Compte rendu d’activités

Réalisé lors du stage de 2e année à la DGFiP

BTS SIO SLAM
Quentin Saujot

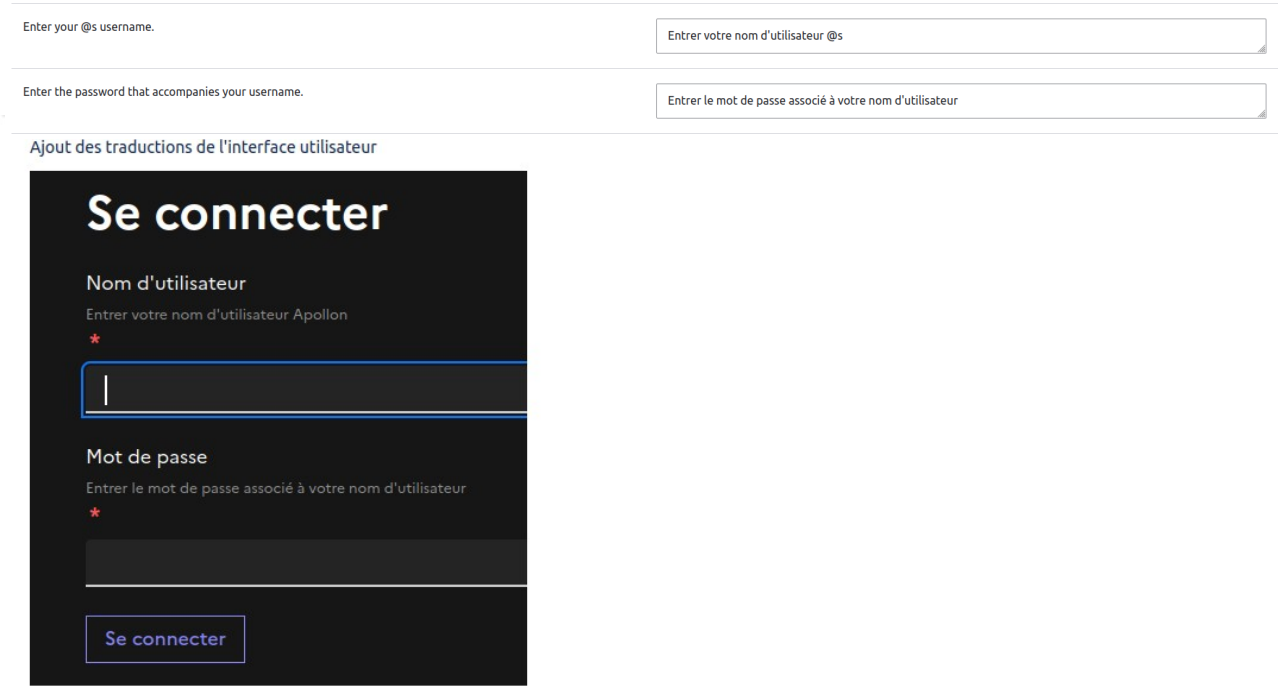
Environnement de travail :

Dans le cadre de ce stage, j'ai intégré l’équipe PGC (Portails Gestion de Contenus) pour contribuer au projet Apollon, une application web développée en Drupal 10. On m’a confié un ordinateur avec un OS Ubuntu, et comme éditeur de code AptanaStudio3. La méthodologie du projet Apollon étant l’agilité, l’outil de gestion de tickets Jira/Confluence était utilisé. Le code quant à lui était hébergé sur le Gitlab de la DGFiP.

De plus, la communication régulière au sein de l'équipe se faisait via la messagerie interministérielle Tchap ou bien par visio avec le système de webconférence de l'État. Conformément à la méthodologie agile, un DSM (Daily Scrum Meeting) de 15 minutes, auquel je participais, était organisé tous les matins. Durant ces daily, le Product Owner attribuait les tickets aux différents développeurs.

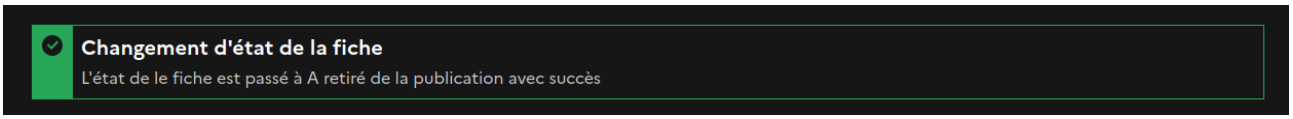
Missions réalisées

En tant que stagiaire, on m’a confié divers tickets d’anomalie de l'application web Apollon. Les deux premiers tickets consistaient à corriger des problèmes de traduction sur l’interface d’Apollon. Pour cela, j'ai dû modifier les paramètres de Drupal pour corriger les traductions de l'interface, puis j'ajoutais un commentaire dans le ticket afin de pouvoir le passer à l'état « À évaluer ».



Le troisième ticket consistait à faire en sorte de modifier un message d'avertissement en fonction du

changement d'état d'une fiche. Pour cela, une modification du code PHP fut nécessaire. J'ai implémenté un double switch pour gérer les différents cas possibles et affiché à la fin le message.




```php
// Add confirmation message.
switch ($new_state) {
  case 'published':
    $titre = "Publication de la fiche";
    $corps = "Votre fiche a été publiée avec succès !";
    break;

  case 'unpublished':
    $titre = "Dé-publication de la fiche";
    $corps = "Votre fiche a été retirée de la publication avec succès !";
    break;

  default:
    $titre = "Changement d'état de la fiche";
    switch ($new_state) {
      case 'draft':
        $corps = "L'état de la fiche est passé à l'état En cours de création avec succès !";
        break;

      case 'to_published':
        $corps = "    L'état de la fiche est passé à l'état A publier avec succès !";
        break;

      case 'to_unpublished':
        $corps = "L'état de la fiche est passé à l'état A Retirer de la publication avec succès !";
        break;

      case 'archived':
        $corps = "Votre fiche a été Archivée avec succès !";
        break;

    }
    break;
}
$message = [
  '#markup' => "<h2 id='apl_sheet_save_success_message' class='fr-alert__title'>{$titre}</h2><p>{$corps}</p>",
];
if (!empty($message)) {
  $message = $this->renderer->render($message);
  $this->messenger()->deleteByType('status');
  $this->messenger()->addStatus($message);
}
}
```

Pour le quatrième ticket, j'ai dû faire en sorte que le titre d'une fiche apparaisse lors de l'avertissement de la suppression de ladite fiche des favoris. Pour cela, j'ai ajouté la fonction getTitle dans un fichier PHP et fait une modification d'un fichier de configuration pour qu'elle soit prise en charge.

```
   /**
   * Get page title of form.
   *
   * @param \Drupal\node\NodeInterface $node
   *   The node of the context of the form.
   *
   * @return \Drupal\Core\StringTranslation\TranslatableMarkup
   *   The result.
   */
  public function getTitle(NodeInterface $node = NULL) {
    if (NULL === $node) {
      return $this->t('Supprimer la fiche des favoris');
    }

    return $this->t('Supprimer la fiche "@title" des favoris', ['@title' => $node->label()]);
  }
```

```
apollon_favorite.delete_sheet_favorite:
  path: '/favorite/delete/{node}/{apollon_favorite}'
  defaults:
    _form: '\Drupal\apollon_favorite\Form\DeleteSheetFavoriteForm'
    _title_callback: '\Drupal\apollon_favorite\Form\DeleteSheetFavoriteForm::getTitle'
  requirements:
```

Pour le cinquième ticket, je devais faire en sorte que les boutons soient alignés sous le titre conformément à la maquette du site. Pour cela, le code HTML a dû être modifié en supprimant une div et en modifiant le style CSS appliqué au ul.

```
            <span class="fr-tag fr-tag--sm ds-bg-blue-light ds-color-white fr-mt-1w">Réponse type</span>
        {% endif %}
      </h1>
    </div>
</div>
<!-- END HEADER PAGE TITLE FICHE -->

        <ul class="fr-raw-list ds-flex ds-flex-baseline ds-flex-justify-end">
            {% if has_edit_access %}
                <li>
                <a
                        class="fr-btn ds-btn fr-btn--tertiary-no-outline fr-btn--icon-left fr-icon-edit-fill"
                        href="{{ path('entity.node.edit_form',{'node' : node_id}) }}">
                        Modifier
                        <span class="fr-sr-only">la fiche</span>
                    </a>
                </li>
            {% endif %}
            {% if has_delete_access %}
                <li>
                <a
                        class="fr-btn ds-btn fr-btn--tertiary-no-outline fr-btn--icon-left fr-icon-delete-fill"
                        href="{{ path('entity.node.delete_form',{'node' : node_id}) }}">
                        Supprimer
                        <span class="fr-sr-only">la fiche</span>
                    </a>
                </li>
            {% endif %}
```

Pour chaque ticket, un workflow devait être suivi : une fois le ticket attribué par la Product Owner, il passait à l'état « Évaluation » pour que le développeur reproduise l'anomalie et vérifie qu'elle est toujours existante ; puis il passait à « En cours », là le développeur appliquait une première solution et si cela concernait du code, le code devait être revu par un autre développeur, dans mon cas le tech leader, et si le code était validé, il était mergé dans la branche principale de Gitlab pour déploiement sur l'environnement de développement ; enfin, le ticket passait à « À tester » pour que la Product Owner valide ou non la disparition de l'anomalie.

À la moindre difficulté, je pouvais joindre l'un des membres de l'équipe pour m'aider.

À la fin du sprint, j'ai participé aux cérémonies de fin de sprint, dans le cadre de la méthode agile : la démo du sprint qui consistait à présenter le produit construit durant le sprint, ensuite la review du sprint, qui consistait à faire le point sur le sprint écoulé en regardant les attendus et les résultats finaux, et pour finir une rétrospective pour échanger avec l'équipe, afin d'identifier les forces, faiblesses, succès et échecs de l'équipe pour évoluer en continu.