
Fecomércio RS
Senac

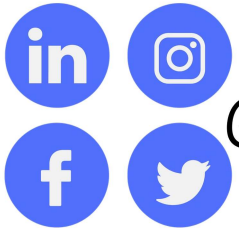
Desenvolvimento de Serviços e API II -
Aula14

Segurança e Autenticação nos Web Services RESTful

(JWT – JSON Web Token)

Prof. MSc. Adalto Selau Sparremberger

assparremberger@senacrs.com.br


@adaltoss


/assparremberger

Fecomércio RS | Senac

JWT – JSON Web Token

- JWT são um método padrão aberto do setor da industria (RFC 7519), para representar requisições de forma segura entre duas partes.
- Define uma maneira compacta e independente de 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 senha (com o algoritmo HMAC) ou um par de chaves pública/privada usando RSA ou ECDSA.

JWT – JSON Web Token

- JWTs possam ser criptografadas para também fornecer sigilo entre as partes.
- Os tokens assinados podem verificar a integridade das requisições contidas nele, enquanto os tokens criptografados ocultam essas requisições de outras partes.
- Quando os tokens são assinados usando pares de chaves pública/privada, a assinatura também certifica que apenas a parte que detém a chave privada é a pessoa que a assinou.

Quando devemos usar JSON Web Tokens?

- Alguns cenários em que JSON Web Tokens são úteis:
  - Autorização: este é o cenário mais comum para o uso do JWT.
  - Após o login do usuário, cada solicitação subsequente incluirá o JWT, permitindo que o usuário acesse rotas, serviços e recursos permitidos com esse token.
  - O Logon único é um recurso que usa amplamente o JWT atualmente, devido à sua pequena sobrecarga e à capacidade de ser facilmente usado em diferentes domínios.

Quando devemos usar JSON Web Tokens?

- Alguns cenários em que JSON Web Tokens são úteis:
  - Troca de informações: os JSON Web Tokens são uma boa maneira de transmitir informações com segurança entre as partes.
  - Como as JWTs podem ser assinadas (por exemplo, usando pares de chaves públicas/privadas), você pode ter certeza de que os remetentes são quem eles dizem que são. Além disso, como a assinatura é calculada usando o cabeçalho e a carga, você também pode verificar se o conteúdo não foi violado.

Estrutura JSON Web Token?

- Em sua forma compacta, os JSON Web Tokens consistem em três partes separadas por pontos (.), que são:
  - Header
  - Payload
  - Signature

- Ex: xxxxx.yyyyy.zzzzz

Estrutura JSON Web Token?

- **Header**
  - O cabeçalho geralmente 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.
  - Ex:

Fonte: https://jwt.io/

  - Em seguida, esse JSON é codificado em Base64Url para formar a primeira parte do JWT.

Estrutura JSON Web Token?

- **Payload**
  - A segunda parte do token é a carga útil, que contém as *claims* (propriedades no JSON). *Claims* são declarações sobre uma entidade (normalmente, o usuário) e dados adicionais.
  - Existem três tipos de *claims*: registradas, públicas e privadas.

Estrutura JSON Web Token?

- **Payload**
  - **Claims registradas**: são um conjunto de propriedades predefinidas que não são obrigatórias, mas recomendadas, para fornecer um conjunto de informações úteis e interoperáveis.
    - Alguns deles são: **iss** (emissor), **exp** (tempo de expiração), **sub** (assunto), **aud** (público) e outros.
    - Observe que os nomes das *claims* têm apenas três caracteres, já que o JWT deve ser compacto.

Estrutura JSON Web Token?

- **Payload**
  - **Claims públicas**: elas podem ser definidas à vontade por aqueles que usam JWTs.
    - Mas, para evitar colisões, elas devem ser definidas no Registro de token da IANA JSON Web Token Registry ou definidas como um URI que contém um espaço para nome resistente a colisões.
  - **Claims privadas:** são as propriedades personalizadas, criadas para compartilhar informações entre as partes que concordam em usá-las e não são propriedades registradas ou públicas.

Estrutura JSON Web Token?

- **Signature**
  - Para criar a parte da assinatura, é necessário pegar o cabeçalho codificado, a carga útil codificada, um segredo, o algoritmo especificado no cabeçalho e assinar isso.
  - Por exemplo, se você deseja usar o algoritmo HMAC SHA256, a assinatura será criada da seguinte maneira:
  - EX:

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

Fonte: https://jwt.io/

  - A assinatura é usada para verificar se a mensagem não foi alterada ao longo do caminho e, no caso de tokens assinados com uma chave privada, também pode verificar se o remetente da JWT é quem diz ser.

Estrutura JSON Web Token?

- **Juntando tudo**
  - A saída são três sequências de caracteres Base64-URL separadas por pontos que podem ser passadas facilmente nos ambientes HTML e HTTP, além de serem mais compactas quando comparadas aos padrões baseados em XML, como SAML.
  - A seguir, é mostrado um JWT com o cabeçalho e a carga útil anteriores codificados e assinado com um segredo.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaXNTb2NpYWwiOnRydWV9.
4pcPyMD09olPSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

Fonte: https://jwt.io/

Como os JSON Web Tokens funcionam?

- Na autenticação, quando o usuário efetua login usando suas credenciais, um JSON Web Token será retornado. Como os tokens são credenciais, deve-se tomar muito cuidado para evitar problemas de segurança. Em geral, você não deve manter os tokens por mais tempo do que o necessário.

- Você também não deve armazenar dados de sessão confidenciais no armazenamento do navegador devido à falta de segurança.

Como os JSON Web Tokens funcionam?

- Sempre que o usuário deseja acessar uma rota ou recurso protegido, o agente do usuário deve enviar o JWT, normalmente no cabeçalho de Autorização, usando o esquema Portador. O conteúdo do cabeçalho deve ter a seguinte aparência:

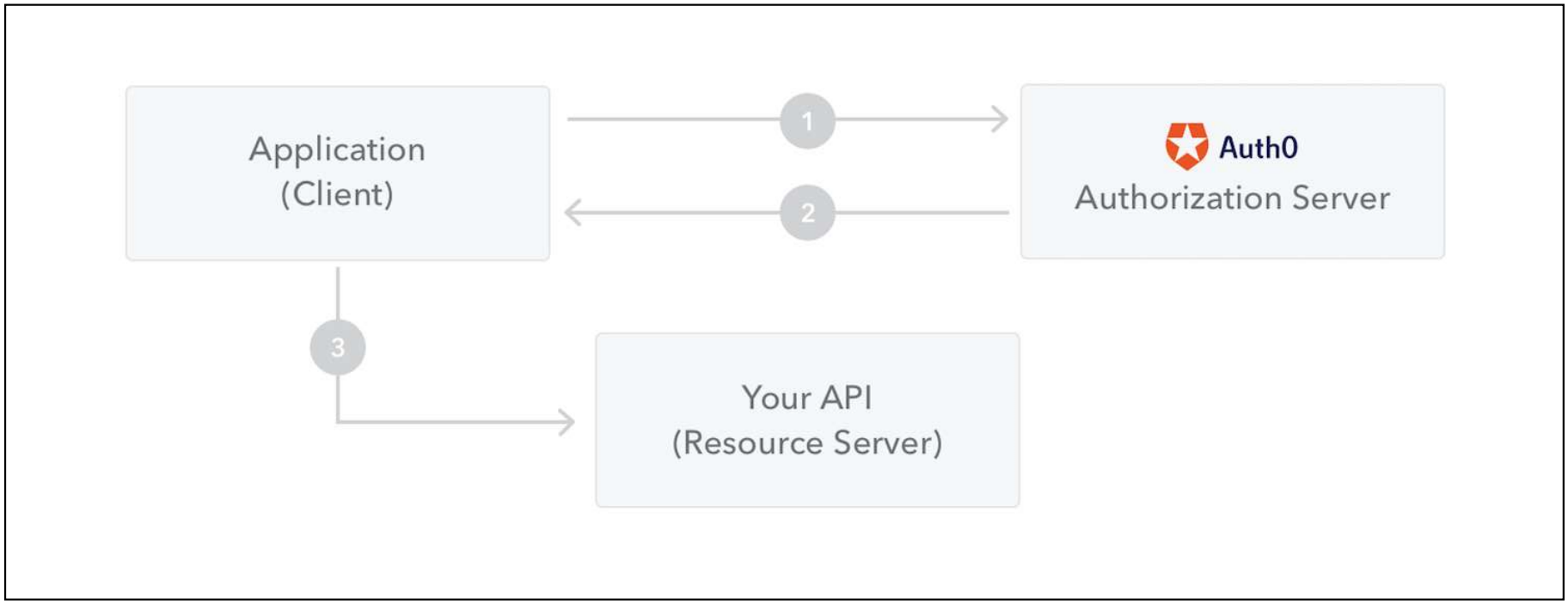
```
Authorization: Bearer <token>
```

Como os JSON Web Tokens funcionam?

- Isso pode ser, em certos casos, um mecanismo de autorização sem estado. As rotas protegidas do servidor procurarão um JWT válido no cabeçalho da autorização e, se houver, o usuário poderá acessar os recursos protegidos. Se o JWT contiver os dados necessários, a necessidade de consultar o banco de dados para determinadas operações poderá ser reduzida, embora isso nem sempre seja o caso.
- Se o token for enviado no cabeçalho da autorização, o CORS (compartilhamento de recursos de origem cruzada) não será um problema, pois não usa cookies.
- O diagrama a seguir mostra como um JWT é obtido e usado para acessar APIs ou recursos:

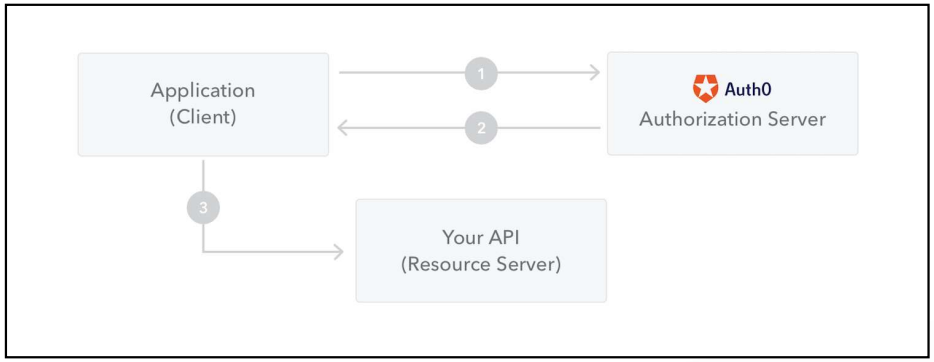
Como os JSON Web Tokens funcionam?

- O diagrama a seguir mostra como um JWT é obtido e usado para acessar APIs ou recursos:



Como os JSON Web Tokens funcionam?

1. O aplicativo ou cliente solicita autorização ao servidor de autorização. Isso é realizado através de um dos diferentes fluxos de autorização. Por exemplo, um aplicativo Web compatível com OpenID Connect típico passará pelo terminal / oauth / authorize usando o fluxo de código de autorização.
2. Quando a autorização é concedida, o servidor de autorização retorna um token de acesso ao aplicativo.
3. O aplicativo usa o token de acesso para acessar um recurso protegido (como uma API).



Observe que, com os tokens assinados, todas as informações contidas no token são expostas a usuários ou outras partes, mesmo que não possam alterá-lo. Isso significa que você não deve colocar informações secretas no token.

Por que devemos usar JSON Web Tokens?

- Benefícios do JSON Web Tokens (JWT) quando comparado ao Simple Web Tokens (SWT) e ao Security Assertion Markup Language Tokens (SAML):
  - Como o JSON é menos detalhado que o XML, quando é codificado, seu tamanho também é menor, tornando o JWT mais compacto que o SAML. Isso faz do JWT uma boa opção para ser transmitida nos ambientes HTML e HTTP.
  - Em termos de segurança, o SWT pode ser assinado simetricamente apenas por um segredo compartilhado usando o algoritmo HMAC. No entanto, os tokens JWT e SAML podem usar um par de chaves pública/privada na forma de um certificado X.509 para assinatura. Assinar XML com assinatura digital XML sem introduzir falhas de segurança obscuras é muito difícil quando comparado à simplicidade de assinar JSON.

Por que devemos usar JSON Web Tokens?

- Benefícios do JSON Web Tokens (JWT) quando comparado ao Simple Web Tokens (SWT) e ao Security Assertion Markup Language Tokens (SAML):
  - Os analisadores JSON são comuns na maioria das linguagens de programação porque são mapeados diretamente para objetos. Por outro lado, o XML não possui um mapeamento natural de documento para objeto. Isso facilita o trabalho com JWT do que com asserções SAML.
  - Em relação ao uso, o JWT é usado em escala da Internet. Isso destaca a facilidade de processamento do token JSON Web no lado do cliente em várias plataformas, especialmente móveis.

Vídeos resumindo JWT

- https://www.youtube.com/watch?v=Gyq-yeot8qM
- https://www.youtube.com/watch?v=KFNGgc34UXE

Obs: copie o link e cole no navegador

Desafio

- Construir uma implementação simples de JWT.

Referências

- https://jwt.io/
- https://blog.codeexpertslearning.com.br/conhecendo-o-jwt-114afbfcad95


Fecomércio RS
Senac