

Rapport de stage

Atelier Numérique



ATELIER NUMÉRIQUE

Romorantin

Période de stage : Du 11 mars au 13 avril 2024

Atelier Numérique

58 Rue des Capucins 1er étage, 41200 Romorantin-Lanthenay

Lycée Claude de France

9 Avenue de Paris, 41200 Romorantin-Lanthenay

Théo Collet

Stage de 2eme année

Sommaire du rapport de stage

(Chaque ligne sera un hyperlien permettant d'accéder à la partie concernée)

SOMMAIRE	PAGES
<u>Rapport de stage</u>	3
<u>Remerciement et Introduction</u>	3
<u>Développement</u>	4
<u>Conclusion</u>	7
<u>Annexes</u>	8
<u>Tout document utile</u>	8
<u>Compétence du référentiel</u>	19
<u>Glossaire</u>	20
<u>Bibliographie/Webographie</u>	21

Remerciement et Introduction

Les objectifs du stage sont de mettre un API fonctionnelle qui sera mis en place plus tard pour faire en sorte que les utilisateurs du site puissent récupérer les données concernées à eux sur leur propre site qui propose la formation. On a pu mettre en place avec mes autres camarades un get qui permet de récupérer les informations dans la base de données et un put pour rentrer les informations dans la base de donnée. On a utilisé l'ajax pour que cela rafraichit automatiquement.

J'ai choisi ce stage car il proposait du travail enrichissant pour moi et que l'année dernière j'ai fait du réseau et que cette année je voulais développer mes compétences de programmeur.

L'atelier numérique est un organisme à but non lucratif qui propose un espace dédié pour du numérique et tous les équipements qui vont avec. Ce qui permet d'apprendre l'utilisation du numérique où utiliser les équipements que la personne n'a pas.

L'atelier numérique propose un site du nom "formadmin" permettant à plusieurs entreprises de proposer des formations grâce à un gestions de formations.

Je souhaite remercier mon maître de stage, monsieur Franck PICHON qui a pu m'aider et me faire comprendre mes incompréhension sur la manière de faire et d'avoir pris du temps pour me prendre en stage. Je remercie aussi mes camarades de classe et collègues de stagiaire avec qui j'ai pu travailler car on a pu s'entraider.

Développement

1er semaine :

Lundi 11 mars :

On a préparé l'environnement de développement et donc notre environnement d'apprentissage le matin, on a utilisé git avec l'utilisation du gestion bitbucket pour qu'on puisse récupérer le code déjà existant, sauvegarder nos modifications à chaque étape réussie et récupérer les modifications prochainement quand l'étape finie. Après je me suis imprégné du code déjà existant pour pouvoir m'adapter et utiliser un le site web pour voir comment cela fonctionne, ce qui a fait que cela c'est fait l'aprem.

mardi 12 mars :

On a fait un brainstorming du développement de l'application web sur ce qu'on doit faire, comment cela va fonctionner, pour cela chacun travailler de leur côté et que si on avait besoin de poser des question, on le faisait. On a préparé un document vierge pour partager que si on avait besoin d'avoir un info, on se le partager dans ce document. Après cela je travailler dans mon coin, ce que j'ai fais, j'ai regarder en détails les fonction déjà existant pour permettre de récupérer sous format JSON les informations d'une méthodes (là c'était getAllFormation() une méthode qui permet de récupérer tous les formations de l'utilisateur du site). Et permettre de rafraîchir automatiquement grâce à de l'Ajax.

mercredi 13 mars :

J'ai pu commencer à travailler ce que j'ai commencé à regarder hier à afficher la méthode toute en utilisant l'Ajax pour éviter de rafraîchir la page pour récupérer les résultats qu'on a pu réussir et on a fait la même chose pour d'autre méthode qui utilise le getAll. Cela permet d'afficher en clair les données sous format JSON. Suite à cela on a fait la même chose pour sessionCollection (pour getAllSession), session pour récupérer la session concernée et formation qui permet de faire la même chose mais pour cette fois la formations

Jeudi 14 mars :

Le maître de stage nous demande d'utiliser une dépendance qui est le json viewer d'andypf qui permet de lire le json correctement et de le personnaliser, suite à cela j'ai essayé de mettre une limite d'affichage pour que la méthode n'affiche pas toutes les données. Pour cela on a créé en brut la valeur l' offset pour savoir quelle données on commence et la limite d'affichage mais on a toujours pas réussi.

vendredi 15 mars :

On a continué ce qu'on a fait et on a réussi à le faire mais vu qu'on a fait cela toute la matinée et qu'on fini tous les vendredis à 12h, on a pas pu faire autre chose.

2eme semaine :

Lundi 18 mars :

Le matin on fait en sorte que les limites et l'offset fonctionne dans l'ajax mais en brute pour que cela fonctionne, mais après dans l'après midi, j'ai du créer des inputs pour permettre sans le mettre en brute les limites et offset qu'on veut pour qu'on puisse rafraîchir les données automatiquement grâce à l'ajax. Ce qui permet de rentrer le nombre qu'on veut.

Mardi 19 mars :

J'ai dû faire une boucle pour que les inputs soit créé automatiquement pour chaque formation qui est concerné les information demandées, donc chaque méthode a des inputs qui est demander et cela se créé automatiquement. Ce qui permet de créer les inputs en fonction des besoins des méthodes, du coup on a créé un tableau qui contient le nom de la méthode et les attributs qu'on a besoin pour permettre de créer les inputs en récupérant le nom dans le tableau.

mercredi 20 mars :

J'ai été absent pour cause pour un speed job dating à Tours pour trouver une alternance qui se trouvait dans l'école SupInfo à Tours. Donc il ont fais le put mais il était pas fini mais je m'en suis pas occupé du stage.

Jeudi 21 mars :

J'ai dû récupérer mon retard et regarder les codes qui ont été ajoutés lorsque j'étais absent, j'ai demandé à mes collègues ce qu'ils ont fait. Suite à cela j'ai dû j'ai du regarder comment automatiser le get avec maxime et donc j'ai regarder les codes concernée bien me l'assimilé le code qui concerne. le but automatisé le get c'est de récupérer les valeurs d'input sans que qu'on doit demander quelle input il doit récupérer, j'ai donc cherché sur internet comment le faire.

vendredi 22 mars :

On a continué avec Maxime et on a essayé de le finir mais cela n'a pas été le cas. Et vu qu'on a fini a midi on a pas pu faire plus pour essayer de le régler.

3eme semaine :

Lundi 25 mars :

On a dû continuer du coup on a créé une variable dynamique qui permet de récupérer le nom de la de l'input et le nom de la méthode pour savoir quelle input le concerne pour ensuite récupérer les valeurs pour permettre de le récupérer dans le get.php. Et on a tenté de supprimer le switch case pour l'automatiser vu que dedans on a mis les nom de méthode

mardi 26 mars et mercredi 27 mars:

On a continué à tenter de le supprimer du coup on a utilisé la même chose que récupéré le nom des inputs mais cette fois prendre le className qui permet de pouvoir récupérer juste le nom de la méthode mais en faisant quand même en sorte que les inputs fonctionne encore.

Jeudi 28 mars et Vendredi 29 mars :

Tenter de faire afficher les statistiques d'une formation mais on a pas réussi car cette fois c'est lié à une autre méthode qui permet d'afficher autrement. et donc on a été bloqué. Il fallait que les valeurs qu'on a besoin s'affiche dans l'url et de le récupérer dans le get.php

4eme semaine :

Mardi 2 avril :

Demander a notre maître de stage qui nous débloquent pour permettre d'afficher les données du coup on a fait une réunion pour permettre de nous débloquent et expliquer comment faire. Il nous a montré ce qu'on a fait. Et ensuite de cela il nous demande de faire en sorte que la méthode ne le fasse pas buguer. Il nous a montré de récupérer les valeur des input dans le l'url ce qui veut dire le nom de la méthode concernée et les valeur qui concerne les inputs et de le récupérer dans le get.php grâce au \$_GET. Maintenant les autres méthodes bug et on doit le corriger.

mercredi 3 avril et jeudi 4 avril :

On a dû créer plusieurs JSONviewer pour que cela affiche les statistiques et les méthodes. Donc on a rajouté dans le tableau dimensionnel le nom de la méthode concernée et faire en sorte que ça parcourt dans la dimension du tableau concerné suite à cela on a essayé de faire fonctionner et cela était le cas du coup le jour même et le lendemain on a rajouté toutes les méthodes et les attributs de ces méthodes dans le tableau multidimensionnelle. On a testé chaque méthode et on a noté tous ceux qui ne fonctionnaient pas.

Vendredi 5 avril :

On a testé les dernières méthodes pas encore qui n'ont pas encore été testé puis essayer de corriger les méthodes qui ne fonctionne pas, ensuite on a un début de correction des méthodes.

5eme semaine :

Lundi 8 avril

On a continuer de corriger les derniers méthodes non spécifique tel que le getFormationSession mais après on a essayé de faire fonctionner les méthodes counts, les méthodes doit s'afficher autrement donc j'ai essayer de commencer à afficher mais j'ai changer l'après midi car je devais afficher les valeur de l'ajax de la page vers la fin de l'aprem.

Mardi 9 avril jusqu'à jeudi 11 avril

j'ai continué à travailler sur l'affichage de l'Ajax pour que des personnes puissent copier. Et donc j'ai réussi à l'afficher pour chaque nom de la classe et chaque méthode de chaque classe et j'ai fais en sorte qu'il soit bien formater, au début j'avais eu du mal à comprendre ce qu'il attendait le maître de stage jusqu'à ce que je le réussi à le comprendre et que je le mette en place.

Vendredi 12 avril

Correction de l'affichage d'erreur vu qu'on utilisait le xhr, on a utilisé le jqXHR et on a fini ce qu'on devait faire du coup j'ai avancé le rapport.

Conclusion

En conclusion de mon stage j'ai pu me rendre compte de mon manque de connaissance et que je vais devoir en apprendre encore plus pour combler mes lacunes. Mais le stage m'a vraiment plu et progresser. Les compétences que j'ai pu développer et améliorer mes gestions de code, mieux utiliser les langages de programmation, me créer un tableau de bord, faire la méthode scrum tous les jours. En soit avoir de l'organisation. J'ai pu aussi travailler le travail d'équipe avec les autres. L'utilisation de git un peu plus pousser m'a permis de mieux de comprendre aussi. Et connaître l'ajax et mieux utiliser le json.

Au sujet de l'organisation, je pense qu'il pourra encore plus se développer dans les prochaines années et permettre de l'utilisation dans le numérique pour beaucoup de personnes.

Maintenant pour mon avenir, j'appréhende beaucoup pour après le BTS ou je me pose des question, de si je vais réussir, mais j'essayerai de m'améliorer et d'acquérir mon manque de connaissance. Je suis encore plus déterminé à faire des études supérieures.

Tout document utile

Site web :

Je montre le fonctionnement que pour la partie formation collection pour le getAll :

FormationCollection

idSpecialite : avecStagiaires : objectif : idBp : order :

order : offset : limit :

```

{ 6 items
  ▶ "pdo": {...} 0 items
  ▶ "_pdo": {...} 0 items
  ▶ "arrayObjFormation": {...} 10 items
  ▶ "_arrayObjFormation": {...} 10 items
  "ID_USER": string "8"
  "_ID_USER": string "8"
}

```

```

{ 6 items
  ▶ "pdo": {...} 0 items
  ▶ "_pdo": {...} 0 items
  ▶ "arrayObjFormation": [ 10 items
    ▶ 0: { 46 items
      "ID": int 237
      ▶ "arrayObjFormationPhotos": {...} 1 item
      "ID_USER": int 8
      "ID_SPECIALITE": int 331
      "ID_OBJECTIF": int 8
      "REF": string ""
      "TITRE": string "GESTES ET POSTURES TECHNIQUES DE MANUTENTIONS MANUELLES"
      "TITRE_PAGE": string "gestes-et-postures-techniques-de-manutentions-manuelles"
      "FAMILLE":
      "DUREE": string "1 jour"
      "NB_HEURES_FORMATION": int 7
      "NB_MINUTES_FORMATION": string "00"
      "OBJECTIFS": string ""
      "RECYCLAGE": -
      string "
    }
  ]
}

```



```
$.ajax ({
  URL : http://localhost/formadmin/admin/webservice/externe/get.php
  Type : POST
  DataType : json
  Data :
  {
    "objet": "FormationCollection",
    "param": {
      "ID_USER": ""
    },
    "methode": [
      {
        "name": "getall",
        "value": {
          "idSpecialite": "",
          "avecStagiaires": "",
          "objectif": "true",
          "idBp": "",
          "order": "",
          "offset": "",
          "limit": "10"
        }
      }
    ]
  }
})
```

Et on a fait pour plusieurs méthode :

The screenshot shows a REST API client interface with the following endpoints and parameters:

- FormationCollection**
 - idSpecialite : avecStagiaires : objectif : idBp : order : offset : limit : Requête
- SessionCollection**
 - order : offset : limit : Requête
- ID_CLIENT**
 - ID_CLIENT : order : getAllSessionClient
- ID_PARTICIPANTS**
 - ID_PARTICIPANTS : ID_ORGANISME : order : getAllSessionStagiaireByIdOrganisme
- ID_FORMATION**
 - ID_FORMATION : getByIdFormation

ID_PARTICIPANTS : order :

order :

ID_FORMATION :

REF :

FacturesCollection

order : offset : limit :

ID_CLIENT : order :

ClientCollection

order : offset : limit :

EMAIL :

search :

countClient

StagiaireCollection

offset : limit :

ID_CLIENT :

ID_SESSION : ID_CLIENT :

TYPE_FORMATION :

TYPE_FORMATION : ID_CLIENT :

ID_SESSION :

ID_CLIENT : order :

Formation

ID :

Stagiaire

ID : Requête

Q Search

Client

ID : Requête

Q Search

order : getSessionClient

Q Search

Email : table : mailExiste

Q Search

Devis

ID : Requête

Q Search

ID_SESS : ID_CLIENT : getDevisBySession

Q Search

Devis

ID : Requête

Q Search

ID_SESS : ID_CLIENT : getDevisBySession

Q Search

Factures

ID_FACTURE : Requête

Q Search

ID_FACTURE : getFactureUser

Q Search

Formation

REF

TITRE

TITRE_PAGE

DUREE

et en dessous avec l'écriture en vert c'est la partie put :

Formation

REF

TITRE

TITRE_PAGE

DUREE

NB_HEURES_FORMATION

NB_MINUTES_FORMATION

OBJECTIFS

PUBLIC

PREREQUIS

PROGRAMME

METHODE

EVALUATION

CONTACT

BP

PDF_PROGRAMME

COMPETENCES

LIEN_CPF

Enregistrer à la base de données

Et pareil la personne a mis plusieurs méthode dedans, mais je ne vais pas tout montrer car c'est le même principe que vous voyez juste au dessus, et d'autant plus que ce n'est pas la partie que j'ai faites.

code :

api.view.php :

```
<?php
ob_start();
?>
<!-- Content Header (Page header) -->
<section class="content-header">
    <script src="https://cdn.jsdelivr.net/npm/@andypf/json-viewer"></script>
    <h1>= $titre ?&gt;&lt;small&gt;&amp;nbsp;&nbsp;&nbsp;?&gt;&lt;/small&gt;&lt;/h1&gt;
    &lt;ol class="breadcrumb"&gt;
        &lt;li&gt;&lt;a href="index.html"&gt;&lt;i class="fa fa-dashboard"&gt;&lt;/i&gt; Accueil&lt;/a&gt;&lt;/li&gt;
        &lt;li class="active"&gt;Améliorations&lt;/li&gt;
    &lt;/ol&gt;
&lt;/section&gt;

&lt;!-- Main content --&gt;
&lt;?php
$arrayApiClassName = [
    'FormationCollection' =&gt; ['attribut' =&gt;['idSpecialite', 'avecStagiaires', 'objectif', 'idBp', 'order', 'offset',
    'SessionCollection' =&gt; ['attribut' =&gt;['order', 'offset', 'limit'], 'methode' =&gt;['getAllSessionClient' =&gt; ['ID_CLI
    'getByIdFormation'=&gt;['ID_FORMATION'], 'getAllSessionStagiaire' =&gt;['ID_PARTICIPANTS', 'ord
    'FacturesCollection' =&gt; ['attribut' =&gt;['order', 'offset', 'limit'], 'methode'=&gt;['getAllFacturesClient'=&gt;['ID_CLI
    'ClientCollection'=&gt; ['attribut' =&gt;['order', 'offset', 'limit'], 'methode' =&gt;['getByEmail'=&gt;['EMAIL'], 'getSearch
    'StagiaireCollection' =&gt; ['attribut' =&gt;['offset', 'limit'], 'methode' =&gt; ['getAllStagiaireClient' =&gt; ['ID_CLIENT
    'DevisCollection' =&gt; ['attribut' =&gt;['order', 'offset', 'limit'], 'methode' =&gt;['countDevisTousTypes'=&gt;[''], 'count
    'Formation' =&gt; ['attribut' =&gt;['ID'], 'methode' =&gt;['getStatistique' =&gt;['']],
    'Session' =&gt; ['attribut' =&gt;['ID', 'dateSup', 'noDetail'], 'methode' =&gt;['countGetStagiaireClient' =&gt;['ID', 'ID_CLI
    'Stagiaire'=&gt;['attribut' =&gt;['ID']],
    'Client' =&gt;['attribut' =&gt;['ID'], 'methode' =&gt;['getSessionClient' =&gt;['order'], 'mailExiste'=&gt;['Email', 'table']]],
    'Devis' =&gt;['attribut' =&gt;['ID'], 'methode' =&gt;['getDevisBySession' =&gt;['ID_SESS', 'ID_CLIENT']]],
    'Factures' =&gt;['attribut' =&gt;['ID_FACTURE'], 'methode' =&gt;['getFactureUser' =&gt;['ID_FACTURE']]]];

?&gt;

&lt;section class="content"&gt;
&lt;?php foreach ($arrayApiClassName as $className =&gt; $data) { ?&gt;
    &lt;h1 style="color:red;"&gt;&lt;?= $className ?&gt;&lt;/h1&gt;
    &lt;?php foreach ($data['attribut'] as $attributs) { ?&gt;
        &lt;label for="&lt;?= $attributs . $className ?&gt;"&gt;&lt;?= $attributs ?&gt; &lt;/label&gt;
    }
}
&lt;/section&gt;</pre
```

```
<section class="content">
<?php foreach ($arrayApiClassName as $className => $data) { ?>
    <h1 style="color:red;"><?= $className ?></h1>
    <?php foreach ($data['attribut'] as $attributs) { ?>
        <label for="<?= $attributs . $className ?>"><?= $attributs ?> </label>
        <input id="<?= $attributs . $className ?>" placeholder="<?= $attributs ?>" value="">
    <?php } ?>
    <button class="btnJson" className="<?= $className ?>">Requête</button>
    <br>
    <andypf-json-viewer theme="ia-dark" show-toolbar="true" id="jsonViewer<?= $className ?>"></andypf-json-viewer>
    <br>
    <label for="data<?= $className ?>" id="data<?= $className ?>"></label>
    <br>
    <?php if (isset($data['methode'])) { ?>
        <?php foreach (array_keys($data['methode']) as $methode) { ?>
            <?php if ($data['methode'][$methode] != ['']) { ?>
                <?php foreach ($data['methode'][$methode] as $argument) { ?>
                    <label for="<?= $argument . $methode ?>"><?= $argument ?> </label>
                    <input id="<?= $argument . $methode ?>" placeholder="<?= $argument ?>" value="">
                <?php } ?>
            <?php } ?>
            <button id="<?= $methode ?>" class="btnJson" className="<?= $className ?>" methodes="<?= $methode ?>"><?= $methode ?></button>
            <br>
            <andypf-json-viewer theme="ia-dark" show-toolbar="true" id="jsonViewer<?= $className ?><?= $methode ?>"></andypf-json-viewer>
            <br>
            <label for="data<?= $methode . $className ?>" id="data<?= $methode . $className ?>"></label>
            <br>
        <?php } ?>
    <?php } ?>
</section>
```

ceux ci c'est la partie visuelle juste au dessus

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
  $(document).ready(function() {
    $(".btnJson").on('click', function(event) {

      let btn = event.currentTarget;
      let Methode = btn.getAttribute('methodes');
      let className = btn.getAttribute('className');

      console.log(className);
      console.log(Methode);

      var param = {};
      var arrayMethodeValue = {};

      //param.className = className;
      param.ID_USER = "8";
      //id_user "6", id_participant "24", id_organisme "21"
      //param.ID_FACTURE = "10";
      //param.ID_CLIENT = "586";

      <?php
        $jsonArrayApiClassName = json_encode($arrayApiClassName);
      ?>
      var arrayApiClassName = <?php echo $jsonArrayApiClassName; ?>;

      if(Methode === null){
        for (let i = 0; i < arrayApiClassName[className]['attribut'].length; i++) {
          var attribut = arrayApiClassName[className]['attribut'][i];
          var element = document.getElementById(attribut + className);
          console.log(attribut);
          if (element) {
            arrayMethodeValue[attribut] = element.value;
          } else {
            console.log("L'élément avec l'ID ", attribut + className, " n'a pas été trouvé.");
          }
        }
      }
    });
  });
</script>
```

cette partie permet de récupérer le de la classe et de la méthode en fonction de quelle bouton on a cliquer

```
if(Methode === null){
  for (let i = 0; i < arrayApiClassName[className]['attribut'].length; i++) {
    var attribut = arrayApiClassName[className]['attribut'][i];
    var element = document.getElementById(attribut + className);
    console.log(attribut);
    if (element) {
      arrayMethodeValue[attribut] = element.value;
    } else {
      console.log("L'élément avec l'ID ", attribut + className, " n'a pas été trouvé.");
    }
  }
} else {
  for (let i = 0; i < arrayApiClassName[className]['methode'][Methode].length; i++) {
    var argument = arrayApiClassName[className]['methode'][Methode][i];
    var element = document.getElementById(argument + Methode);
    console.log(argument);
    if (element) {
      arrayMethodeValue[argument] = element.value;
    } else {
      console.log("L'élément avec l'ID ", argument, " n'a pas été trouvé.");
    }
  }
}
}
```

on récupère les valeur de la méthode en les comprenant que c'est relier a l'attribut qui se trouve dans le tableau multidimensionnel

```

    if (className.includes("Collection")) {
        if(Methode === null) {
            var methode = [{ 'name': 'getall', 'value': arrayMethodeValue},,];
        }
    } else {
        if(Methode === null) {
            var methode = [{ 'name': 'get', 'value': arrayMethodeValue},,];
        }
    }

    let url = 'http://localhost/formadmin/admin/webservice/externe/get.php';
    if (Methode && arrayApiClassName[className]['methode'][Methode]) {
        url += '?' + Methode;
        arrayApiClassName[className]['methode'][Methode].forEach(function(argument, index) {
            if (argument !== '') {
                if(index != 0) {
                    url += ',';
                } else {
                    url += '=';
                }
                if (arrayMethodeValue[argument] == ''){
                    url += null;
                } else {
                    url += arrayMethodeValue[argument];
                }
            }
        });
    }
}

```

Les appels de la méthode ou on clique en récupérant les valeurs des inputs.

```

$.ajax({
    url: url,
    type: 'POST',
    dataType: "json",
    data: JSON.stringify({
        objet: className,
        param: param,
        methode: methode ? methode : null,
    })
})

```

l'appel de la méthode ajax de jQuery

```

    }).done(function(response) {
        var requestDataString = {
            url: this.url,
            type: this.type,
            dataType: this.dataType,
            data: JSON.parse(this.data),
        };

        function hideDataValues(obj) {
            for (var key in obj) {
                if (obj.hasOwnProperty(key)) {
                    obj[key] = "";
                }
            }
            return obj;
        }

        function deleteDataValues(obj) {
            console.log("La fonction deleteDataValues est appelée avec l'objet suivant :", obj);
            for (var key in obj) {
                if (obj.hasOwnProperty(key)) {
                    if (obj[key] === null) {
                        console.log("Suppression de la clé", key, "car sa valeur est null :", obj[key]);
                        delete obj[key];
                    }
                }
            }
            console.log("Objet après suppression des valeurs null :", obj);
            return obj;
        }
    });

```

Si il réussit, il tente s'afficher la requête ajax juste en dessous du jsonViewer et il cache tous ce qui se trouve dans le param les valeur et si la valeur de la méthode est null, il ne doit pas l'afficher

```

    if (requestDataString.data.param) {
        requestDataString.data.param = hideDataValues(requestDataString.data.param);
    }
    if (requestDataString.data) {
        requestDataString.data = deleteDataValues(requestDataString.data);
    }

```

l'appel des deux fonction que j'ai expliqué juste au dessus

```

        console.log(requestDataString.data);

        var requestDataStringText =
            "$.ajax ({ <br>" +
            "URL : " + requestDataString.url + "<br>" +
            "Type : " + requestDataString.type + "<br>" +
            "DataType : " + requestDataString.dataType + "<br>" +
            "Data :<br>" +
            "<div style='white-space: pre-wrap;'>" +
            JSON.stringify(requestDataString.data, null, 4) +
            "}</div>";

        if (Methode === null) {
            document.getElementById('jsonViewer' + className).data = response;
            document.getElementById('data' + className).innerHTML = requestDataStringText;
        } else {
            document.getElementById('jsonViewer' + className + Methode).data = response;
            document.getElementById('data' + Methode + className).innerHTML = requestDataStringText;
        }
    }).fail(function(jqXHR, textStatus, errorThrown) {
        console.error(textStatus + ' ' + errorThrown);
    });
});
</script>

```

La partie visuelle sur comment s'affiche la requête Ajax en dessous du jsonViewer et on fait afficher les données en format json dans les jsonViewer concernent sinon il affiche une erreur si la requête ajax est ratée.

la partie put que je n'ai pas fait mais par mes collègues :

```
<!-- Ici la méthode put -->
<!-- essayer de faire les params manuellement et de les envoyer dans la base de données -->
<?php
$arrayApiClassName = [
    'Formation' => ['REF', 'TITRE', 'TITRE_PAGE', 'DUREE', 'NB_HEURES_FORMATION', 'NB_MINUTES_FORMATION', 'OBJECTIFS', 'PUBLIC', 'PREREQUIS', 'PROGRAMME'],
    'Session' => ['ID_FORMATION', 'TYPE_FORMATION', 'INTITULE_FORMATION', 'ADR1', 'ADR2', 'CP', 'LIEU', 'DATE_DEBUT', 'DATE_FIN', 'DATES_TXT', 'TARIF', 'TAXES'],
    'Devis' => ['NUM_DEVIS', 'RAISON_SOCIALE', 'CIVILITE', 'NOM', 'PRENOM', 'ADR1', 'ADR2', 'CP', 'VILLE', 'TEL1', 'EMAIL', 'INTITULE_FORMATION', 'DUREE', 'TARIF'],
    'Stagiaire' => ['CIVILITE', 'NOM', 'PRENOM', 'ADRESSE', 'CODEPOST', 'VILLE', 'TEL', 'EMAIL', 'DATE_NAISS', 'TEXTE', 'RAISON_SOCIALE', 'SIRET', 'NOM_JEUX'],
    'Factures' => ['ID_CLIENT', 'TYPE_FACTURE', 'TOTAL_HT', 'NUM_FAC', 'DATE_CREA', 'ENVOYEE', 'NB_JOURS', 'NB_HEURES', 'NB_MINUTES', 'NUM_INTERNE'],
    'Client' => ['_RAISON_SOCIALE', '_STATUT', '_STATUT_ENTREPRISE', '_REF_CLIENT', '_SIRET', '_ADR1', '_ADR2', '_CP', '_VILLE', '_PAYS', '_TEL1', '_TEL2']
];

<section class="content">
    <?php foreach($arrayApiClassName as $className => $attributs): ?>
        <h1 style="color:green;"><?=$className ?></h1>
        <?php foreach($attributs as $attribut): ?><br>
            <div>
                <label for="<?=$attribut . $className ?>"><?=$attribut ?></label>
                <input type="text" id="<?=$attribut . $className ?>" name="<?=$attribut . $className ?>" placeholder="<?=$attribut ?>" value="">
            </div>
        <?php endforeach; ?>
        <button class="btnPut" data-classname="<?=$className ?>">Enregistrer à la base de données</button>
    <?php endforeach; ?>
</section>
```

le visuel de la partie put

```
<script>
document.addEventListener('DOMContentLoaded', function() {
    $(".btnPut").on('click', function(event) {
        let btn = event.currentTarget;
        let className = btn.getAttribute('data-classname');
        console.log(className);

        var param = {};
        param.className = className;

        var arrayApiClassName = <?php echo json_encode($arrayApiClassName); ?>;

        <?php if(true) { ?>
            for (let i = 0; i < arrayApiClassName[className].length; i++) {
                var attribut = arrayApiClassName[className][i];
                var element = document.getElementById(attribut + className); // Utilisation de l'attribut comme partie de l'ID
                if (element) {
                    param[attribut] = element.value;
                } else {
                    console.log("L'élément avec l'ID ", attribut + className, " n'a pas été trouvé.");
                }
            }
        <?php ?>

        $.ajax({
            url: 'http://localhost/formadmin/admin/webbservice/externe/put.php',
            type: 'PUT',
            dataType: "json",
            data: JSON.stringify(param),
        }).done(function(response) {
            console.log(response);
        }).fail(function(jqXHR, textStatus, errorThrown) {
            console.error(textStatus + ' ' + errorThrown);
        });
    });
});
</script>
<?php
$html_content = ob_get_clean();
?>
```

récupération des classname et inputs, puis appel ajax

get.php


```

32
33
34 if (isset($request->objet)) {
35
36     foreach ($request->param as $attribut => $value) {
37         ${$attribut} = $value;
38     }
39
40     $className = $request->objet;
41     $obj = new $className($pCon->getInstance(), $ID_USER);
42     $methods = get_class_methods($className);
43
44
45
46     if(is_array($request->methode)) {
47         foreach($request->methode as $methode) {
48             if (method_exists($obj, $methode->name)) {
49                 $obj->{$methode->name}($methode->value);
50             }else{
51                 echo "la methode n'existe pas";
52             }
53         }
54     } else {
55     }
56 }

```

Ceux qui permet d'appeler les méthode get et getAll avec ces valeurs

```

foreach ($_GET as $methode => $arg) {
    $arrayArg = explode(',', $arg);

    if (strpos($methode, 'count') !== false && method_exists($obj, $methode)) {
        $count = $obj->$methode(...$arrayArg);

        $arrayJsonData[] = [
            'classe' => $className,
            'methode' => $methode,
            'resultat' => $count
        ];

        print json_encode($arrayJsonData);
        exit;
    }elseif ($methode === 'mailExiste' && method_exists($obj, $methode)) {
        $bool = $obj->$methode(...$arrayArg);
        $arrayJsonData[] = [
            'classe' => $className,
            'methode' => $methode,
            'resultat' => (bool)$bool
        ];
        print json_encode($arrayJsonData);
        exit;
    }elseif (strpos($methode, 'count') === false && method_exists($obj, $methode)){
        $obj->{$methode}($arrayArg);
    } else {
        echo "La méthode $methode n'existe pas";
    }
}
}

```

Ceux qui permet d'afficher les données avec les méthode qui ont les valeur rentrer dans l'url donc les méthodes les plus spécifique avec les méthode particulier sur l'affichage

```

$arrayJsonData = $obj->getJsonData($obj);
$resultJson = json_encode($arrayJsonData);

if ($resultJson !== false) {
    print $resultJson;
} else {
    http_response_code(400);
    echo json_encode(array('error' => 'Propriété "objet" manquante'));
}

?>

```

La récupération des données pour après le faire afficher

put.php

```

$postdata = file_get_contents("php://input");
$request = json_decode($postdata, true);
$affDump = isset($_GET['varDump']) ? true : false;
$arrayJsonData= array();

if (!empty($request))
{
    $className = $request['className'];
    $obj = new $className($pCon->getInstance(), 7);
    $saveMethod = 'save' . $className;

    foreach ($request as $key => $value) {
        if ($key !== 'className') {
            $obj->setVal($key, $value);
        }
    }

    if (method_exists($obj, $saveMethod)) {
        $success = $obj->$saveMethod();
        if ($success) {
            echo json_encode(array('success' => true, 'message' => 'Données enregistrées.));
        } else {
            echo json_encode(array('success' => false, 'message' => 'Échec de l\'enregistrement des données.));
        }
    } else {
        echo json_encode(array('success' => false, 'message' => 'La classe '.$className.' ne possède pas de méthode de sauvegarde.));
    }
}
}

```

récupération des valeur pour appeler la méthode concernée, c'est le même principe que get.php mais cette pour insérer les données dans la base de données.

```

public function get($ID){
    if (is_object($ID)){
        $arrayTemp = $ID;
        $ID = (isset($arrayTemp->ID) && $arrayTemp->ID != "") ? $arrayTemp->ID : $ID;
    }
    $this->getClient($ID);
}

```

On a aussi du ternaire sur les méthodes qui en ont besoin pour prendre la valeur de soit l'input soit la valeur de base

Compétences référentiels

Les compétences citée en dessous correspond à ce que j'ai travaillé :

B1.3.3, B1.4.1, B1.4.2, B1.6.1, B2B.1.3, B2B.1.8, B2B.1.9, B2B.3.1, B2B.3.2

B1.3.3	Participer à l'évolution d'un site Web exploitant les données de l'organisation.	P1.3.5	Le site Web a évolué conformément au besoin exprimé.
B1.4 Travailler en mode projet		B1.4 Travailler en mode projet	
B1.4.1	Analyser les objectifs et les modalités d'organisation d'un projet	P1.4.1	Les objectifs et les modalités d'organisation du projet sont explicités.
		P1.4.2	L'analyse des besoins et de l'existant est pertinente.
B1.4.2	Planifier les activités	P1.4.3	Les activités personnelles sont planifiées selon une méthodologie donnée et les ressources humaines, matérielles et logicielles nécessaires sont mobilisées de manière efficace et pertinente.
		P1.4.4	Le découpage en tâches est réaliste.
		P1.4.5	Les livrables sont conformes.
		P1.4.6	Le projet est documenté.
B1.6 Organiser son développement professionnel		B1.6 Organiser son développement professionnel	
B1.6.1	Mettre en place son environnement d'apprentissage personnel	P1.6.1	Les besoins de formation sont identifiés pour assurer le support ou mettre à disposition un service.
		P1.6.2	L'environnement d'apprentissage personnel est délimité et expliqué.
B2B.1.3	Modéliser une solution applicative	P2B.1.1.2	La maquette des éléments applicatifs de la solution respecte les fonctionnalités exprimées ;
B2B.1.8 B2B.1.9	Intégrer en continu les versions d'une solution applicative Réaliser les tests nécessaires à la validation ou à la mise en production d'éléments adaptés ou développés	P2B.1.6.2	Les tests d'intégration sont réalisés ;
		P2B.1.6.3	Un outil collaboratif de gestion des itérations de développement et de versions est utilisé ;
		P2B.1.6.4	Une documentation des versions vient appuyer l'intégration continue ;
B2B.3 Gérer les données		B2B.3 Gérer les données	
B2B.3.1	Exploiter des données à l'aide d'un langage de requêtes	P2B.3.1	L'exploitation des données permet de construire l'information attendue.
B2B.3.2	Développer des fonctionnalités applicatives au sein d'un système de gestion de base de données (relationnel ou non)	P2B.3.2	Les accès aux données sont contrôlés conformément aux habilitations définies par le cahier des charges.
		P2B.3.3	Les traitements pris en charge par les composants développés dans la base de données sont conformes aux demandes du cahier des charges.

Glossaire

Ajax : Le terme "ajax" fait référence à "Asynchronous JavaScript and XML". Il s'agit d'une technique de développement web utilisée pour créer des applications web interactives. Avec AJAX, les données peuvent être échangées avec le serveur web en arrière-plan sans avoir besoin de recharger toute la page.

Ternaire : L'opérateur ternaire est un concept utilisé dans de nombreux langages de programmation, y compris JavaScript, Python, et bien d'autres. C'est une forme condensée d'une instruction if-else, utilisée lorsque vous avez besoin d'effectuer une évaluation conditionnelle simple.

Get : Ce terme est souvent utilisé dans le contexte des requêtes HTTP, où la méthode "GET" est l'une des méthodes standard utilisées pour récupérer des données à partir d'un serveur web.

Put : Le terme "put" est également couramment utilisé dans le contexte des requêtes HTTP. Contrairement à la méthode "GET" qui est utilisée pour récupérer des données, la méthode "PUT" est utilisée pour envoyer des données à un serveur web afin de mettre à jour une ressource existante.

Tableau multidimensionnelles : Un tableau multidimensionnel, également appelé tableau à plusieurs dimensions, est une structure de données qui stocke des données dans plusieurs dimensions. Contrairement aux tableaux unidimensionnels, qui ne contiennent qu'une seule liste de valeurs, les tableaux multidimensionnels peuvent contenir plusieurs listes de valeurs organisées en une ou plusieurs dimensions.

JSON : JSON (JavaScript Object Notation) est un format de données léger, basé sur du texte, qui est largement utilisé pour l'échange de données sur le web. Il est facile à lire et à écrire pour les humains, tout en étant facile à analyser et à générer pour les machines.

JSONViewer : JSONViewer est un outil utilisé pour visualiser les données JSON de manière lisible et organisée dans un navigateur web. Il est souvent présenté sous la forme d'une extension de navigateur ou d'un module complémentaire qui peut être installé dans différents navigateurs tels que Chrome, Firefox, etc.

Bibliographie/Webographie

<https://api.jquery.com/jquery.ajax/>

<https://www.php.net/manual/fr/reserved.variables.get.php>

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Operators/Conditional_operator

présentation ajax :

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=video&cd=&cad=rja&uact=8&ved=2ahUKEwjBldynzr-FAXV_UaQEHWzCD2sQz40FegQIEBAg&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DJT-Rq_16HOk&usg=AOvVaw1o4tGHrXCIV8qEZ6xk0M08&opi=89978449

<https://www.pierre-giraud.com/php-mysql-apprendre-coder-cours/tableau-multidimensionnel/>

<https://www.php.net/manual/fr/features.file-upload.put-method.php>

<https://www.conseil-webmaster.com/formation/php/09-variables-get-php.php>

<https://stackoverflow.com/questions/377644/jquery-ajax-error-handling-show-custom-exception-messages>

<https://www.journaldunet.fr/developpeur/developpement/1202527-comment-faire-apparaître-un-message-d-exception-spécifique-dans-un-message-d-erreur-jquery-ajax/>

<https://www.npmjs.com/package/@andypf/json-viewer>