

# TD n°5 - Scripts côté client

**Institut Universitaire de Technologie**  
Département Informatique - Campus III Ifs  
Janvier-Février 2015

## -- Pré-requis

Réalisation TD n°4  
Phalcon JQuery

## -- Installation

1. Créer un nouveau projet Phalcon avec les devtools
2. Injectez JQuery au démarrage de l'application (dans le fichier services.php)
3. Intégrez JQuery dans le fichier views/index.volt [par la version 2.1.3](#), ou utilisez l'insertion par CDN automatiques
4. Intégrez Bootstrap pour faciliter certains effets de présentation

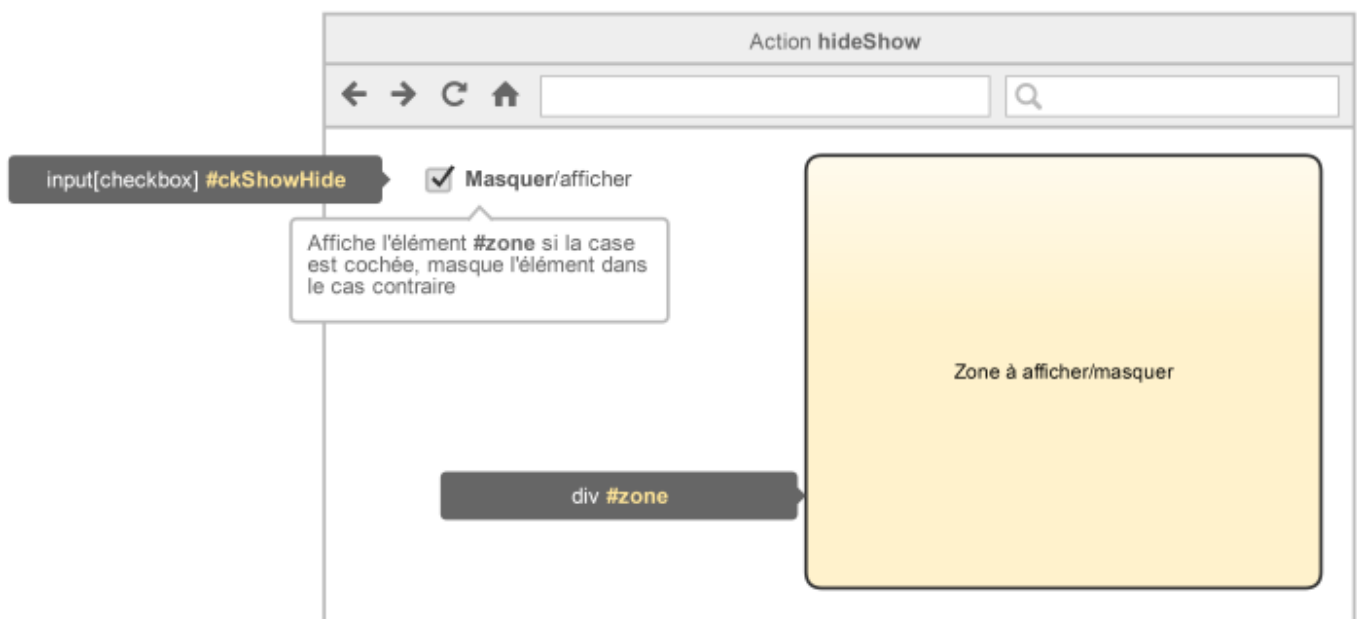
## -- Tests

Effectuer la connexion à la base de données Invo, et reprenez les models du projet précédent.

Créer un contrôleur **testController**

## -- Afficher/masquer des éléments

- Créer la vue **views/test/hideShow.volt** sur le modèle suivant :



- Créer une action **hideShowAction** incorporant le comportement suivant dans la vue :

- Sur Changement de valeur de la case à cocher, la div #zone s'affiche si la case est cochée, et se masque dans le cas contraire
- Ajouter le changement de texte de la checkbox (Elle doit faire apparaître "Masquer" si elle est cochée et "Afficher" dans le cas contraire)

## -- Changement de classe CSS

- Créer la vue **views/test/changeCss.volt** sur le modèle suivant :



- Créer une action **changeCssAction** incorporant le comportement suivant dans la vue :
  - Sur Changement de valeur de la case à cocher, la div #zone s'affiche si la case est cochée, et se masque dans le cas contraire
- Ajouter le changement de texte de la checkbox (Elle doit faire apparaître "Masquer" si elle est cochée et "Afficher" dans le cas contraire)

## -- Requetes Ajax Get sur event

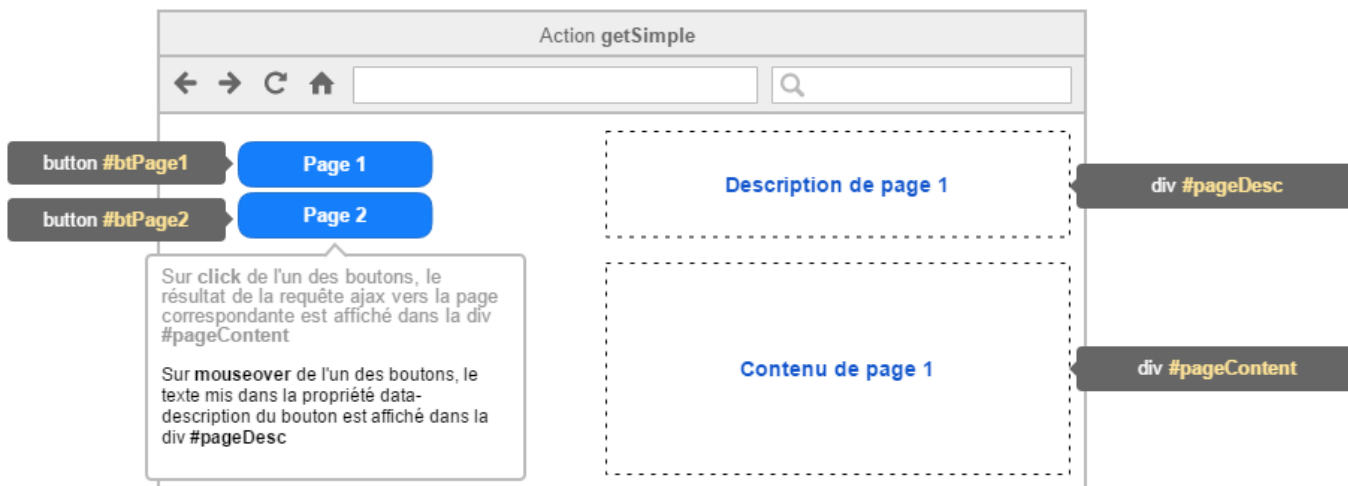
- Créer la vue **views/test/changeCss.volt** sur le modèle suivant :



- Créer une action **getSimpleAction** incorporant le comportement suivant dans la vue :
  - Sur click du bouton **#btPage1**, la page **/test/page1/** est affichée dans la div **#pageContent**
  - Sur click du bouton **#btPage2**, la page **/test/page2/** est affichée dans la div **#pageContent**
- Créer les actions **page1Action** et **page2Action**, se limitant à afficher "Contenu de la page #", désactiver l'affichage de la vue correspondante

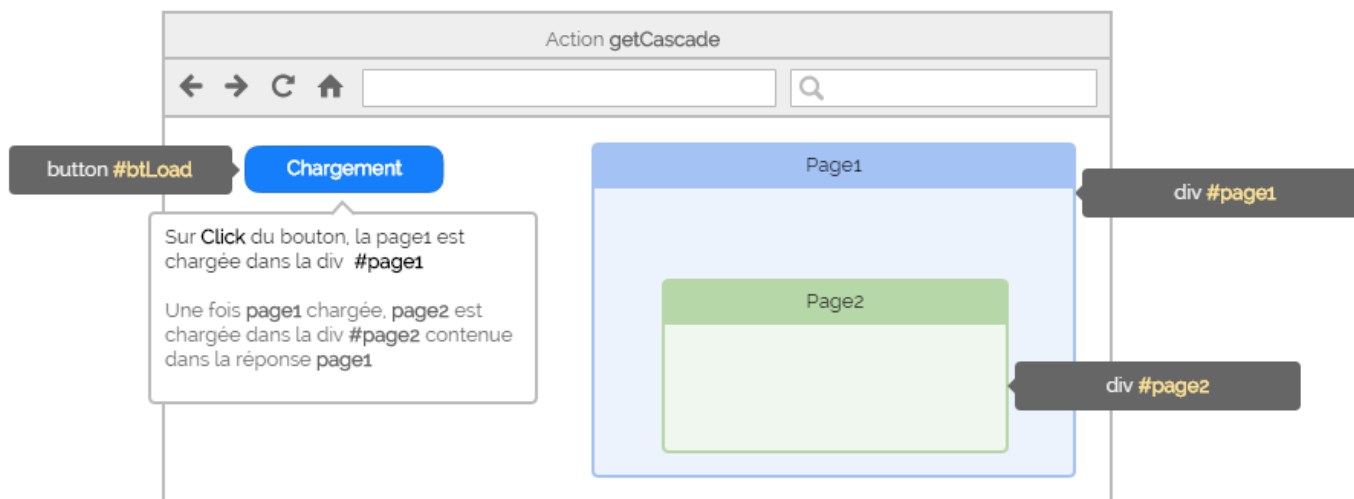
## -- MouseOver

- Dans `getSimpleAction`, ajouter le comportement suivant :
  - sur **mouseover** des bouton, son attribut **data-description** est affiché dans la zone **#pageDesc**



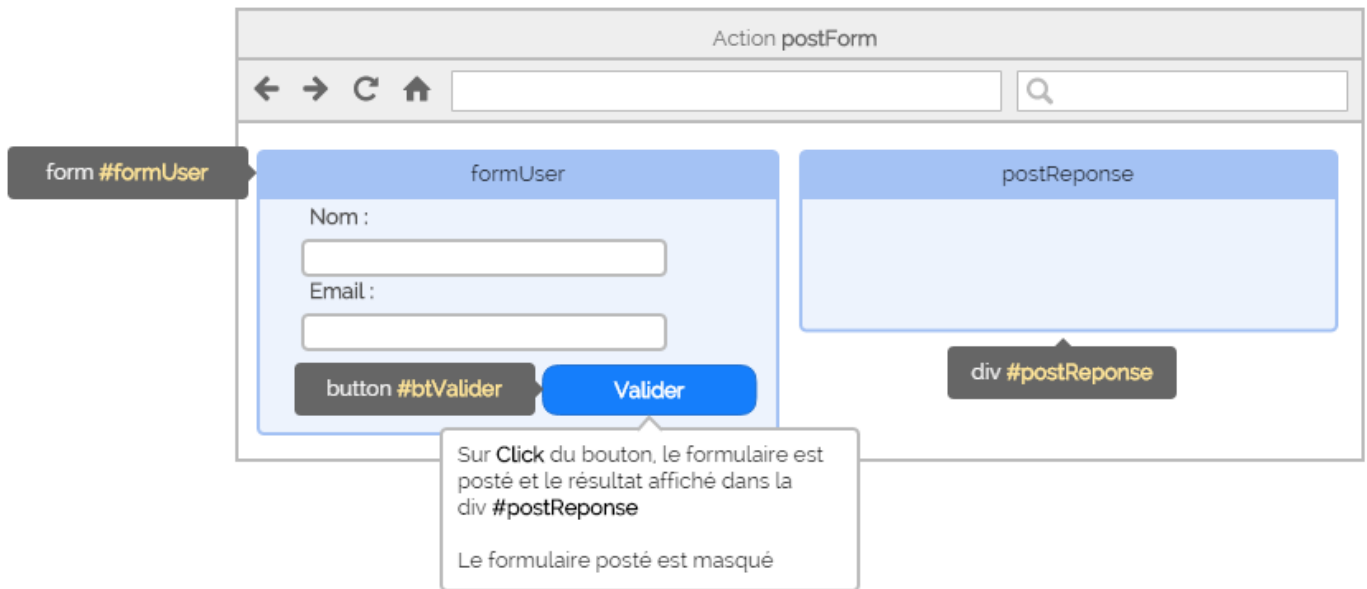
### -- Requetes en cascade

Créer les actions **page1Action**, **page2Action** et les vues (`/views/test/page1` et `/views/test/page2`) correspondant au fonctionnement décrit ci dessous :



### -- POST de formulaire

- Créer l'action **postFormAction**, affichant le formulaire affiché dans la vue **views/test/postForm**, ajouter le comportement défini ci-dessous
- Créer l'action **postReponseAction** affichant la réponse du post du formulaire



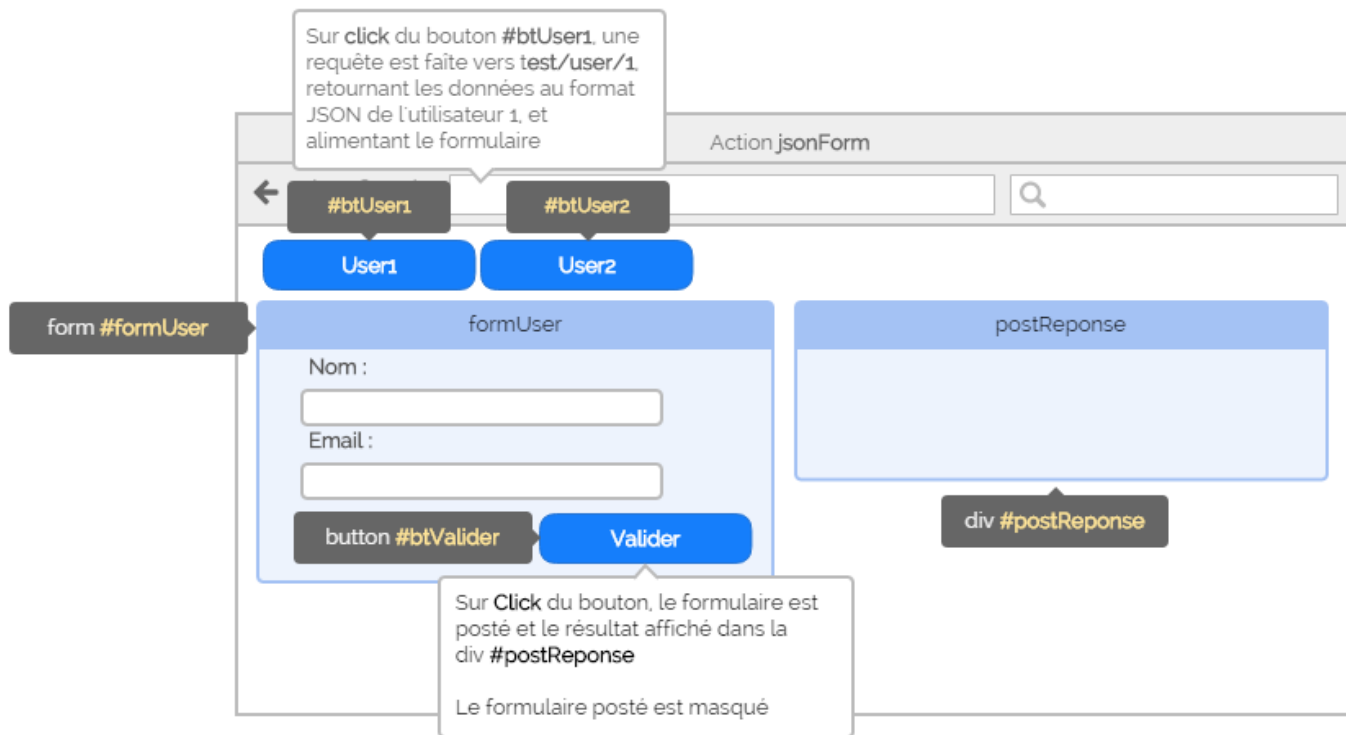
## -- Requête de récupération de données JSON

On ajoute au formulaire précédent la possibilité de récupérer via JSON les données initiales du formulaire à afficher :

- Le **btUser1** permet de récupérer et d'afficher les données de **l'utilisateur1**
- De même le **btUser2** pour **l'utilisateur2**

Créer l'action **userAction** retournant la partie du tableau php contenant les données JSON correspondant à l'utilisateur dont l'index est passé en paramètre :

```
$userArray=array(  
    '{"nom": "SMITH", "email": "BSMITH@mail.com"}',  
    '{"nom": "DOE", "email": "jdoe@mail.com"}'  
);
```



## -- Application

### -- Intégration

1. Générer le model **Products** correspondant à la table products de la base de données **Invo**

From:  
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:  
<http://slamwiki2.kobject.net/slam4/php/phalcon/td5?rev=1425428231>

Last update: **2019/08/31 14:41**

