

ASP.NET Core

Módulo 3

Prof. Samuel Santos

In the top left corner, there is a large, irregular purple shape and a smaller, solid purple circle below it.

Nesta aula

ASP.NET Core WEB API

A large, light grey, curved shape is positioned in the bottom right corner of the slide.

ASP.NET Core



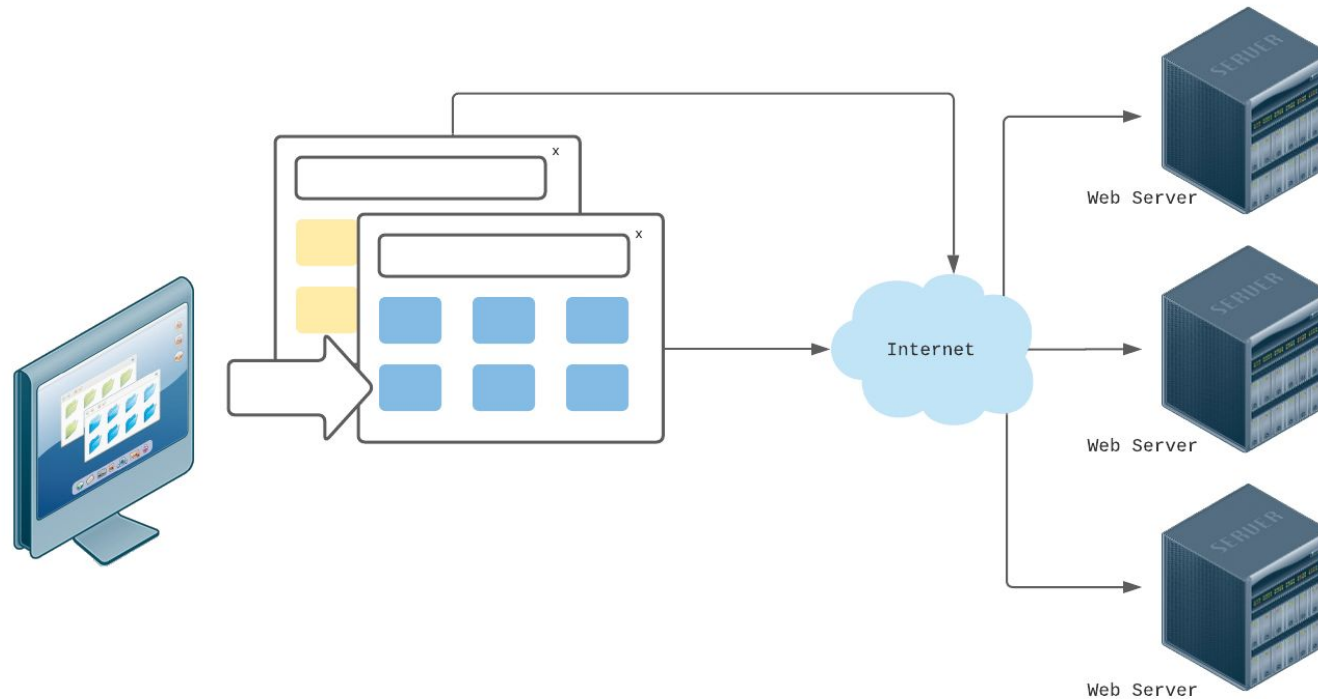
- ☐ O que é ?
- ☐ Injeção de Dependência.
- ☐ Manutenção da API.
- ☐ Por que escolher o ASP.NET Core?

Nesta aula

Protocolo HTTP

Protocolo HTTP

❑ Como funciona HTTP



Comunicação HTTP

Mecanismos HTTP



- Requisição (Request) HTTP.
- O que é um método HTTP?
- Cabeçalho da Requisição (Request) HTTP.
- Corpo da Requisição (Request) HTTP.
- Resposta (Response) HTTP.
- Código HTTP.
- Cabeçalho da Resposta (Response) HTTP.
- Corpo da Resposta (Response) HTTP.
- HTTPS.



Nesta aula

O que é uma API ?

API - Interface de Programação de Aplicação

- ☐ O que é REST?
- ☐ O que é SOAP?
- ☐ SOAP API vs. REST API.
- ☐ Benefícios da API REST sobre SOAP.
- ☐ Usando métodos HTTP para serviços RESTFUL.

Métodos HTTP para Serviços RESTFUL

HTTP VERB	CRUD	COLEÇÃO (/clientes)	ITEM ESPECÍFICO (/clientes/{id})
POST	Create (Criar)	201 (Criado), cabeçalho 'Location' com link para /customers/{id} contendo novo ID.	404 (não encontrado), 409 (conflito) se o recurso já existir.
GET	Read (Ler)	200 (OK), lista de clientes. Use paginação, classificação e filtragem para navegar por grandes listas.	200 (OK), cliente único. 404 (não encontrado), se ID não encontrado ou inválido.
PUT	Update/Replace	405 (Método não permitido), a menos que você queira atualizar / substituir todos os recursos em toda a coleção.	200 (OK) ou 204 (sem conteúdo). 404 (não encontrado), se ID não encontrado ou inválido.
PATCH	Update/Modify	405 (Método não permitido), a menos que você queira modificar a coleção em si.	200 (OK) ou 204 (sem conteúdo). 404 (não encontrado), se ID não encontrado ou inválido.
DELETE	Delete (Deletar)	405 (Método não permitido), a menos que você queira excluir toda a coleção - o que não é desejável com frequência.	200 (OK). 404 (não encontrado), se ID não encontrado ou inválido.

Nesta aula

Arquitetura REST

ARQUITETURA REST



- ☐ Servidor Cliente.
- ☐ Sem estado (Stateless).
- ☐ Cache.
- ☐ Interface Uniforme.
- ☐ Sistema em Camadas.
- ☐ Código sob demanda (opcional).

Nesta aula

Ambientes de Desenvolvimento

Visual Studio 2019



Visual Studio 2019

- ☐ Visual Studio (versão completa) é um ambiente de desenvolvimento "completo" e "conveniente".
- ☐ Community é uma versão simplificada da versão completa e substitui como edições expressas usadas antes de 2015.
- ☐ É uma solução completa usada principalmente por desenvolvedores .NET e relacionados. Inclui tudo, desde controle de origem a ferramentas de rastreamento de bugs etc. Tem tudo o que é necessário para desenvolver.

Visual Studio Code



Visual Studio Code

- ☐ É um editor de código-fonte leve que pode ser usado para visualizar, editar, executar e depurar códigos-fonte para aplicativos.
- ☐ Simplesmente é o Visual Studio sem a interface visual.
- ☐ É principalmente orientado em torno de arquivos, não projetos.
- ☐ Não tem suporte para o sistema de controle de versão da Microsoft; Team Foundation Server.
- ☐ É usado principalmente por desenvolvedores em um Mac que lida com as tecnologias do lado do cliente (HTML, JavaScript e CSS).

Visual Studio For Mac



Visual Studio For Mac

- ☐ IntelliSense avançado.
- ☐ Visual Studio for Mac traz o IntelliSense.
- ☐ Depurador efetivo.
- ☐ Refatoração Inteligente.
- ☐ Controle de fonte integrado.
- ☐ Testes.

Nesta aula

Criando um projeto no Visual Studio Community 2019

Estrutura de um projeto Web API



- ☐ Setup.
- ☐ Gerenciador de Soluções.
- ☐ Controllers.
- ☐ Arquivo AppSettings.json.
- ☐ Arquivo AppSettings.Development.json.
- ☐ Classe Programa.cs.
- ☐ Classe Startup.cs.
- ☐ Método ConfigureServices().
- ☐ Método Configure().
- ☐ Classe WeatherForecast.cs.

Nesta aula

Controllers

Controllers



- ❑ As controllers em uma aplicação agrupam logicamente tipos semelhantes de ações. Esse agrupamento de ações permite definir conjuntos de regras, como roteamento, cache e autorização, que serão aplicadas coletivamente.

Controllers



- ❑ Classes Controller x ControllerBase no ASP.NET Core.
- ❑ Classe ControllerBase.
- ❑ Atributo [ApiController].

Nesta aula

Rotas

Rotas



- ❑ O roteamento é responsável por mapear as solicitações de entrada do navegador para ações específicas da controller.
- ❑ No ASP.NET Core, o roteamento é gerenciado por um middleware de roteamento, que corresponde às URLs das requisições de entrada até as Actions ou outros pontos de extremidade.

Rotas



- Roteamento Convencional.
- Roteamentos por Atributo.
- Métodos HTTP.
- Prefixos de Rota.
- Restrições de Rota.
- Inferência de Parâmetros (Binding).

Nesta aula

Action Results

Actions Result



- ❑ ASP.NET Core oferece diferentes opções para tipos de retorno das actions (método ao qual são configurados as rotas, verbos, entrada de parâmetros e a resposta (response)) nas controllers da Web API.

Actions Results

- Tipo Específico.
- Retornar IEnumerable<T> ou IEnumerableAsync<T>.
- Tipos IActionResult.
- Action Síncrona.
- Action Assíncrona.

Nesta aula

Convenções

Convenções



- ❑ ASP.NET Core inclui uma maneira de extrair documentação e aplicá-la aos vários elementos, como: actions, controllers ou todos controllers em um assembly. As convenções da Web Api são um substituto ao decorar ações individuais com **[ProducesResponseType]**.

Actions Results



- Aplicando Convenções Web API.
- Atributos para Tipos de Response.

Nesta aula

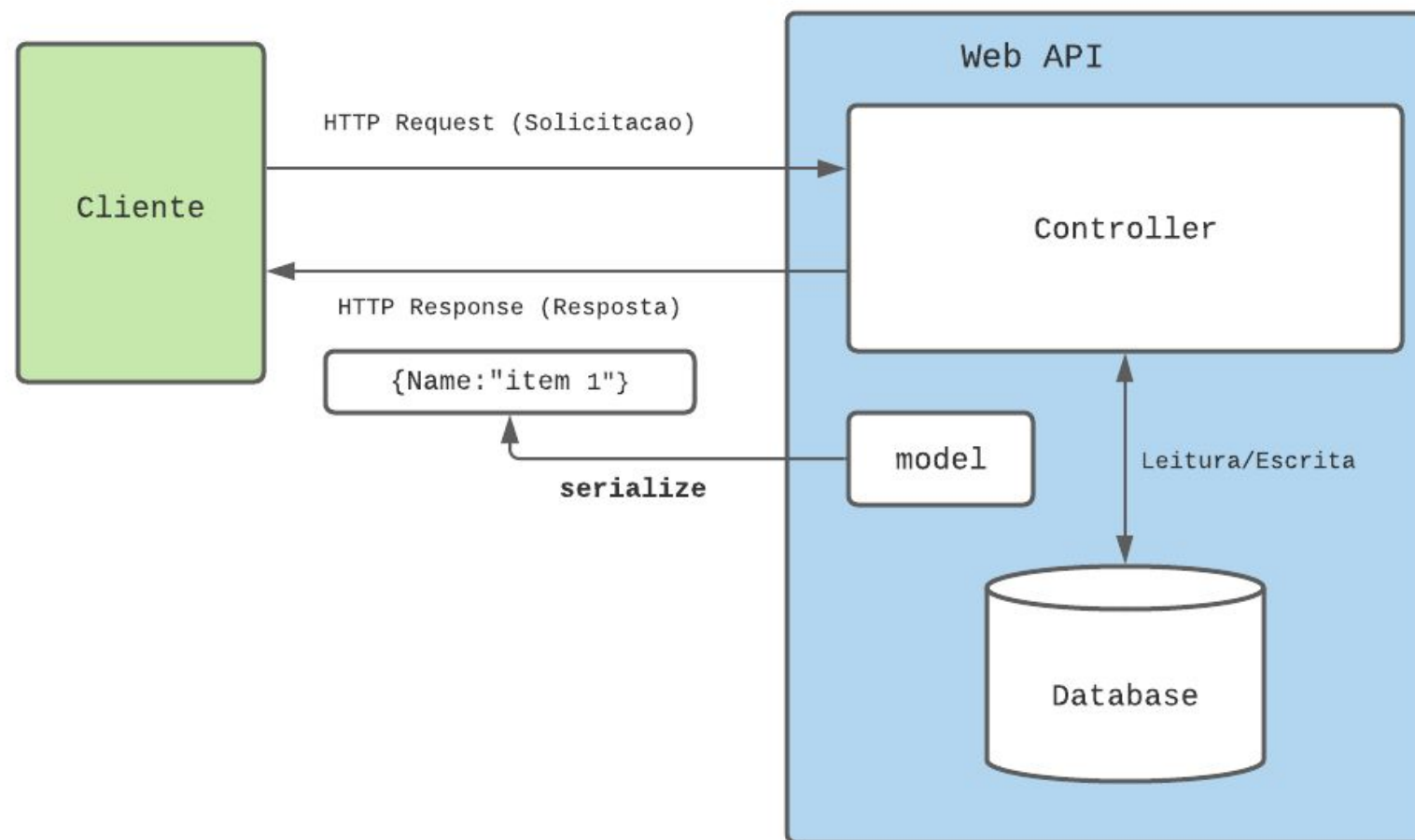
CRUD - (Criar, Ler, Atualizar e Deletar)

VERBOS HTTP/CRUD



API	Descrição	Corpo da Solicitação	Corpo da Resposta
GET /api/livros	Obter todos os livros	Nenhum	Coleção de itens de livros.
GET /api/livros/{id}	Obter um livro por ID	Nenhum	Livro
POST	Adiciona um novo livro	Um novo livro	Livro
PUT /api/livros/{id}	Atualizar um item existente	Item de tarefas pendentes	Nenhum
DELETE /api/livros/{id}	Exclui um livro	Nenhum	Nenhum

Diagrama



Projeto Web API



- ☐ Adicionar um contexto de Banco de Dados.
- ☐ Classe contexto DbContext e Geração de dados.
- ☐ Classes Startup.cs e Program.cs.
- ☐ Controller LivroController.cs.
- ☐ Método POST.
- ☐ Métodos GET.
- ☐ Método PUT.
- ☐ Método DELETE.
- ☐ Roteamento.
- ☐ Valores Retornados.
- ☐ Instale Postman.

Nesta aula

Segurança

Segurança



- ☐ Autenticação.
- ☐ Autorização.
- ☐ Autenticação vs. Autorização.
- ☐ Token.
- ☐ JWT (JSON Web Token).

Autenticação vs. Autorização



	Autenticação	Autorização
O que faz ?	Verifica a identidade com credenciais	Concede (ou nega permissão de acesso)
Como funciona?	Principalmente por meio de senhas e organização biométrica	Por meio das configurações da equipe de segurança de uma organização
É visível para o usuário?	Sim	Não
O usuário pode alterá-lo?	Possivelmente	Não
Como os dados se movem?	Por meio de tokens de identificação	Por meio de tokens de acesso

JWT - JSON Web Token



- ❑ JSON Web Token (JWT) é um padrão aberto (RFC 7519) que define uma forma compacta e independente para transmitir informações com segurança entre as partes como um objeto JSON. Essas informações podem ser verificadas e confiáveis, porque são assinadas digitalmente. Os JWTs podem ser assinados usando um segredo (com o algoritmo HMAC) ou um par de chaves pública/privada usando RSA ou ECDSA.



JWT - Formato



❑ Cabeçalho

- ❑ O cabeçalho normalmente consiste em duas partes: o tipo do token, que é JWT, e o algoritmo de assinatura que está sendo usado, como HMAC SHA256 ou RSA.

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

JWT - Formato



❑ Payload

- ❑ A segunda parte do token é a carga útil, que contém as declarações. As declarações são declarações sobre uma entidade (normalmente, o usuário) e dados adicionais. Existem três tipos de reivindicações: registradas, públicas e privadas.

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

JWT - Formato



❑ Assinatura

- ❑ Para criar a parte da assinatura, você deve pegar o cabeçalho codificado, a carga útil codificada, um segredo, o algoritmo especificado no cabeçalho e assiná-lo.

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret)
```


JWT - Resultado



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4  
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.  
4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

Imagem do site Jwt.io

JWT - Site oficial (<https://jwt.io/>)



A screenshot of the JWT.io website interface. The top navigation bar is black with the JWT logo on the left and links for 'Debugger', 'Libraries', 'Introduction', and 'Ask' in the center. On the right, it says 'Crafted by auth0'. The main area is split into two columns: 'Encoded' on the left and 'Decoded' on the right. The 'Encoded' column has a text input field containing a JWT token. The 'Decoded' column shows the token's structure: a header with 'alg' as 'HS256' and 'typ' as 'JWT', and a payload with 'sub', 'name', and 'iat' fields. Below the payload is a 'VERIFY SIGNATURE' section with a code snippet for HMACSHA256 and a checkbox for 'secret base64 encoded'. At the bottom left, a green checkmark indicates 'Signature Verified'. At the bottom right, there is a blue button labeled 'SHARE JWT'.

Imagem do site Jwt.io

Nesta aula



Documentando APIs

Swagger



- ❑ Uma API definitivamente perde seu propósito se não for acessível e se não tivermos documentação que nos ajude a entendê-la. Como eles poderiam ser acessíveis? Um dos piores problemas com as APIs é que, em muitos casos, a documentação que as acompanha é inútil. Swagger nasceu com o propósito de resolver esse problema. Seu objetivo é padronizar o vocabulário usado pelas APIs. É o dicionário da API.


Swagger



- ☐ Swagger/OpenApi.
- ☐ Especificação Swagger.
- ☐ Swagger.
- ☐ Integrando a UI do Swagger.

Swagger UI



 **Swagger**
Supported by SMARTBEAR

Select a definition My API V1

Book Store API ^{v1} OAS3

<https://localhost:44381/swagger/v1/swagger.json>

Livros

POST

/api/Livros

✓

GET

/api/Livros

✓

GET

/api/Livros/{id}

✓

PUT

/api/Livros/{id}

✓

DELETE

/api/Livros/{id}

✓

Schemas

Livro >