

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE ROUEN

Département ASI

Architecture des Systèmes d'Information

PAO

Guide de l'utilisateur

Projet

Agent Web

Auteur

Alexandre LE LAIN

Version

v3.1

18 mai 2017

Table des matières

Introduction	2
1 Installation du plugin	3
2 Personnalisation du plugin et de ses fonctionnalités	4
2.1 Personnalisation des fonctionnalités intégrées au plugin	4
2.2 Ajout de nouvelles fonctionnalités en utilisant l'api du plugin	4
2.3 L'API disponible	5

Introduction

Ce projet, tutoré par Alexandre Pauchet, enseignant chercheur à l'INSA Rouen Normandie est réalisé dans le cadre de la formation ASI de l'INSA Rouen en tant que PAO.

Il s'agit d'un plugin JavaScript dont le but est d'apporter aux développeurs d'applications Web des outils pour assister les utilisateurs de leurs produits. Ce document constitue le guide d'installation de ce plugin, et contient toutes les instructions nécessaires pour son utilisation.

Chapitre 1

Installation du plugin

1. Placez le plugin à la racine de votre projet.
2. Dans votre page web :
 - (a) Soyez sûr d'avoir lié les libraires suivantes : JQuery et JQuery UI. Sinon voici leur liens :

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js"></script>
<link rel="stylesheet"
      href="https://ajax.googleapis.com/ajax/libs/jqueryui/1.12.1/themes/smoothness/jquery-ui.css">
```

- (b) Linkez de plus la librairie TTS suivante à la suite dans votre page web :

```
<script src="https://code.responsivevoice.org/responsivevoice.js"></script>
```

- (c) Voici l'appel au plugin que vous pouvez mettre soit à la suite, soit dans le footer de votre page web :

```
<script src="agentWeb/js/script-agentWeb.js" ></script>
```

3. Votre projet doit être situé côté serveur
4. Si vous décidez de renommer le plugin, faites la modification dans le fichier [nomDuPlugin]/js/script-agentWeb.js à la ligne 7

Chapitre 2

Personnalisation du plugin et de ses fonctionnalités

2.1 Personnalisation des fonctionnalités intégrées au plugin

Vous pouvez personnaliser le plugin en utilisant l'interface conçue à cet effet "configuration.php" située à la racine du plugin (ie *agentWeb/*), ou tout simplement éditer directement les fichiers xml dans *agentWeb/src/*.

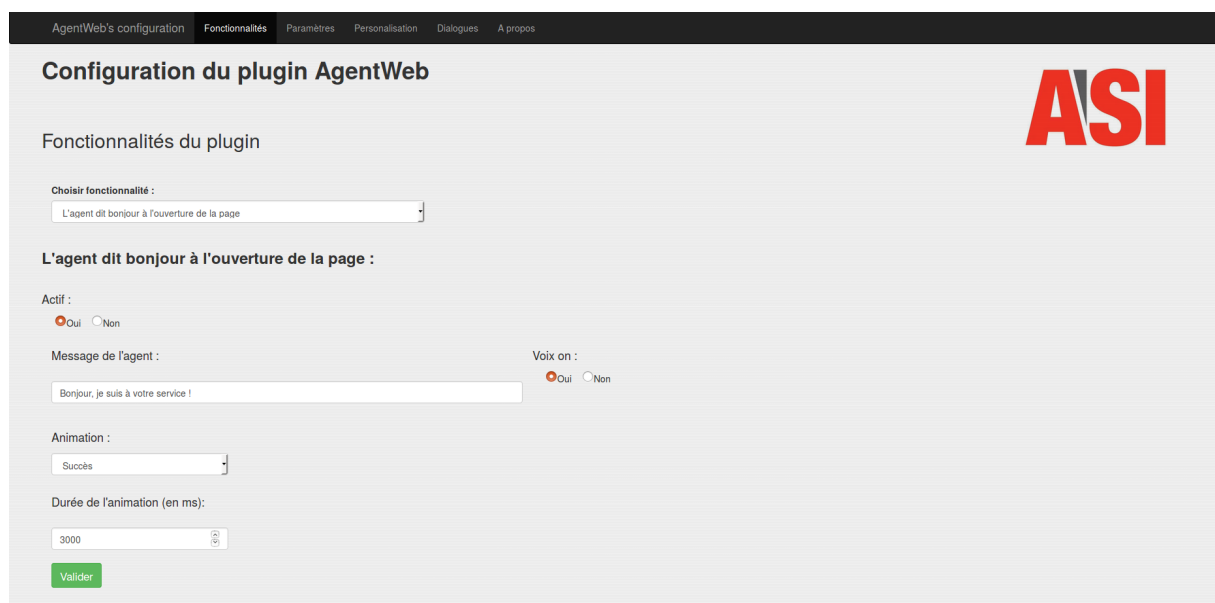


FIGURE 2.1 – L'interface de personnalisation du plugin

2.2 Ajout de nouvelles fonctionnalités en utilisant l'api du plugin

Le plugin propose un agent web qui est capable d'interagir avec l'utilisateur à travers un ensemble d'actions. Une action se caractérise par une animation (un gif), une durée d'animation, un message que l'agent peut prononcer oralement ou non, et d'un event qui correspond à l'événement qui déclenche cette action (ce dernier est cependant optionnel).

Le plugin possède déjà par défaut quelques actions telles que la réaction lorsqu'on lui clique dessus, lorsqu'on écrit dans un champ texte ou lorsqu'on le déplace.

Cependant, si vous le souhaitez, vous pouvez ajouter vos propres actions en suivant ces 2 étapes :

1. Vous devez tout d'abord ajouter les options de votre nouvelle fonctionnalité dans le fichier *agentWeb/src/actions.xml* tel que respectant le pattern suivant :

```

<action>
  <name>[NOM DE VOTRE FONCTION]</name> // le nom de votre action
  <on>[TRUE/FALSE]</on> // True pour activer votre fonction, false sinon.
  <msg>[MSG AGENT]</msg> // Le message que l'agent web dira
  <say>[TRUE/FALSE]</say> // True si vous voulez faire parler l'agent, false sinon.
  <event>[EVENT DE VOTRE FONCTION]</event> // Optionnel : remplissez ce champ si vous
  voulez que votre action se produise à un moment très précis sous certaines conditions.
  <animation>
    <anim>[ANIMATION - CF images]</anim> // l'animation jouée dans votre action
    <duration>[DUREE DE L'ANIMATION]</duration> // sa durée
  </animation>
</action>

```

2. Ensuite, vous pouvez tout simplement implémenter votre fonction dans le fichier *agentWeb/js/utls/actions.js* respectant le pattern suivant :

```

function [NOM DE VOTRE FONCTION](agentWeb,xml,action){
  ...
}

```

Très important : le nom de votre fonction dans le .js doit être le même que dans le .xml pour que votre fonction soit appelée automatiquement. les variables en entrée sont les suivantes :

- **agentWeb** : c'est l'objet de la classe Agent : cf *agentWeb/js/classes/Agent.js* pour connaître son contenu ;
- **xml** : contient toutes les données de tous les fichiers xml de *agentWeb/src/* ;
- **action** : contient tous les paramètres de votre fonction définis à l'étape 1 ;

2.3 L'API disponible

Voici les fonctions disponibles qui vous aideront à créer une nouvelle fonctionnalité pour l'agent web :

```
tellMsg(agentWeb,msg,bool)
```

- **agentWeb** : l'objet agentWeb de la classe Agent
- **msg** : le message que vous voulez que l'agent web affiche
- **bool** : true si vous voulez que l'agent web récite le message à l'oral, false sinon

Cette fonction vous permet de faire dire à l'agent web un message.

```
playAnim(agentWeb,action)
```

- **agentWeb** : l'objet agentWeb de la classe Agent
- **action** : l'objet SimpleXMLElement contenant l'ensemble des balises (<on>,<msg>,<animation>,etc...) de votre action que vous avez renseigné dans le xml. Vous pouvez utiliser les fonctions décrites plus bas pour utiliser cet objet facilement.

Cette fonction vous permet de jouer l'animation de votre fonction définie dans le xml.

```
doAction(agentWeb,action)
```

- **agentWeb** : même objet que celui cité dans la fonction précédente
- **action** : même objet que celui cité dans la fonction précédente

Cette fonction appelle les 2 fonctions listées précédemment : elle permet d'appeler à la fois la fonction qui fait parler l'agent et celle qui joue l'animation.

```
getMsg(action)  
sayMsgOrNot(action)  
getEvent(action)  
getAnim(action)  
getDuration(action)
```

- **action** : l'objet SimpleXMLElement contenant l'ensemble des balises (<on>,<msg>,<animation>,etc...) de votre action que vous avez renseigné dans le xml.

Ces fonctions vous permettent d'obtenir directement les paramètres de votre action définis dans le xml sans avoir à connaître la façon dont on manipule un objet SimpleXMLElement.