JSON WEB TOKEN

Atul

IDENTIFICATION OF THE NEED

WE NEED SCALABLE APPLICATIONS

- Applications have become complex and need third party services
- Explosion of APIs
  - Mobile and Single Page Applications
  - Applications have started relying more on APIs- for example, Google's map API
    - The value of Facebook
      - Social Network for a layman
      - Platform for applications- use APIs to access data
    - The App Store
    - AWS
- Need scalable web apps with better Authentication mechanism

DEFINITION OF THE PROBLEM

HTTP SESSION

- HTTP being stateless, every request needs to be authenticated
- Server sessions
- Problems
  - Memory or disk
  - Scalability- Since stored in memory, this provides problems with scalability. Also vital information in memory will limit our ability to scale
  - CORS- Cross Origin Resource Sharing When accessing resources from another domain
  - CSRF- Cross-Site Request Forging

WE NEED BETTER AUTHENTICATION MECHANISM

- Token based authentication
- Tokens enable
  - Statelessness
    - no user information is stored on the server
    - Scalable servers as they are stateless
    - Passing authentication to other applications
- A token need to be a signed one
- A token(authentication mechanism) is not a cookie(storage mechanism)

TOKENS

- Every request requires the token.
- Sent in the HTTP header
- We will also need to set our server to accept requests from all domains using Access-Control-Allow-Origin: *
  - Designating * in the ACAO header does not allow requests to supply credentials like HTTP authentication, client-side SSL certificates, or cookies
- Third party access

WHAT IS JWT

WHAT IS A JSON WEB TOKEN

- It is a string.
- Here is an example token taken from https://jwt.io/

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

WHAT ELSE IS IT

- JWT is an open standard(RFC 7519)
- It is a way for transmitting information between parties as a JSON object
- Compact
  - can be sent through an URL, POST parameter, or inside an HTTP header
  - transmission is fast
- Self-contained:
  - The payload contains all the required information about the user
  - Verified and trusted
  - because it is digitally signed
  - using a secret (with the **HMAC** algorithm) or a public/private key pair using **RSA**.

JWT APPLICABILITY

- Authentication:
  - After login, each subsequent request includes the JWT
  - Single Sign On
  - Small footprint
  - Ease of use across domains
- Information Exchange:
  - Secure
  - Signature is calculated using the header and the payload
  - Verify that the content hasn't been tampered with

JWT STRUCTURE

- A JWT consist of three parts each separated by a dot
  - Header
  - Payload
  - Signature

JWT HEADER

- { "alg": "HS256", "typ": "JWT" }
- Algorithm and type of token

JWT PAYLOAD

- { "sub": "1234567890", "name": "John Doe", "admin": true }
- Claims
  - statements about an entity (typically, the user)
  - additional metadata
  - Three types of claims: *reserved*, *public*, and *private*
  - Reserved- predefined, recommended- 3 letter
    - **iss**(issuer), **exp**(expiration time), **sub**(subject), **aud**(audience)
- Public- defined by those using JWTs
  - To avoid collisions
    - define in the IANA JSON Web Token Registry
    - define as a URI that contains a collision resistant namespace.
- Private claims- Custom claims
  - created to share information between parties that agree on using them.

JWT SIGNATURE

- To create the signature
  - Take the encoded header, the encoded payload, a secret, the algorithm specified in the header, and sign that
  - For example if you want to use the HMAC SHA256 algorithm, the signature will be created in the following way:

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

TOKEN

- The output is
  - three Base64 strings (header, payload, signature)
  - separated by single dots
- The following is a JWT that has the previous header and payload encoded, and it is signed with a secret.(https://jwt.io/)

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

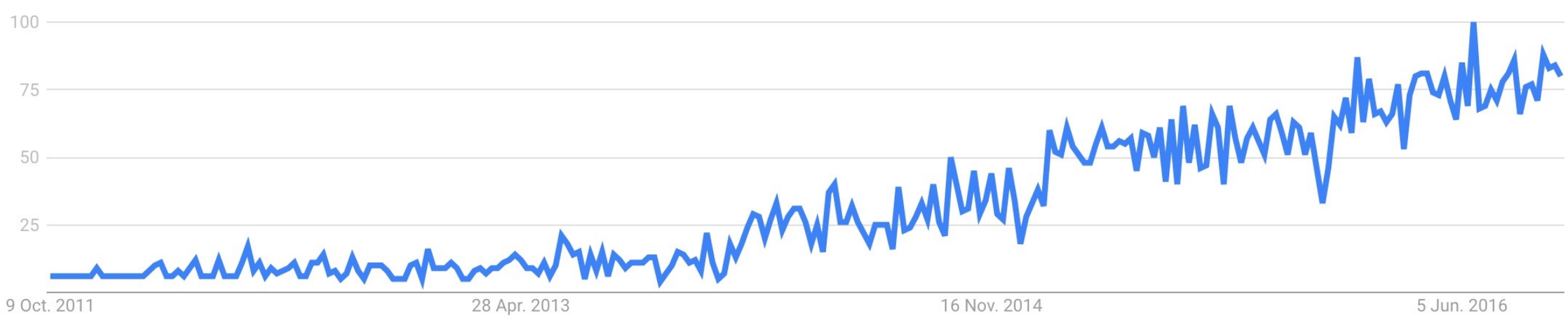
JWT TRENDS

json web token
Search term

+ Compare

Worldwide | Past 5 years | All categories | Web Search

Interest over time



JWT ADOPTION

- JSON
- Simple
- Support from languages- libraries
- Symmetric and asymmetric crypto
- 3k+ github repositories
- 8k+ SO threads

JAVA AND JWT

- http://stackoverflow.com/questions/23808460/jwt-json-web-token-library-for-java
- https://jwt.io/ lists a few libraries

JWT AND OAUTH

- JWT:
  - Simple with faster development time
  - More mobile friendly
  - Needs expiration header as once issued token cannot be revoked
    - Or have to maintain server side storage for revocation before expiry
  - More of a protocol
- OAuth:
  - More of an delegated authentication framework
  - Dependency on Auth Provider (Login with Google, Facebook, etc.)
  - Can introduce provider specific code