**Compte rendue TD3 KIM Sulyvan RT112**

Question 1 : Que signifie JPA ? Quel est son intérêt ?

* **JPA signifie « Java Persistence API » il s’agit d’une interface de programmation permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java.**

Question 2 : Quels sont les deux modèles ?



Question 3 : A quoi sert le ”@Id” dans la classe Product ?

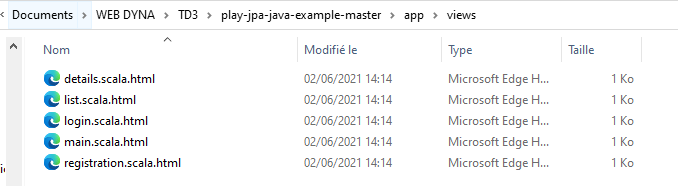
* **Le « @id » spécifie la clé primaire d’une entité c’est un identifiant unique.**

Question 4 : A quoi correspond le dossier validators ?

* **Le dossier validators correspond à**

**1.3 Présentation de la vue (dans app/views)**

Question 5 : Listez les 5 vues.



Question 6 : Quelle est l’utilité de details.scala ?

* **L’utilité de details.scala sert à donner les détails d’un produit en donnant son nom, la description, une photo etc…**

Question 7 : Que fait listing 1 dans list.scala ?

* **Le listing 1 dans list.scala permet d’afficher les détails d’un produits.**



Question 8 : Que fait listing 2 dans list.scala ?

* **Le listing 2 dans list.scala permet de supprimer un produits**



Question 9 : Quelle est la différence entre login.scala et registration.scala ?

* **Le « login.scala » permet de se connecter à son compte grâce à un email et un mot de passe, tandis que « registration.scala » permet de se créer un compte avec un email et en mettant un mot de passe.**

**1.4 Présentation des contrôleurs**

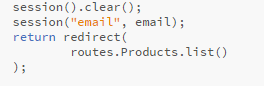
Question 10 : Quelles sont les méthodes de la classe Application dans le dossier conf en mentionnant leur utilité ?



**Cette méthode permet d’afficher une page de connexions.**



**Cette méthode permet d’écrire en chaîne de caractère l’email et le mot de passe, si l’email ou le mot de passe n’est pas le bon il retournera un message d’erreur.**



**Cette méthode une fois la connexions établis affichera la liste des produits.**

Question 11 : Dans Products.java, quel type de structure de données contient les produits ? Quels sont ses avantages et inconvénients par rapport au tableau ? par rapport au vecteur ?

* **Dans product.java, le type de structure de données est un « ArrayList » l’avantage c’est que nous n’avons pas besoin d’indiquer sa taille elle est plus flexible**, **il augmente à mesure que nous ajoutons des éléments et qu’il se rétrécit au fur et à mesure que vous en retirez des éléments. Le désavantage est que le tableau peut contenir à la fois des valeurs primitives et des objets, mais « ArrayList » ne peut contenir que des objets.**

Question 12 : Que fait listing 3 dans users.java ?



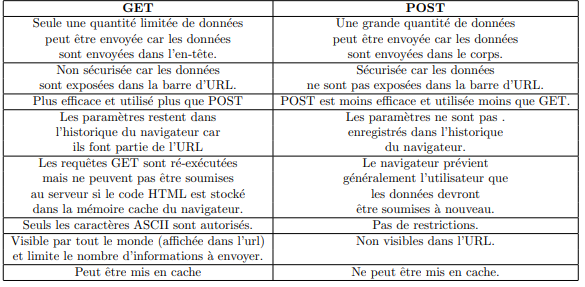
* **La listing 3 enverra un email comme quoi c’est bon et retournera sur la page de connexion.**

Question 13 : Détaillez un schéma UML avec les classes de contrôleur et les modèles.

Question 14 : Quelle est utilité de /:ean dans listing 4 ?

* **L’utilité de / :ean dans listing 4 sert à la génération de code barre.**

Question 14 : Quelles sont les différences entre une route avec GET et celle avec POST ?



Question 15 : Que fait cette route dans listing 5 ?

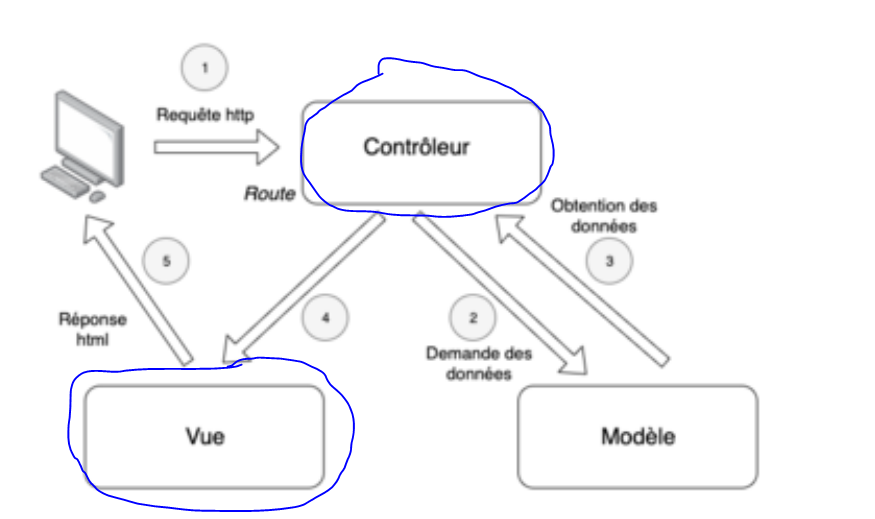


**Elle supprime le code barre du produit.**

Question 17 : A quoi sert le dossier dao dans application ?

**Le dossier DAO signifie Data Access Object, c’est pour la persistance**

Question 18 : Schématiser les différentes relations entre le contrôleur, la vue et l’utilisateur en utilisant les routes et les actions définies dans ce programme.



Question 19 : Réaliser l’UML correspondant au projet final que vous avez choisi, en vous inspirant de la question 13 (ce modèle peut varier dans le temps et selon le membre du groupe).

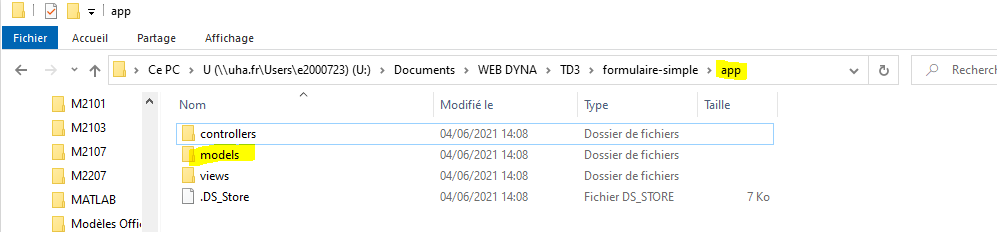
**//**

Question 20 : Réaliser les relations entre contrôleur/utilisateur/vue correspondant au projet final que vous avez choisi, en vous inspirant de la question 18 (ce modèle peut varier dans le temps et selon le membre du groupe).

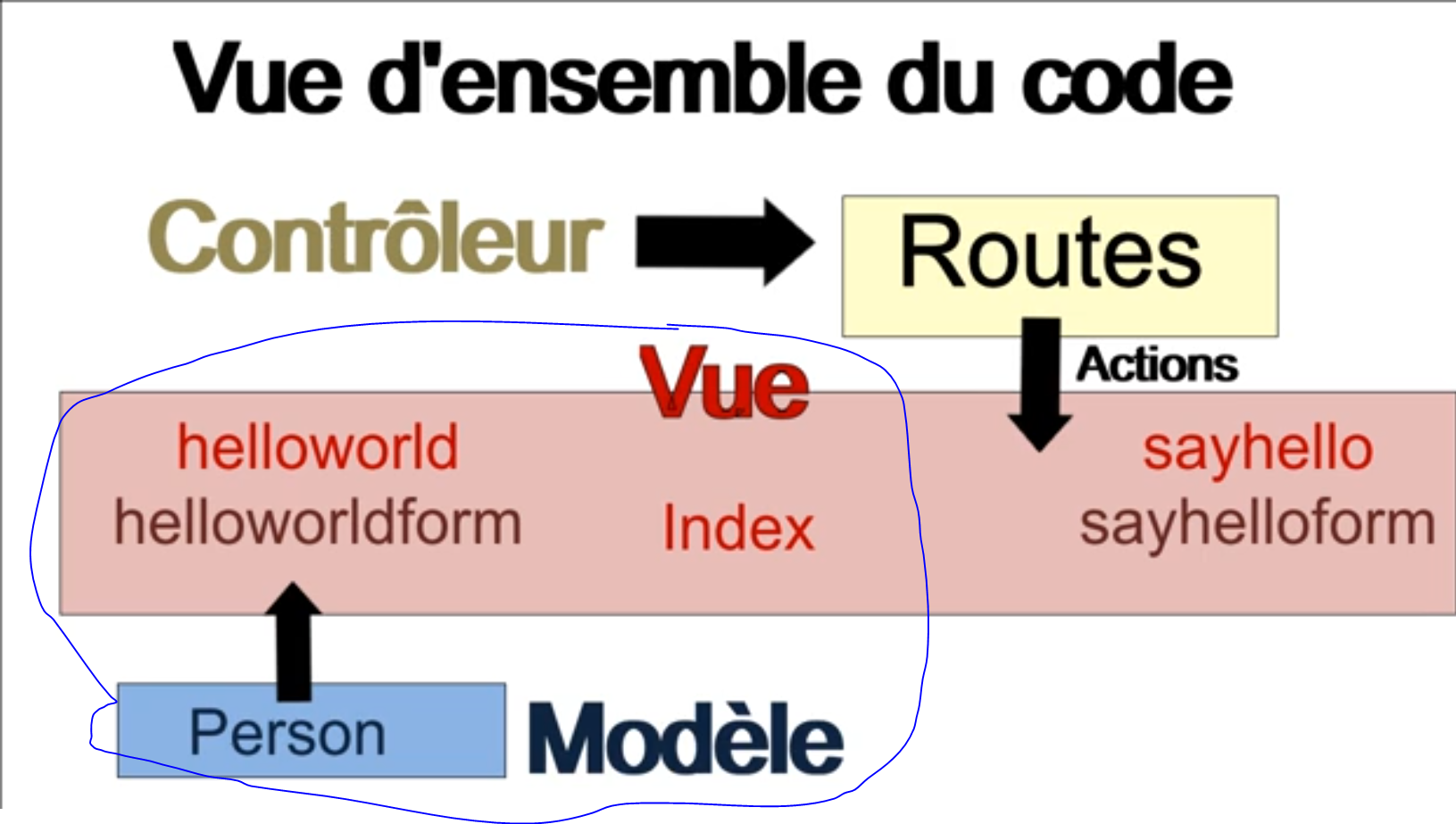
///

**TD3 PRATIQUE**

Exercices : Il faut d’abord rajouter le package models **dans app**~~-conf-lib-public-text~~ (entourer le dossier correct (0.25 pt).



Afin de pouvoir l’utiliser, il est nécessaire de lier la partie modèle à la partie ~~modèle~~-**vue**-~~contrôleur~~ (entourer la réponse correcte, 0.25 pt).



Analysez ensuite ce code (il faut l’écrire pour cela). Les imports permettent de lier les différents éléments. Précisez dans le tableau ci-dessous leur utilité (3 pts) :

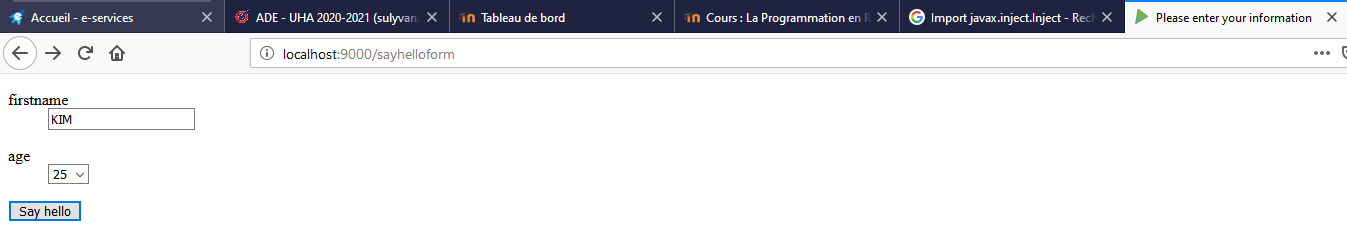
|  |  |
| --- | --- |
| Import java.until.List ; | **Elle permet d’importer une ArrayList** |
| Import play.mvc.\* ; | **Elle permet d’importer le fichier play package qui contient tout le nécessaire pour écrire des contrôleurs.** |
| Import play.data.\* ; | **Elle permet d’importer le fichier play package qui contient tous les data.** |
| Import javax.inject.Inject ; | **Elle permet d’importer une injection de dépendance** |
| Import views.html\* ; | **Elle permet d’importer la page html views** |
| Import models.Person ; | **Elle permet d’importer le modèle Person** |

Pour faire marcher le code, quel chemin faut-il rajouter au fichier route dans le dossier conf (0.5pt) ?



**Il faut rajouter cette ligne.**

Vérifiez que le code fonctionne avant de passer 0 la suite (N’oubliez pas la route pour l’action helloworldform ).



Que fait le code ModelFormulaire ci − dessus (0.5pt) ?

**Le code formulaire affichera comme quoi le firstname doit commencer avec une majuscule et un minimum de 2 caractères, le @min est une contrainte elle doit être situé entre 18 et 25**

Que fait le code CodeMystere (0.5pt) ?

**Le codeMystere vérifie les informations des requête du formulaire, si il y’a une erreur il retournera un message d’erreur.**