

Projet de T'Chat



Figure 1: Cutie Cutie <3 !

Le projet est une **application Web de T'Chat**, les groupes seront définis *aléatoirement*, vous allez utiliser [Firebase](#) (une base de données à distance basée sur la technologie AJAX) et certains Préprocesseurs CSS : [Less](#) & [Sass](#) !

Ce dont vous avez besoin, c'est ce que vous avez déjà appris : HTML, CSS, Bootstrap, JS et jQuery !

Le premier jour, vous aller vous concentrer sur l'interface et l'expérience utilisateur de l'application, et écrire un cahier des charges / spécifications fonctionnelles (quelques outils pour créer vos mockups <http://mashable.com/2012/06/07/mockup-tools/>).

Et enfin, vous aurez la liberté totale d'utiliser des plugins ou frameworks (ex : [Bootstrap](#) ou [Foundation](#) ou [Material Design Light](#)).

Spécifications fonctionnelles

- [Version française](#)
- [Version anglaise](#)

Caractéristiques

C'est un t'chat, vous devez bien sûr laisser les gens discuter ensemble :

- Tout d'abord créer un grand canal global, puis autoriser d'autres canaux comme des discussions privées;
- Permettre aux utilisateurs de choisir un pseudo et peut-être de définir un avatar (vous pouvez utiliser [Gravatar](#)) ;
- Que les gens puissent s'envoyer des *Smileys Emoji, Liens, Photos, Vidéos Youtubes, Tags d'utilisateurs*, etc.. (vous devrez utiliser les **REGEX**) ;
- Rendre tout cela joli (avec plein de **CSS**) et ne pas oublier que l'application doit être responsive, pensez à la façon dont elle s'affichera sur les appareils mobiles !

Conception

Voici quelques conseils et astuces pour vous aider à travailler ensemble sur le même code.

Tout d'abord travailler avec toute l'équipe sur la conception globale de l'application, puis séparer l'équipe, entre par exemple des développeurs Backend et Frontend, une bonne idée est d'utiliser des fonctions de démonstration, par exemple :

```
/* DEMO FUNCTIONS */

var sendMessage = function (message) {
  console.log("DEMO: sendMessage: " + message);
};

var retrieveMessages = function () {
  var messages = [
    { pseudo: "Roméo", message: "Ô Roméo ! Roméo ! pourquoi es-tu Roméo ? Renie ton père !" },
    { pseudo: "Juliette", message: "Dois-je l'écouter encore ou lui répondre ?" }
  ];
  console.log("DEMO: retrieveMessages : " + messages);
  return messages;
};
```

Le but est de commencer à coder l'interface et de tester l'utilisation de ces fonctions et **dans le même temps** un autre membre de l'équipe peut travailler sur les fonctions pour les faire fonctionner avec des données réelles.

Voici le code d'exemple Javascript donné sur le site officiel de **Firebase** :

```

// Créé une connexion à votre base de données Firebase
var ref = new Firebase("https://<YOUR-FIREBASE-APP>.firebaseio.com");
// Enregistre les données
ref.set({ name: "Alex Wolfe" });
// Écoute les changements en temps réel
ref.on("value", function(data) {
    var name = data.val().name;
    alert("My name is " + name);
});

```

Firebase est conçu avec un système de mise à jour en temps réel, de sorte que vous pouvez remplacer la fonction `retrieveMessages` par juste une fonction `updateConversation` qui mettra à jour le contenu de la conversation et sera appelé dans la méthode `ref.on`, et bien sûr `sendMessage` utilisera `ref.set`.

Ressources

REGEX (Expressions Rationnelles)

- https://fr.wikipedia.org/wiki/Expression_rationnelle
- Vous pouvez les tester en ligne avec <https://regex101.com/> & <http://regexr.com/>
- Et pratiquer un peu ici: <https://www.hackerrank.com/domains/regex/>

Par exemple : `:smile_cat:` va être remplacé par ``.

AJAX

- [https://fr.wikipedia.org/wiki/Ajax_\(informatique\)](https://fr.wikipedia.org/wiki/Ajax_(informatique))
- Vous pouvez lire les pages de la documentation officielle jQuery <https://api.jquery.com/jquery.get/> & <http://api.jquery.com/jquery.ajax/>
- Et un peu de OpenClassrooms <https://openclassrooms.com/courses/simplifiez-vos-developpements-javascript-avec-jquery/premiers-pas-avec-ajax> & <https://openclassrooms.com/courses/un-site-web-dynamique-avec-jquery/le-fonctionnement-de-ajax>