

TP 5: Programmation Web Répartie

Un rendu de TP est à envoyer à pablo.oliveira@uvsq.fr avant le 20/03/2016 23h59 avec pour sujet “[WEB] TP5 votre-nom”. Le rendu contient le code que vous aurez écrit sous la forme d’une archive compressée aux formats `zip` ou `tar.gz`.

L’IDE Cloud9

Dans ce cours nous allons utiliser un environnement de développement (IDE) *dans le cloud*. Même si ce TD pourrait être développé entièrement en local (il suffit d’un browser et d’un éditeur de texte), nous allons en profiter pour nous familiariser avec Cloud9.

Important : vous devez avoir un **browser récent** pour pouvoir utiliser Cloud9, de préférence la **dernière version** de Firefox, Chrome ou Safari. Contrôlez votre version avant de commencer.

Dirigez votre browser sur <http://c9.io/>, créez un compte si ce n’est pas déjà fait, et connectez-vous à votre espace personnel. Si vous possédez déjà un compte [GitHub](#) ou [Bitbucket](#), il est possible de l’utiliser pour une identification sur C9 sans besoin de créer de nouveau compte ; ceci a l’avantage ajouté de vous permettre d’importer vos projets GitHub et Bitbucket en un seul click.

- (1) Créez un nouveau projet ou ouvrez le projet par défaut nommé **demo-project** avec un click. Après un peu de temps de chargement, vous êtes dans un IDE comme Eclipse ou Netbeans : navigation des fichiers à gauche, éditeur de fichiers au milieu, et terminal Linux en bas.

Objectif du TP

Le but de ce TP est de réaliser une application de chat multi-utilisateurs en utilisant des technologies web recentes (Node.js et Websockets).

Création du fichier HTML pour votre chat

- (2) Tout d’abord créez un nouveau fichier **index.html** qui contiendra le document HTML envoyé à chaque client se connectant au chat. Vous êtes libres dans la conception du document, néanmoins un certain nombre de règles doivent être respectées:
- Le document contiendra au minima: une zone pour taper des messages et une zone pour afficher les messages envoyés et reçus. Les éléments

composant ces zones seront imbriqués dans une balise `<div>` avec l’attribut `id="chat"`.

- La zone d’affichage des messages utilisera la balise `` et aura pour identifiant `messages`.
 - La zone d’entrée de texte utilisera la balise `<input>` et aura pour identifiant `texteenvoi`. Elle sera imbriquée dans un formulaire.
 - Un bouton d’envoi sera ajouté au formulaire. Vous utiliserez la balise `<button>`
- (3) Une fois votre document prêt et sauvegardé, cliquez droit sur l’onglet du fichier et choisissez “Run this file”. Cela exécutera un serveur qui vous permettra de visualiser votre document dans un navigateur.

Voici un exemple de ce que ça peut donner:

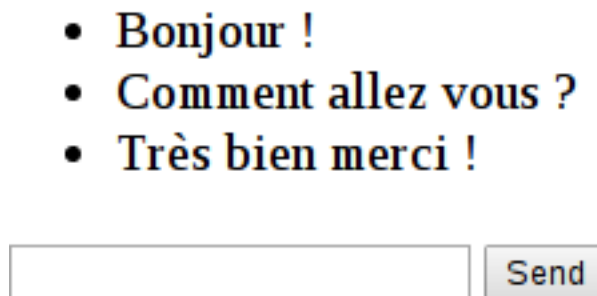


Figure 1: HTML basique du Chat

Création de règles CSS

- (4) Rajoutez des règles CSS en utilisant les balises `<style>` dans les entêtes, de manière à rendre votre document plus élégant.

Par exemple la règle,

```
#messages li:nth-child(odd) { background: #eee; }
```

permet de changer la couleur de fond des balises `` (list item) impaires imbriquées dans un élément ayant pour identifiant `messages`.

En particulier, il est demandé de:

- Changer la taille de police des messages affichés, vous pouvez utiliser l'attribut `font-size`. Attention, comme vu en cours, les éléments de type `<form>` n'héritent pas automatiquement des règles css parentes, mais utilisent par défaut les règles de style du système.
 - Changer la taille de la zone d'entrée de texte pour avoir toute une ligne pour taper les messages. Vous pouvez utiliser les attributs `width` ou `height` pour agir sur les tailles.
 - Vous pouvez utiliser les attributs `margin-left`, `margin-right`, etc. pour agir sur les marges.
- (5) Pour debugger les règles de style vous pouvez utiliser les outils de développement de votre navigateur, qui vous permettent de voir comment les règles CSS sont appliquées. Ils sont accessibles en appuyant sur F12 sous Chrome ou avec Firefox et l'extension firebug.

Voici un exemple:

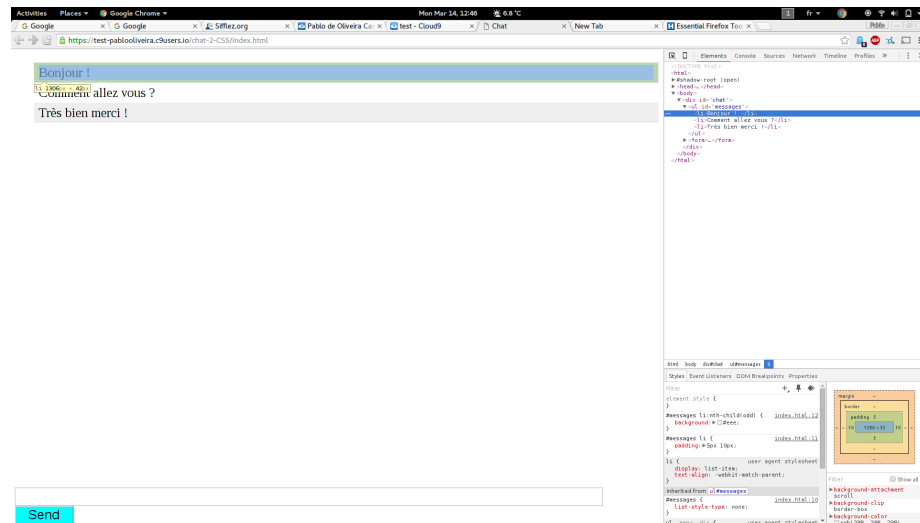


Figure 2: CSS du Chat