



Você está aqui: [Configuração e manutenção](#) > [Integração de dados](#) > [APIs do Archer](#) > Archer RESTful API

Archer RESTful API

Para obter informações sobre a API Archer RESTful, consulte a ajuda da API Archer na [Central de ajuda do Archer](#).

Nesta página

- [Visão geral da API RESTful](#)
- [Exemplo: resposta JSON](#)
- [Critérios do serviço RESTful](#)

Visão geral da API RESTful

O serviço RESTful API é um conjunto de recursos organizados em segmentos funcionais que são acessados por meio de controladores. Pode-se agir sobre cada recurso individualmente, por chave/ID ou em lote. Conjuntos são usados amplamente para enviar solicitações e construir respostas.

A API RESTful usa o formato JSON (JavaScript Object Notation) por padrão em solicitações e respostas, mas também aceita XML. Depois que um recurso é identificado, é possível realizar as operações criar, ler e excluir em relação a ele usando verbos HTTP padrão para indicar qual ação deve ser usada.

Os exemplos de resposta são limitados, pois as respostas variam de acordo com as solicitações de origem.

Exemplo: resposta JSON

Os resultados de resposta variam com base na solicitação. Algumas respostas podem ser muito complexas com base na chamada de origem, por exemplo, Obter todos os aplicativos.

O exemplo a seguir mostra a resposta JSON para uma solicitação de uma lista de aplicativos apenas por nome e descrição:

```
[{
  "Links": [],
  "RequestedObject": {
    "Name": "Technologies",
    "KeepLicensed": false,
    "IsDeprecated": false,
    "Description": "<html><head><style type=\"text/css\">.c0 { font-
family: 'Arial' } .c1 { margin: 0px 0px 13px } </style></head><p
class=\"c1\">The Technologies application provides a searchable and
extensible repository of technology version information that can be
leveraged to relate objects of like technology.</p></html>"
  },
  "IsSuccessful": true,
  "ValidationMessages": []
},
{
  "Links": [],
  "RequestedObject": {
    "Name": "Test Application 10",
    "Description": null
  },
  "IsSuccessful": true,
  "ValidationMessages": []
},
{
  "Links": [],
  "RequestedObject": {
    "Name": "Test Application",
```

"Description": "This is my description"

},

"IsSuccessful": true,

"ValidationMessages": []

},

{

"Links": [],

"RequestedObject": {

"Name": "Test Application 2",

"Description": "This is my description"

},

"IsSuccessful": true,

"ValidationMessages": []

},

{

"Links": [],

"RequestedObject": {

"Name": "Test Application 3",

"Description": "This is my description"

},

"IsSuccessful": true,

"ValidationMessages": []

},

{

"Links": [],

"RequestedObject": {

```
"Name": "Mobile Packaging Application",  
"Description": "This is my description"  
},  
"IsSuccessful": true,  
"ValidationMessages": []  
}]
```

Critérios do serviço RESTful

A tabela a seguir descreve como o Archer implementou serviços RESTful para o aplicativo móvel.

Critérios	Implementação
Identificação de recursos	Em vez de expor métodos que podem ser chamados, o Archer expõe os recursos que podem ser recuperados. Por exemplo, em vez de listar métodos que têm nomes com verbos, como UpdateQuestionnaire, o Archer expõe questionários como recursos com os meios de interagir com eles usando verbos HTTP padrão, como GET e POST.
Manipulação de recursos por meio de representações	<p>Em vez de ter um serviço que retorna objetos fortemente tipados (objetos C# com todas as propriedades esperadas), o Archer retorna representações desses recursos. O chamador determina a forma que essas representações assumem.</p> <ul style="list-style-type: none">• Se o chamador quiser uma resposta em JSON, o cabeçalho Accept deverá conter application/json.• Se o chamador quiser uma resposta em XML, o cabeçalho Accept deverá conter application/xml.

Cr�terios	Implementa��o
Mensagens autodescritivas	<p>Em vez de contar com o significado do nome do m�todo para o Archer descobrir como o m�todo interage com o recurso, s�o usados verbos HTTP. Cada mensagem descreve a si mesma.</p> <ul style="list-style-type: none"> • Se o recurso � acessado em <code>/api/core/application</code> e o m�todo da solicita��o � POST (POST to <code>/api/core/application/vendors</code>), o usu�rio est� tentando salvar o novo aplicativo chamado Fornecedores. • Se a solicita��o � uma GET to <code>/api/core/application/vendors</code>, o usu�rio solicita fornecedores de aplicativo.
Hiper�mia como o mecanismo de estado do aplicativo	<p>Em vez de conhecer a API que interage com os m�todos de chamada a RPC, uma API RESTful fornece o URI raiz e um ponto de partida para interagir com o recurso. Cada resposta cont�m os links que ajudam a percorrer o restante da API.</p> <p>Por exemplo, quando o usu�rio envia um GET to <code>/api/core/application/vendors</code>, a resposta inclui um link que mostra como atualizar esse aplicativo (PUT <code>/api/core/application/vendors</code>) e como excluir o aplicativo (DELETE <code>/api/core/application/vendors</code>).</p> <p>A hiper�mia (links) atuam como o mecanismo de estado do aplicativo (movendo o chamador pelos diferentes estados do aplicativo).</p>