



**TECNOLOGIA E INOVAÇÃO
EM PROL DA INDÚSTRIA**



Curso Técnico em Informática

Desenvolvimento de Sistemas II – 180h

Profª: Francisleide Almeida

- Uma Web API é uma interface de programação de aplicações (API) tanto para um servidor quanto um navegador. É utilizada para se conseguir recuperar somente o valor necessitado num banco de dados de um site.
- Ex:
 - Cliente
 - Garçon
 - Cozinheiro

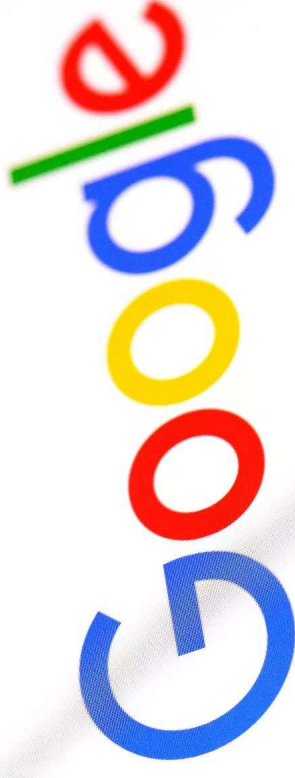
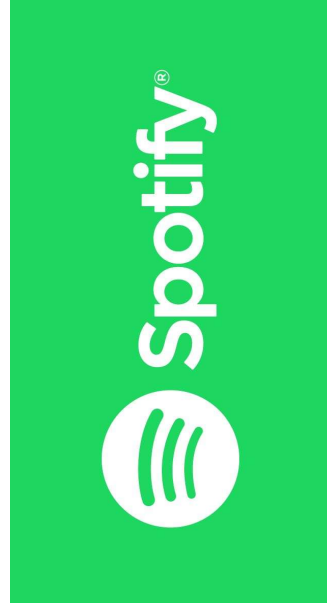


Porque expor os dados?

- Gerar inovação sobre seus próprios dados, de uma maneira que não poderia sozinho;
- Gerar desenvolvimento de sua própria plataforma em novos meios;
- Gerar mais mercado e expandir o público-alvo;
- Organização do setor de TI, em que os consumidores das APIs são as outras equipes internas da empresa, e as APIs expostas funcionam como portais de troca de dados entre setores.

Serviços existentes

- Google;
- Facebook;
- LinkedIn;
- Netflix;
- Spotify;



ASP .net WEB API

- O ASP.Net Web API utiliza HTTP com REST
- REST:
 - Representational State Transfer
 - Baseado no HTTP
 - Utiliza a semântica dos métodos do HTTP (GET, POST, PUT, DELETE)
- Utiliza JSON para retornar os dados

JSON

- JavaScript Object Notation - Notação de Objetos JavaScript;
- Formatação leve de troca de dados;
- JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras

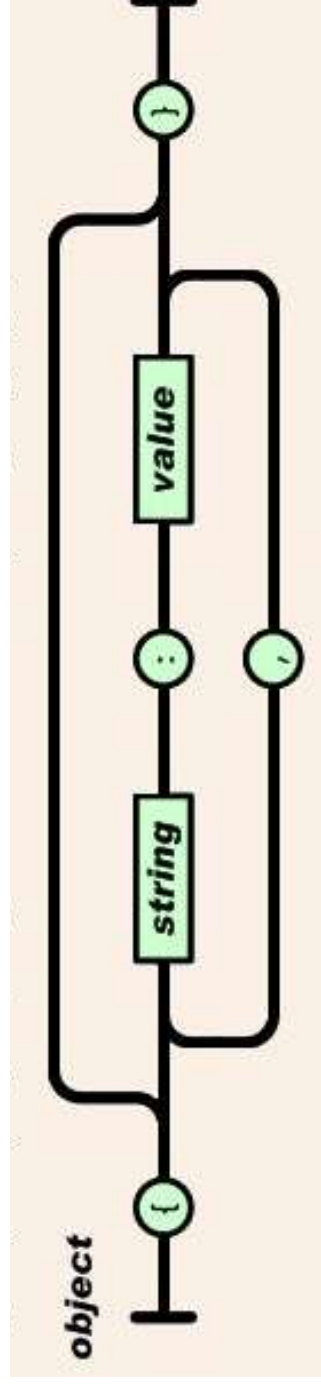
•

JSON

- JSON está constituído em duas estruturas:
 - Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um object, record, struct, dicionário, hash table, keyed list, ou arrays associativas.
 - Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma array, vetor, lista ou sequência.

JSON

- Um objeto é um conjunto desordenado de pares nome/valor. Um objeto começa com { (chave de abertura) e termina com } (chave de fechamento). Cada nome é seguido por : (dois pontos) e os pares nome/valor são seguidos por , (vírgula).



PRÁTICA

- Criar classe Cliente:
 - Id
 - Nome
 - Email
 - Ativo
- Criar um controller web api de Cliente com ações usando EF

WEB API (HTTP)

- Get = Select
- Post = Insert
- Put = Update (ou Insert)
- Delete = Delete

Através do HTTP quando for realizado um GET via URI o Web API retornará o resultado do método Get, um POST via formulário o Web API fará uma inclusão através do método Post, PUT via formulário uma alteração através do método Put e DELETE via URI uma exclusão através do método Delete.


```

public class ClientesController : ApiController
{
    private PrimeiraWebAPIContext db = new PrimeiraWebAPIContext();
    private readonly Cliente[] Clientes = new Cliente[]
    {
        new Cliente { ClienteId = 1,
            Nome = "Francislaide Almeida",
            Email = "fran@gmail.com", Ativo = true },
        new Cliente { ClienteId = 2,
            Nome = "Ana",
            Email = "ana@outlook.com", Ativo = true },
        new Cliente { ClienteId = 3,
            Nome = "Yla",
            Email = "buri@yla.com", Ativo = false }
    };

    // GET: api/Clientes
    public Cliente[] GetClientes()
    {
        return Clientes;
    }
}

```

Para pegar os dados

O serviço funcionou e respondeu sendo chamado pelo browser, o retorno foi em XML, pois o browser não interpreta JSON, por padrão o Web API está configurado para responder JSON sempre que possível.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<ArrayOfCliente xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/08/Client">
  <Cliente>
    <Ativo>true</Ativo>
    <ClienteId>1</ClienteId>
    <Email>fran@gmail.com</Email>
    <Nome>Francisleide Almeida</Nome>
  </Cliente>
  <Cliente>
    <Ativo>true</Ativo>
    <ClienteId>2</ClienteId>
    <Email>ana@outlook.com</Email>
    <Nome>Ana</Nome>
  </Cliente>
  <Cliente>
    <Ativo>false</Ativo>
    <ClienteId>3</ClienteId>
    <Email>buri@yla.com</Email>
    <Nome>Yla</Nome>
  </Cliente>
</ArrayOfCliente>
```

JSON

Para gerar um Json e renderizar na tela:

```
// GET: api/Pessoas
public HttpResponseMessage GetPessoas()
{
    return new HttpResponseMessage()
    {
        Content = new
        StringContent(JArray.FromObject(db.Pessoas).ToString(), Encoding.UTF8,
        "application/json")
    };
}
```