialize<SomeBackendData>(arr);  
}  
return value;
```


IHybridCache

IMemoryCache/IDistributedCache

Нет защиты от Stampede

Сериализация только в byte[]

Предоставляет мутабельные объекты

IHybridCache

Защита от Stampede реализована по умолчанию

Конфигурируемая сериализация

Позволяет работать с мутабельными и иммутабельными типами

Использует реализацию IDistributedCache с низкой аллокацией

```
builder.Services.AddHybridCache(...);  
// ...  
return await hybridCache.GetOrCreateAsync(  
    $"/somepath/{id}",  
    cancellation => UnderlyingDataFetchAsync(id, cancellation),  
    cancellationToken: cancellationToken  
);
```



Оптимизация
доставки
статических
ресурсов



Улучшения
обработки
исключений и
отладки



Authentication
enhancements



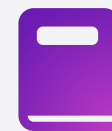
Новый шаблон
Blazor Hybrid



Улучшенные
метрики Kestrel



Обновление
distributed
tracing в
SignalR



Удобная
отладка
словарей

Что нового в ASP.NET Core .NET 9



Встроенная
поддержка
OpenAPI



Чтение
rendering
mode Blazor



Hybrid
Cache



Поддержка AOT
SignalR



Обновленный
реконнект
Blazor



Trust Developer
certs on Linux

Blazor
загружается на

25%

быстрее



Поддержка
Keyed service в
middleware

Анализ улучшений в SignalR
.NET 9

Что нового для Web API в
ASP.NET Core 9

Authorization and
authentication improvements
on .NET 9

Дальше на SpbDotNet!

Видео



What's New for
ASP.NET Core &
Blazor in .NET 9

API-ly Ever After:
OpenAPI in .NET 9



Introducing
HybridCache in
ASP.NET Core

Статьи



What's new in
ASP.NET Core 9.0



Generate OpenAPI
documents

HybridCache
library in
ASP.NET Core

