

Projet BTS : Le Jumeau Numérique

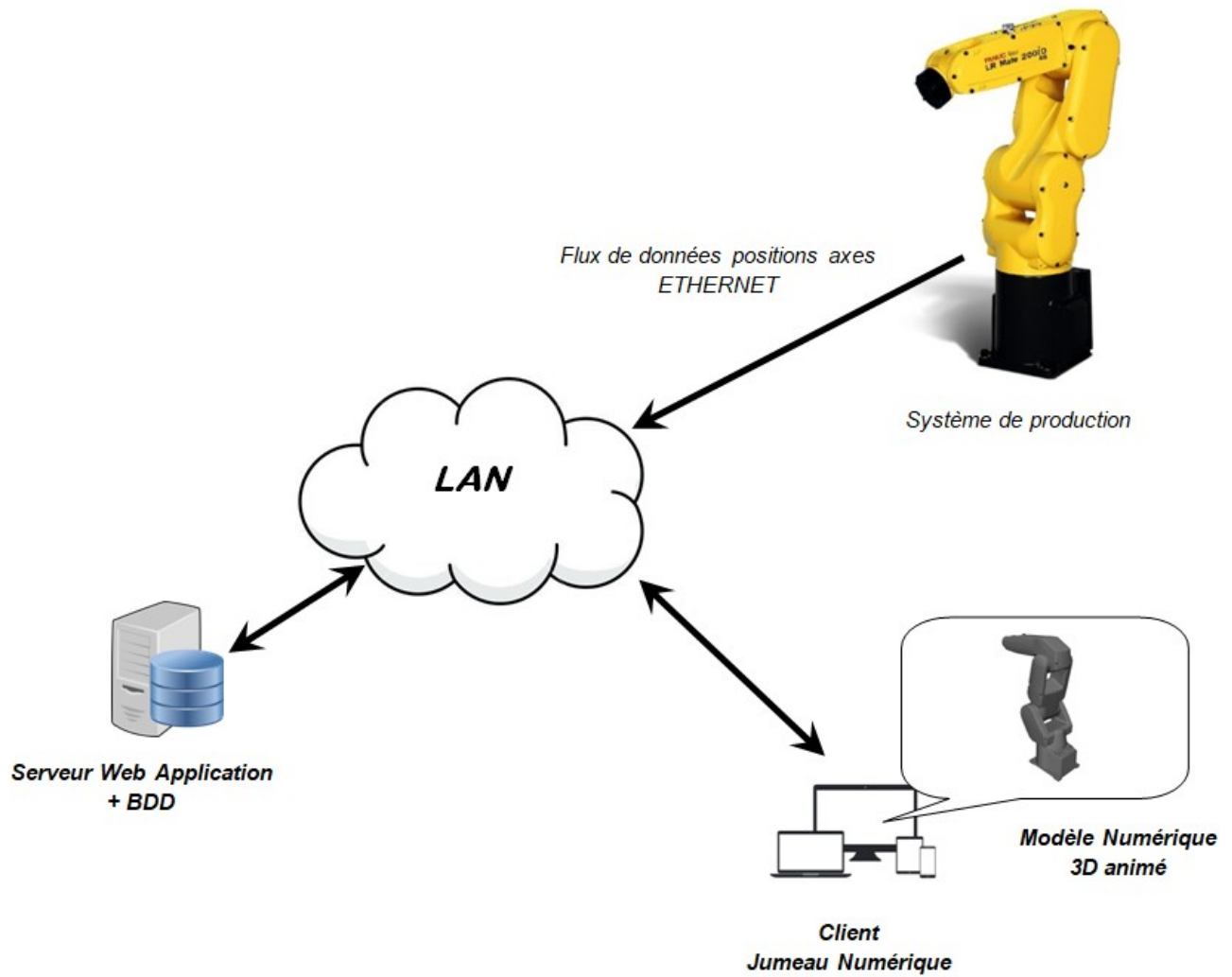


Table des matières

I-Introduction :	3
II-Documentation technique partie séquences jumeau.....	4
A) Dépendances installées :	4
B) Contrôleur SequenceController.php :	4
C) Les dépendances incluses :	6
D) Fonctions dans lesquelles on utilise les dépendances:	6
E) Les extensions twig : Localisées dans « src/twig/ ».....	6
F) Templates dans lesquels elles sont appelées :.....	7
G) Les extensions js : localisées dans « public/js/sequences/ ».....	7
H) Templates dans lesquels elles sont appelées :.....	7

I-Introduction :

Le projet qui doit être réalisé est de suivre à distance en temps réel sur un modèle numérique 3D l'état d'un système de production qui pour nous sera le robot Fanuc ds-Irmate-200id-4s.

Tâche numéro 4 : GESTION BDD / TRAITEMENT REQUETES /
ENREGISTREMENT LECTURE SEQUENCES JUMEAU

Énoncé des tâches à réaliser par les étudiants : (Prévisionnel*)

ETUDIANT 1 :

SERVEUR WEB / INTEGRATION INFRASTRUCTURE / SECURITE /
PROFILS UTILISATEURS / COMPTE ADMINISTRATEUR

Framework Symfony
PHP



ETUDIANT 2 :

AFFICHAGE/ANIMATION 3D DU JUMEAU NUMERIQUE

JavaScript
Bibliothèque three.js
Framework Symfony



ETUDIANT 3 :

GESTION/TRAITEMENT FLUX DE DONNEES SYSTEME REEL

PHP
JavaScript
Framework Symfony



ETUDIANT 4 :

GESTION BDD / TRAITEMENT REQUETES / ENREGISTREMENT
LECTURE SEQUENCES JUMEAU

PHP
Java
XML
Framework Symfony



II-Documentation technique partie séquences jumeau

A) Dépendances installées :

-symfony/orm-pack

Comment l'installer : Pour Eclipse, faire un clic droit sur le nom du projet puis aller sur « show in » puis « terminal ». Ensuite dans le terminal, taper : **composer require symfony/orm-pack**.

L'endroit où je l'ai trouvé : <https://symfony.com/doc/current/doctrine.html>

-symfony/form

Comment l'installer : Pour Eclipse, faire un clic droit sur le nom du projet puis aller sur « show in » puis « terminal ». Ensuite dans le terminal, taper : **composer require symfony/form**.

L'endroit où je l'ai trouvé : <https://symfony.com/doc/current/forms.html>

-symfony/filesystem

Comment l'installer : Pour Eclipse, faire un clic droit sur le nom du projet puis aller sur « show in » puis « terminal ». Ensuite dans le terminal, taper : **composer require symfony/filesystem**.

L'endroit où je l'ai trouvé : <https://symfony.com/doc/current/components/filesystem.html>

OU

dans le terminal sur Eclipse on tape la commande : **composer install** et ça va installer toutes les dépendances nécessaires.

B) Contrôleur SequenceController.php :

Dans ce contrôleur, il y a plusieurs fonctions :

Leur utilité :

-**accueilSequence()** : redirige vers le template «accueilSequence.html.twig» qui .

-VerifNom(\$var) : Prend en paramètre d'entrée la chaîne de caractères saisie et vérifie si le nom a déjà été saisi avant et si c'est le cas, on rajoute un chiffre au nom de la séquence qui s'incrémente automatiquement et si rien n'est saisi, la séquence est nommée automatiquement « sequence ». Elle retourne le nom du fichier.

-fluxJson(\$var) : Prend en paramètre d'entrée les flux de données et modifie la structure des flux de données pour avoir plusieurs tableaux associatifs clé-valeur ayant chacun 6 valeurs. Retourne le tableau encodé en json.

-rename(\$nomFichierSequence, Request \$request, SequenceRepository \$repo,

EntityManagerInterface \$manager) : Permet de renommer la séquence donnée en premier paramètre d'entrée en utilisant la fonction VerifNom() qui permet de vérifier si le nom par lequel on veut renommer la séquence a déjà été utilisé une fois avant.

Le paramètre d'entrée \$repo permet de récupérer dans la base de données la séquence ayant le nom donné en premier paramètre.

Le paramètre d'entrée \$manager permet d'enregistrer la séquence dans la base de données une fois le nom modifié.

A la fin de la fonction, si on a soumis le formulaire, on est renvoyé vers la route

«sequence_selectionnerSequence» qui récupère tous les éléments de la base de données et ensuite retourne le template « formulaireSequence.html.twig » qui va afficher toutes les séquences ainsi que la séquence renommée.

-enregistrer(\$nomFichierSequence=null, Request \$request, SequenceRepository \$repo,

EntityManagerInterface \$manager): permet d'enregistrer la séquence données en premier paramètre en utilisant la fonction VerifNom() qui permet de vérifier si le nom a déjà été tapé une fois et la fonction fluxJson() qui permet de modifier la structure du fichier json en mettant plusieurs tableaux associatifs clé valeur toutes les six valeurs.

Le paramètre d'entrée \$repo permet de récupérer dans la base de données la séquence ayant le nom donné en premier paramètre.

Le paramètre d'entrée \$manager permet d'enregistrer l'url de la séquence dans la base de données.

A la fin de la fonction, si on a soumis le formulaire, on est renvoyé vers la route

«sequence_selectionnerSequence» qui récupère tous les éléments de la base de données et ensuite retourne le template « formulaireSequence.html.twig » qui va afficher toutes les séquences.

-afficher(\$var) : Le premier paramètre est le fichier que l'on veut afficher. Permet de récupérer le contenu du fichier json qui contient les séquences et de l'afficher. A la fin de la fonction, on est renvoyé vers le template « afficher.html.twig ».

-selectionnerSequence(SequenceRepository \$repo) : Sélectionne toutes les séquences dans la table « sequence » et les renvoie sur le template « formulaireSequence.html.twig » avec le repository.

-traitement(EntityManagerInterface \$manager, SequenceRepository \$repo):

Le paramètre d'entrée \$repo permet de récupérer dans la base de données la séquence ayant le nom donné en premier paramètre.

Le paramètre d'entrée \$manager permet d'enregistrer l'url d'une séquence dans la base de données .

Récupère les informations des boutons du template «formulaireSequence.html.twig».

-suppressionFichier(\$var, EntityManagerInterface \$manager, SequenceRepository \$repo) :
Permet de supprimer les séquences et de supprimer l'enregistrement de la base de données.

C) Les dépendances incluses :

```
use Symfony\Component\Filesystem\Filesystem;  
use Symfony\Component\Form\Extension\Core\Type\HiddenType;  
use Symfony\Component\Form\Extension\Core\Type\SubmitType;  
use Symfony\Component\Form\Extension\Core\Type\TextType;
```

D) Fonctions dans lesquelles on utilise les dépendances:

-Filesystem : on l'utilise dans la fonction enregistrer pour pouvoir créer des dossiers.

-Form\Extension\Core\Type\HiddenType : dans la fonction enregistrer() pour ne pas afficher le champ 'data' ça ne sert de l'afficher car les données sont récupérées automatiquement.

- Form\Extension\Core\Type\SubmitType : dans la fonction renommer() pour créer un bouton.

- Form\Extension\Core\Type\TextType : dans la fonction renommer() pour spécifier que ce que l'on marque est du texte.

E) Les extensions twig : Localisées dans « src/twig/ »

jsonMultiAsso qui correspond au fichier « jsonMultiAsso.php » permet de transformer un tableau en tableaux associatifs clé-valeur ayant chacun six valeurs.

-affichageSequence qui correspond au fichier « affichageSequence.php » permet de séparer une url en fonction du séparateur de linux ou de celui de windows.

F) Templates dans lesquels elles sont appelées :

-jsonMultiAsso, qui correspond au fichier « jsonMultiAsso.php », est appelé dans le template « afficher.html.twig.

-affichageSequence, qui correspond au fichier « affichageSequence.php », est appelé dans le template.

G) Les extensions js : localisées dans « public/js/sequences/ »

- sequence.js qui permet de présenter des tableaux plus facilement.

-enregistrer_sequence.js qui permet de se connecter au serveur du robot qui affiche les données en local.

H) Templates dans lesquels elles sont appelées :

-sequence.js : il est appelé dans le template « formulaireSequence.html.twig » car dans le fichier « sequence.js » il y a un plugin jquery qui permet de présenter des tableaux plus facilement avec des fonctionnalités existantes comme la pagination, le tri que ce soit par date, par ligne...

- enregistrer_sequence.js : il est appelé dans le template « save.html.twig » parce que c'est le fichier « enregistrer_sequence.js » qui va accéder au serveur du robot et récupérer les données.