



Cours Angular

3.1

| TP Login

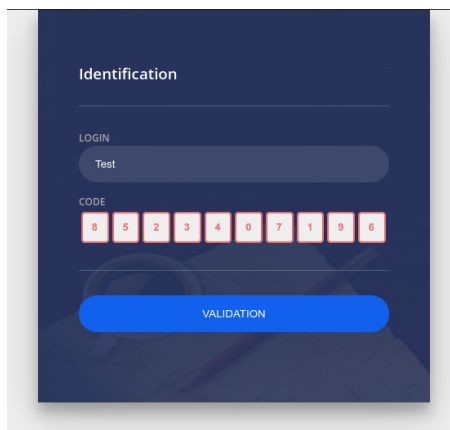
| Changement de principe d'authentification - Améliorations

1] Récupération de l'existant

Vous êtes dans la situation d'un stagiaire qui intègre une entreprise. Le responsable informatique dispose d'une maquette d'application Web HTML/JavaScript et souhaite transposer une partie du code dans une application Angular.

- **Site JS_Login**

T1.1 Récupérez l'archive `JS_Login.zip`, décompressez-la dans un serveur HTTP (XAMP), testez-le site et regardez le code surtout au niveau des méthodes JavaScript :



Explications :

- le login ne doit pas être vide sinon cela gère une erreur
- le mot de passe est remplacé par une série de chiffres aléatoire et ou le nombre entré doit être supérieur ou égal à 6
- le script JavaScript crée un événement sur le bouton au chargement de la page et retourne les deux valeurs entrées (login et code) normalement par AJAX à un serveur (non traité)

2] Modification de l'application Angular

T2.1 A l'aide du code fourni et du précédent TD, changez le mot de passe de l'application Angular suivant la même démarche que l'application classique JavaScript.

Remarques :

- les changements se feront principalement dans le composant `ident`
- les initialisations au chargement sont à voir dans la méthode `ngOnInit()`
- on ne crée plus d'évènement sur le bouton, l'envoi se faisant par le formulaire
- **on ne traitera pas de la validité de l'authentification sur le login et le code du mot de passe dans cette première phase**

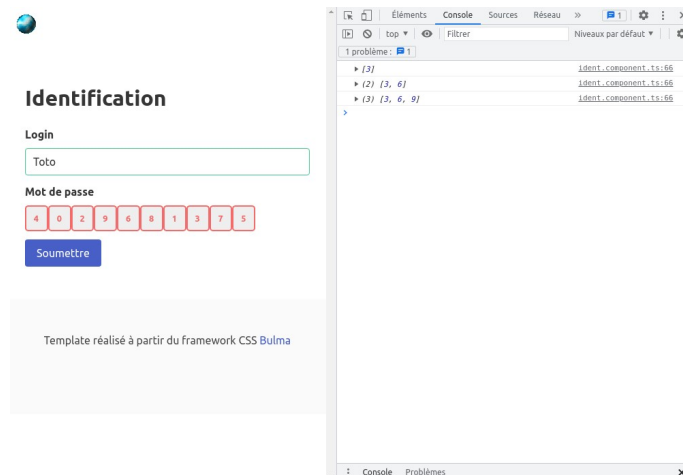
Le responsable informatique, conscient de vos difficultés rencontrés en temps que stagiaire vous donne en plus le travail que lui même avait effectué avant de vous recruter :

Éléments de résolution :

- fichier concerné : `ident.component.ts`
- au niveau des déclarations/initialisations de variables dans la partie `export`, initialisations des variables `tabChiffres` et `tabChiffresSelect` sous la forme d'un tableau de nombres sous cette forme :
`nomDuTableau : number[] = [] ;`
- récupération de la méthode `remplissageTableau()` pour la mettre à la suite de la méthode `authentification()` avec ajustement de l'initialisation du tableau `leTab`.
- appel de la méthode `remplissageTableau()` dans `ngOnInit()` pour remplir le tableau `tabChiffres` au chargement de la page ; vous devrez utiliser le mot clé `this` (et comprendre pourquoi...).
- intégrez dans le fichier CSS du composant `ident` les styles du fichier `chiffres.css` pour un meilleur visuel.
- utilisez la directive Angular `*ngFor` dans la partie d'une `<td>` (on oublie `insertCell`) :

```
<input *ngFor="let chiffre of tabChiffres" id="{{chiffre}}" type='button' class='chiffre'
(click)='remplissageSelect(chiffre)' value='{{chiffre}}' />
```

T2.2 Vérifiez le bon remplissage du code entré dans le tableau `tabChiffreSelect` en intégrant ensuite la méthode `remplissageSelect(chiffre: number)` ; à vous de le faire et vérifiez-en le bon fonctionnement :



Le responsable informatique, en plus de l'aide apporté et avant de s'envoler pour un séminaire informatique sur l'île de la Barbade, vous confie les tâches supplémentaires suivantes :

T2.3 Intégrez le contrôle du code du mot de passe pour que la longueur entrée soit au minimum de 6 chiffres.

T2.4 Intégrez dans le header une structure conditionnelle faisant afficher le mot « Administrateur » à la suite du login dans le cas où celui-ci serait **Toto**.

T2.5 Cherchez (et comprenez) sur Internet le principe de la **méthode d'authentification par jeton**.