

Documentation de l'Application de Déduplication

Architecture du Système

Composants du Service

1. Frontend

- Gère toutes les interactions avec l'interface utilisateur
- Fournit des affichages visuels pour le suivi des processus et des résultats

2. Backend

- Traite les requêtes provenant du frontend
- Gère la communication avec la base de données de l'application
- Fait l'interface avec le service T4Face pour les opérations biométriques
- Exécute toute la logique métier

Ressources du Service

Base de Données de l'Application

La base de données doit être modélisée et stockera:

1. Données du Processus de Déduplication

- ID du processus (identifiant unique)
- Date et heure de création
- Nom d'utilisateur du créateur
- Date et heure de début du traitement
- Date et heure de fin du traitement
- Statut (prêt à démarrer, en traitement, terminé, en pause, etc.)
- Nom d'utilisateur de nettoyage (utilisateur ayant initié le nettoyage)

- Date et heure de nettoyage

2. Fichiers du Processus

- Statut (téléchargé, inséré, conflit, erreur, supprimé, etc.)
- Date et heure de téléchargement
- Date et heure d'insertion dans T4Face
- Date et heure de suppression dans T4Face

3. Étapes du Processus

- Nom de l'étape (Insertion, Identification, etc.)
- Date et heure de début
- Date et heure de fin

4. Données d'Exception

- ID d'exception
- Nom du fichier
- Nom du fichier candidat
- Score de comparaison
- ID du processus de déduplication

5. Données de Conflit

- ID de conflit
- Date de création
- Nom du fichier et images associées
- ID du processus de déduplication

Service T4Face

- Moteur facial T4ISB qui gère les requêtes biométriques
- Communication via API HTTP
- Responsable de la correspondance et de l'identification des visages

Fonctionnalités Principales

1. Télécharger un Fichier ZIP

1. Extraire les fichiers dans le système de fichiers dans un répertoire nommé avec l'ID du processus de déduplication
 - Le répertoire peut avoir des indicateurs de statut temporaires
 - Le système peut nettoyer les répertoires temporaires après une période d'inactivité
 - Le statut du répertoire se met à jour vers "final" lorsque toutes les entrées de la base de données sont stockées avec succès
2. Créer un nouvel enregistrement dans la base de données pour le processus de déduplication avec:
 - ID du processus
 - Nom d'utilisateur du téléchargeur
 - Statut: "en téléchargement"
 - Horodatage de création
3. Pour chaque fichier extrait:
 - Ajouter un enregistrement dans la base de données avec le statut "téléchargé"
 - Associer au processus de déduplication actuel
4. Lorsque tous les fichiers sont traités, mettre à jour le statut du processus de déduplication à "prêt à démarrer"

2. Démarrer la Déduplication

Cette fonction lance un processus en arrière-plan qui:

1. Crée un enregistrement d'étape "Insertion" dans la base de données avec:
 - Horodatage de création actuel
 - ID du processus de déduplication associé
 - Met à jour le statut du processus à "en traitement"

2. Pour chaque fichier téléchargé:

- Interroger la base de données pour vérifier si le même nom de fichier existe dans d'autres processus avec le statut "inséré"
- Si une correspondance est trouvée:
 - Envoyer une demande de vérification à l'API T4Face (`POST /facematcher/verify_64`)
 - Si la vérification indique des personnes différentes, ajouter un enregistrement de conflit et définir le statut du fichier à "conflit"
- Si aucune correspondance n'est trouvée:
 - Ajouter le visage à la base de données en utilisant le nom de fichier comme champ `user_name`

3. Pour les insertions réussies:

- Mettre à jour le statut du fichier à "inséré"
- Stocker l'ID retourné par la réponse de l'API T4Face

4. Après avoir traité tous les fichiers, mettre à jour l'étape d'Insertion avec l'horodatage de fin

5. Créer un enregistrement d'étape "Déduplication" avec l'horodatage de création actuel

6. Pour chaque fichier inséré:

- Envoyer une demande d'identification à l'API T4Face (`POST /identify`)
- Passer les données d'image de l'ID retourné lors de l'insertion
- Si des candidats sont trouvés (à l'exclusion de l'ID source):
 - Ajouter un enregistrement d'exception dans la base de données avec:
 - Horodatage actuel
 - Nom de fichier actuel
 - Liste des noms de candidats (autres noms de fichiers)

Note Importante: Le système doit valider les candidats ayant la même relation d'ID. Par exemple, si l'ID 1 montre une forte similarité avec l'ID 3, lorsque l'ID 3 est

ensuite comparé à la base de données, il montrera également une similarité avec l'ID 1. Comme cette relation a déjà été stockée dans la base de données d'exceptions, elle peut être ignorée pour éviter les enregistrements en double.

3. Mettre en Pause la Déduplication

1. Mettre à jour le statut du processus à "en pause" dans la base de données
2. Arrêter le processus en arrière-plan

4. Reprendre la Déduplication

1. Mettre à jour le statut du processus à "en traitement" dans la base de données
2. Redémarrer le processus en arrière-plan
3. Continuer à partir de la dernière tâche exécutée

5. Nettoyer la Déduplication

Lance un processus en arrière-plan qui:

1. Interroge TOUS les processus de déduplication dans la base de données
2. Pour chaque processus, récupère tous les fichiers associés
3. Envoie des requêtes DELETE à T4Face pour chaque fichier en utilisant l'ID retourné lors de l'insertion
4. Met à jour le statut du fichier à "supprimé" dans la base de données
5. Après avoir traité tous les fichiers, met à jour le processus de déduplication avec:
 - Horodatage de nettoyage
 - Nom d'utilisateur du demandeur

6. Afficher Toutes les Déduplications

Récupère tous les processus de déduplication de la base de données

- Peut être filtré par:
 - Statut
 - Plage de dates

- Nom d'utilisateur
- Autres paramètres selon les besoins

7. Rechercher des Candidats par ID de Personne

1. Interroge les exceptions dans la base de données (créées pendant l'étape d'identification) par IDN (nom de fichier)
2. Renvoie une liste de candidats avec:
 - Images faciales
 - Empreintes digitales (si disponibles)
 - Scores de comparaison
 - Rapports PDF
 - Autres données pertinentes

8. Vérifier les Conflits (1:1)

Interroge tous les enregistrements de conflit

- Peut être filtré par IDN (identifiant)
- Utilise le point de terminaison `POST /verify_64` pour la vérification

Communication API

Le système s'appuie sur les points de terminaison de l'API T4Face, notamment:

- `POST /facematcher/verify_64` : Pour la vérification faciale
- `POST /identify` : Pour l'identification faciale
- Point de terminaison DELETE: Pour supprimer des entrées de T4Face

Chaque interaction API est enregistrée avec des horodatages et des résultats dans la base de données de l'application.

POST /verify_64

```

verifyFaces(
  imageData: string,
): Observable<ICompareFacesResult> | undefined {
  var ident = { person_face: imageData, person_name: 'exemple nizar' };

  return this.http
    .post<ICompareFacesResult>('https://192.81.212.133:9448' + '/api/v1/facematc
    .pipe(
      retry(5),
      shareReplay(1),
      catchError(
        this.handleError<ICompareFacesResult>(
          'get Face Ident Candidates',
          undefined
        )
      )
    );
}

```

Pour l'intégrer dans la fonctionnalité "Check 1:1" (vérification de conflit) de votre application de déduplication, vous pourriez l'adapter comme suit:

```

identifyFace(
  imageData: string //base64 sans 'image/png;base64,'
): Observable<IFaceIdentResponse> | undefined {
  const exemple = {
    "matcherTransactionOriginId": "0.0.0.0",
    "matcherTransactionOriginName": "unknown",
    "matcherTransactionPersonId": "d66c4f98-ff22-4134-b311-a4d1328c4fdc",
    "matcherTransactionTcn": "bd83b62a-a33e-4127-b8a9-8dbd0a474de4",
    "matcherTransactionToT": "IDE",
    "faces": [
      {

```

```

    ...
  }
]
};

let tcn = uuidv4();
let person_id = uuidv4();
let face_id = uuidv4();
var ident = {
  matcherTransactionOriginId: "0.0.0.0",
  matcherTransactionOriginName: "unknown",
  matcherTransactionPersonId: person_id,
  matcherTransactionTcn: tcn,
  matcherTransactionToT: 'IDE',
  faces: [
    {
      faceId: face_id,
      imageData: imageData,
      position: 'FRONTAL',
    },
  ],
};

return this.http
  .post<IFaceIdentResponse>('https://192.81.212.133:9448' +
    '/api/v1/matcherfacetransactions/addidentificationfrommemtransaction',
    ident
  )
  .pipe(
    retry(5),
    shareReplay(1),
    catchError(
      this.handleError<IFaceIdentResponse>(
        'get Face Ident Candidates',
        undefined
      )
    )
  )

```



```
)  
);  
}
```