# Sujet d’examen : Développement d’APIs Bancaires

**Contexte :**

Vous travaillez en tant que développeur dans une banque, et vous êtes chargés de concevoir et de développer des services web pour permettre à vos clients et partenaires d’interagir avec leur compte bancaire. Vous devez documenter les API, implémenter une solution, et optimiser les performances de services.

**Partie 1 : Développement de l’API RESTful**

1. Conception du service RESTful :

* Développez un webservice RESTful pour la gestion des comptes bancaires
* Le service doit inclure à minima les fonctionnalités suivantes :
* **Consultation du solde du compte (‘GET /accounts/{accountId}/balance’)**
* **Historique des transactions (‘GET /accounts/{accountId}/transactions’)**
* **Effectuer un virement (‘POST /accounts/{accountId}/transfer’)**
* Implémentez la pagination pour l’historique des transactions afin d’améliorer les performances.

1. Documentation :

* Utilisez **Swagger** et **OpenAPI** pour documenter l’API REST. Assurez-vous que tous les endpoints, les paramètres, et les réponses sont bien décrits.

**Partie 2 : Développement l’API SOAP**

1. Conception du service SOAP :
   * Développez un service SOAP permettant de réaliser les mêmes opérations que le service RESTful :
     + **Consultation du solde du compte (‘getBalance(accountId)’)**
     + **Historique des transactions (‘getTransactions(accountId, pageNumber, pageSize)’)**
     + **Effectuer un virement (‘transferFunds(creditor, debtor, amount, currency)’)**
   * Ajoutez une gestion de la **pagination** dans le service ‘**getTransactions’**.
2. Créez un fichier **WSDL** complet pour décrire le service SOAP, en incluant les schémas XML pour les requêtes et les réponses.

**Partie 3 : Evaluation et Comparaison**

1. Comparez les deux services en termes de performance, de facilité d’implémentation et d’interfaçage.
2. Discuter des avantages et des inconvénients de chaque approche (REST vs SOAP) dans le contexte d’une banque en ligne.