



# AngularJS

*Jour 5 - Firebase & pratique*

# Au programme

- Firebase : un serveur pour front !  
*Real time, intégration AngularJS, Pas de maintenance server, synchro offline*
- La pratique

# 1 - Firebase + AngularJS

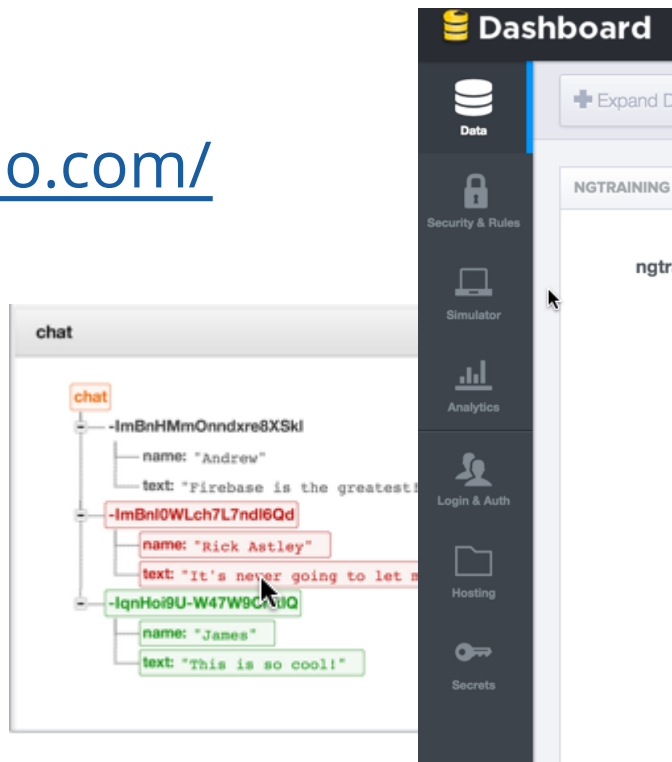


# Firebase + AngularJS

- Firebase
  - Solution SaaS : <https://www.firebase.com/>
  - Fonctionnalités :
    - Authentification, persistance des données, temps réel, etc.
  - DEMO :
    - [http://firehist.github.io/angular-training/TP/project\\_02](http://firehist.github.io/angular-training/TP/project_02)
    - <https://goo.gl/if0jp6>

# Firestore - La forge !

- IHM user-friendly !
  - <https://ng-training.firebaseio.com/>
- Importer / Exporter des data
- Consulter les métriques d'utilisation
- Configurer les paramètres de sécurité



# Firestore - Les données

## - Données hiérarchique

[https://ng-training.firebaseio.com/chat\\_01](https://ng-training.firebaseio.com/chat_01)

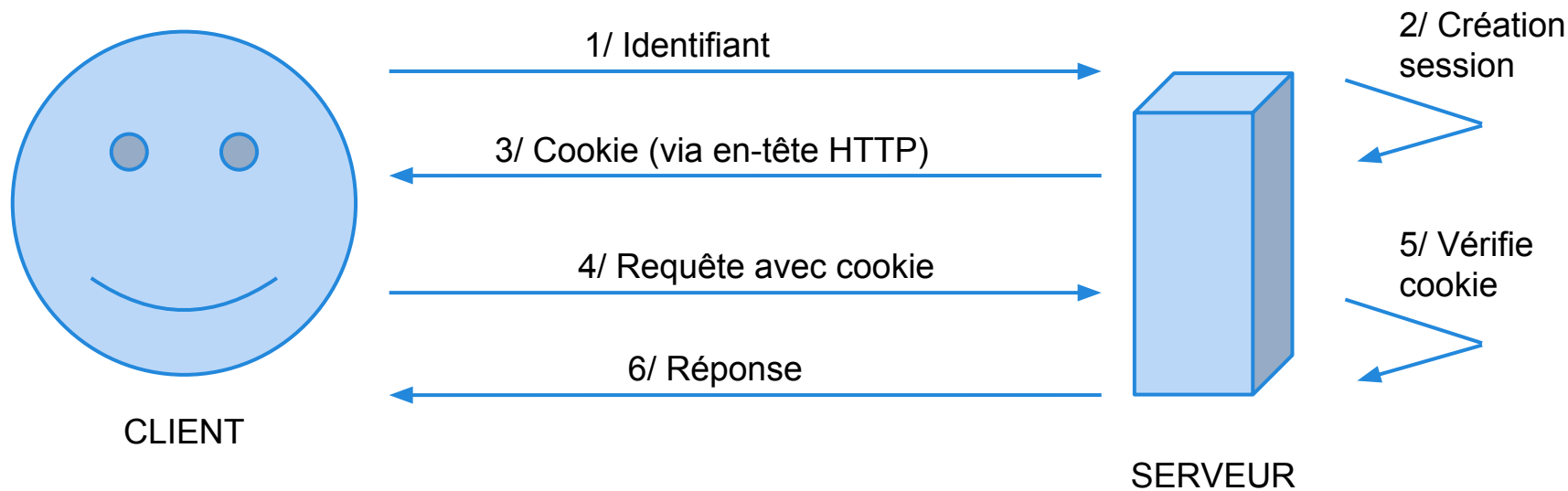


[https://ngtraining.firebaseio.com/chat\\_01/-JsGl00-OGFnP\\_l0bsoM/name](https://ngtraining.firebaseio.com/chat_01/-JsGl00-OGFnP_l0bsoM/name)

# Firebase - L'authentification

- Complicqué !
  - Nécessite son propre serveur
  - Problème de sécurité
  - Synchronisation hors-ligne
  - Intégration réseaux sociaux

# Firestore - Authentication via cookie





# Firestore - Authentication via cookie

- Nécessite :
  - des serveurs toujours actifs
  - que les serveurs se connaissent
  - Restreint à un domaine
- Assujetti au faille CSRF
- Pas de couverture complète sur mobile

# Firebase - Authentication via JWT

- JWT : JSON Web-Token (<http://jwt.io/>)

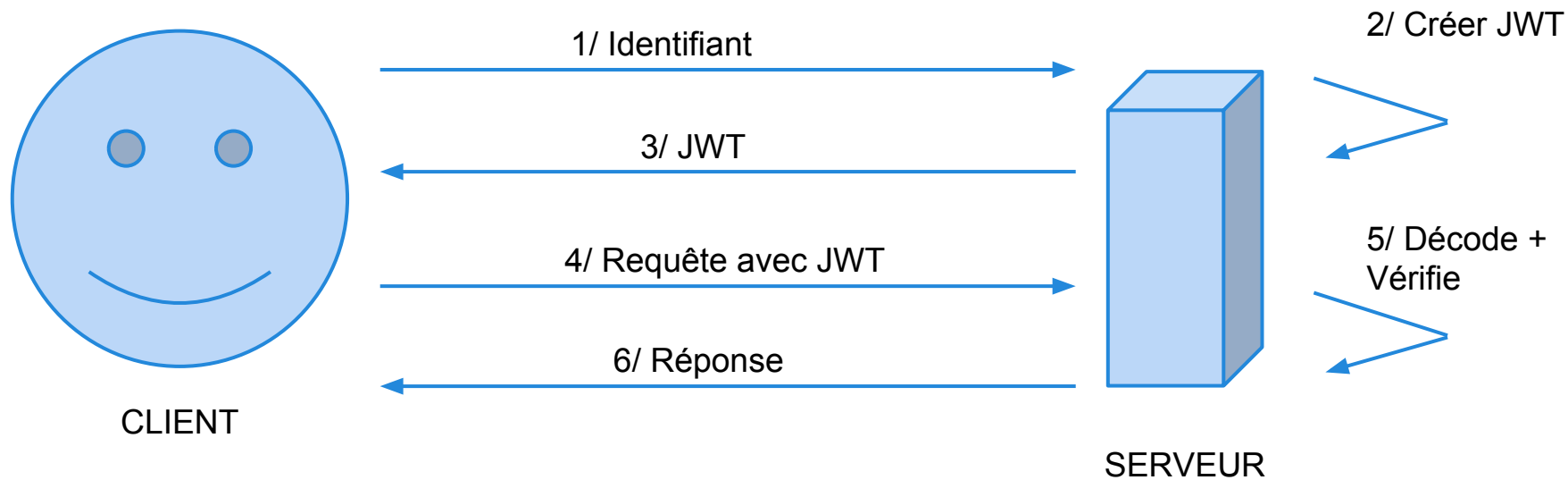
```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6I  
kpvaG4gRG9lIiwiaWQiYW4iOi0nRydWV9.TJVA9  
50rM7E2cBab30RMHrHDcEfxjoYZgeFONFh7Hg  
Q
```

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

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

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret  
) ☐ secret base64 encoded
```

# Firebase - Authentification via JWT

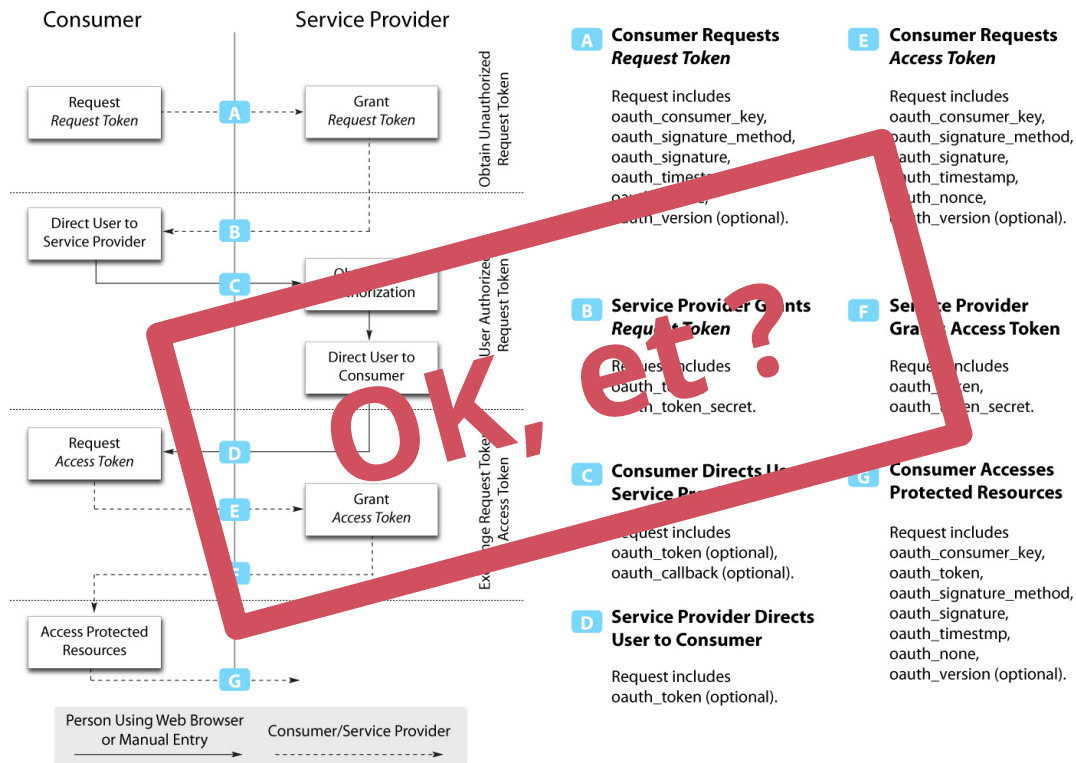


# Firestore - Authentication via JWT

- Mieux que les cookies
  - Mobile compliant : localStorage contre cookie
  - Serveur ne doivent partager que le secret
  - L'état est côté client, non côté serveur
  - Marche sur plusieurs domaine
- Attention
  - Faille XSS (ou cross-site scripting)

# Firestore - OAuth

## OAuth Authentication Flow



# Firebase - Authentication

- JWT
- Pas besoin de serveur
- Beaucoup de type d'authentification
  - anonyme, email, OAuth, etc.
- Fonctionne hors-ligne

# Firebase - Authentication utilisateur

- Authentication Email / Password

```
1  var ref = new Firebase("https://myfirebase.url");
2  ref.authWithPassword({
3      email: "john@doo.com",
4      password: "monpassword"
5  }, function (error, authData) {
6      // Utilisateur authentifié
7  });
```

# Firestore - Authentification utilisateur

- Détecter le changement d'état

```
1  var ref = new Firebase("https://myfirebase.url");
2  ref.onAuth(function (authData) {
3      if (authData) {
4          // Utilisateur authentifié
5      } else {
6          // Utilisateur non connecté
7      }
8  });
```



# Firebase - Authentification utilisateur

- Créer un utilisateur

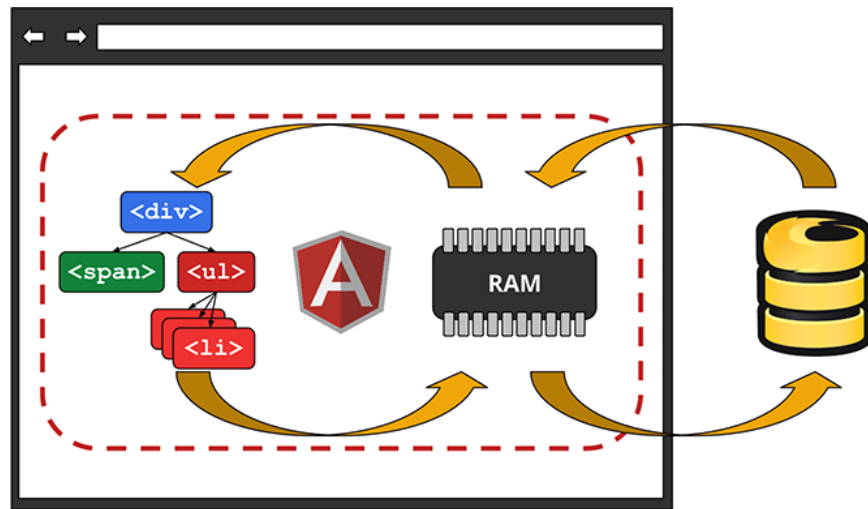
```
1  var ref = new Firebase("https://myfirebase.url");
2  ref.createUser({
3      email: "john@doo.com",
4      password: "monpassword"
5  }, function (error, authData) {
6      // Utilisateur créé
7  })
```

# Firestore - Offline

- Automatique (ou presque :D)
  - Possibilité de connaître l'état de connection
  - De connaître les latences
- Firestore Offline Capabilities
  - <https://www.firebase.com/docs/web/guide/offline-capabilities.html>

# Firebase - AngularFire

- Firebase + AngularJS = angularfire
  - Homepage : <https://goo.gl/JFiYp6>
  - Doc : <https://goo.gl/z3J6fN>
- S'occupe :
  - du data-binding
  - de l'authentification



# Firestore - AngularFire

- `$firebaseArray`
  - Synchronise les collections (`Array<any>`)
- `$firebaseObject`
  - Synchronise les objets
- `$firebaseAuth`
  - Authentification
  - Gestion des utilisateurs
  - Routing

# Firestore - Live coding

- Live coding
  - App simple
  - Authentification avec **GitHub**
    - Récupérer repository
    - Création chat par repository
  - [http://localhost:8080/day\\_05/step\\_01](http://localhost:8080/day_05/step_01)

## 2 - La pratique

# La pratique

- Reprendre le TP précédent
  - Rajouter un stockage du "MOCK" sur firebase
  - Rajouter une fonctionnalité temps réel avec angularfire
    - Chat ?
    - CRUD
    - etc.