



TP1 RIA

PAUL-ANTOINE BISGAMBIGLIA
bisgambiglia@univ-corse.fr

Objectif : apprendre et choisir un framework javascript

Dans ce TP, nous allons tester [Angular](#), [React](#) et [Vue](#) pour développer 4 fois la même petite application. L'objectif n'est pas de développer l'application mais bien de pouvoir avoir du recul pour comparer les « modèles » de programmation des trois frameworks.

Nous allons développer un micro service php qui va prendre en entrée un nom de produit et renvoyer au format json son prix et sa description (et aussi l'évolution du prix).

Il faudra développer une interface utilisateur qui demande le nom du produit, puis la quantité et qui après réponse du serveur va afficher le prix total.

Il doit être possible d'avoir plusieurs produits et connaître leur descriptif.

I. Installation

Installer [nodejs](#)

Node.js® est un environnement d'exécution JavaScript construit sur le [moteur JavaScript V8 de Chrome](#).

[Téléchargement](#).

[Angular](#)

Prérequis : HTML, JS, [TypeScript](#)

Installation avec node > npm install -g @angular/cli

Créer une application > ng new my-app

[React](#)

Prérequis : HTML, JS ES6

Installation

Utilisation :

L'équipe React recommande en premier lieu ces solutions :



- Si vous **apprenez React** ou créez une nouvelle [application web monopage](#), alors utilisez [Create React App](#).
- Si vous construisez un **site web rendu côté serveur avec Node.js**, essayez [Next.js](#).
- Si vous construisez un **site web statique orienté contenu**, essayez [Gatsby](#).
- Si vous construisez une **bibliothèque de composants** ou une **intégration avec du code déjà existant**, essayez [des boîtes à outils plus flexibles](#).

Créer une application :

```
npx create-react-app mon-app  
cd mon-app  
npm start
```

[VueJS](#)

Installation :

```
npm install vue
```

II. Base de données

Vous devez créer une base de données avec les tables :

Produits (id_produit, prix, nom, description, date_input, date_udpate)

Evolutions (id_evolution, id_produit, date_udpate, prix_produit)

```
CREATE TABLE `Produits` (  
  `id_produit` int(11) NOT NULL COMMENT 'clé',  
  `nom` varchar(80) NOT NULL,  
  `date_in` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'date d''ajout',  
  `date_up` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'date de maj',  
  `description` text NOT NULL,  
  `prix` float NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `Evolutions` (  
  `id_evo` int(11) NOT NULL COMMENT 'clé',  
  `id_produit` int(11) NOT NULL COMMENT 'clé étrangère',  
  `date_up` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'date de maj',  
  `prix` float NOT NULL COMMENT 'prix maj'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Vous devez remplir les 2 tables.



III. Partie serveur en PHP :

On va repartir du code de cours du premier semestre.

Fichier microservice.php

```
<?php
header("Content-Type:application/json");
require "data.php";
if(!empty($_GET['name'])) {
    $name=$_GET['name'];
    $price = get_price($name);
    if(empty($price)) {
        response(200,"Product Not Found",NULL);
    }
    else {
        response(200,"Product Found",$price);
    }
}
else {
    response(400,"Invalid Request",NULL);
}
function response($status,$status_message,$data) {
    header("HTTP/1.1 ".$status);
    $response['status']=$status;
    $response['status_message']=$status_message;
    $response['data']=$data;
    $json_response = json_encode($response);
    echo $json_response;
}
```

Fichier data.php à modifier pour lire les données en base et ajouter la description

```
<?php

function get_price($name) {

    $products = [ "book"=>20,

                  "pen"=>10,
```

```

        "pencil"=>5 ];

    foreach($products as $product=>$price) {

        if($product==$name) {

            return $price; break;

        }

    }

}

```

Tester :

<http://localhost/webservice/api.php?name=pen>

Modification du **.htaccess**

RewriteEngine On # Turn on the rewriting engine

RewriteRule ^api/([0-9a-zA-Z_-]*)\$ api.php?name=\$1 [NC,L]

Tester => <http://localhost/projectname/api/pen>

QUESTIONS :

Adaptez le code aux besoins de l'énoncé, puis proposez des fonctions pour :

- Afficher le prix d'un produit (GET)
- Supprimer un produit (DELETE)
- Ajouter un produit (ADD => POST)
- Afficher tous les produits (GET ALL)
- Afficher l'évolution du prix d'un produit (PUT)

Partie client

Fichier **exemple_un_user.php** à adapter

```
<!DOCTYPE html>
```

```
<html lang="en"><head><title>Rest API Client Side Demo</title>
```

```

<meta charset="utf-8"> <meta name="viewport" content="width=device-width,
initial-scale=1"><link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css
"> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></sc
ript> <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"><
/script> </head>

<body><div class="container"> <h2>Rest API Client Side Demo</h2>

<form class="form-inline" action="" method="POST">

<div class="form-group"><label for="name">Name</label>

<input type="text" name="name" class="form-control" placeholder="Enter
Product Name" required/></div>

<button type="submit" name="submit" class="btn btn-default">Submit</button>
</form> <p>&nbsp;</p> <h3>

<?php
if(isset($_POST['submit'])) {
    $name = $_POST['name'];
    $url = "http://localhost/webservice/api/".$name;
    $client = curl_init($url);
    curl_setopt($client,CURLOPT_RETURNTRANSFER,true);
    $response = curl_exec($client);
    $result = json_decode($response);
    echo $result->data; }

?>

</h3> </div> </body> </html>

```

Adaptez le code

TP 2021 js et web service

Menu Home Lecture Ecriture Visualisation

Lecture

Name:

Submit

600

— Paul-Antoine Bisgambiglia

Réalisations

1. Adapter le code aux besoins de l'exercice
2. Proposer une interface client en utilisant seulement de l'ajax (version 1)
3. Proposer 3 interfaces clients à partir des trois frameworks
4. Tirer un bilan de leur utilisation selon les critères suivants :

	React	Angular	Vue
Avis personnel :			
Temps d'apprentissage :			
Complexité du framework :			
Documentation :			
Fonctions proposées :			
Rendu final :			
Autres :			

5. Recoder le webservice php avec node.js