# ワークサンプルテスト（バックエンドエンジニア）

損害保険会社は、様々な種類の保険をお客様に提供しています。お客さまは、各保険代理店で保険を契約することが多いのですが、代理店ごとに独自の DB を持ち、情報が散在しています。現在、顧客レベルで全ての情報を統合する、顧客一元化 DB をおよび、その API を企画中です。本ワークサンプルテストでは、その企画のディスカッション、およびプロトタイプを実施します。

**仕様**

1. 保険代理店 A,B,C は、以下のような保険契約 DB を持つ。

   **代理店契約 DB （agent_a.csv, agent_b.csv, agent_c.csv）**
   契約 ID、商品 ID、氏名、電話番号、郵便番号、住所

2. 保険商品を定義した DB もある。

   **契約商品 DB (products.csv)**
   商品 ID、商品名

3. 各代理店の情報を統合し、顧客起点となる以下のような DB を作成したい。上記代理店契約 DB から、以下の DB をつくるためには、契約情報から「名寄せ」を行い、名寄せされた顧客に顧客 ID を紐付ける必要がある。

   **顧客契約 DB (transactions.csv)**
   顧客 ID、商品 ID、代理店 ID

   **顧客 DB (customers.csv)**
   顧客 ID、氏名、電話番号、郵便番号、住所

4. 顧客 ID をキーに、顧客が現在契約中の保険商品一覧と、その詳細を取得する API を作成したい。ただし、以下のセキュリティ要件がある
   ① 代理店は他の代理店で契約されたものにはアクセスできない
   ② 保険会社は全ての情報にアクセスできる

**ディスカッションポイント**

エンジニアとして、上記仕様を元に、ビジネスサイドの人間と、プロトタイプする内容を詳細に詰めてください

（ア）名寄せロジックをどうするか、サンプルデータを見ながら詳細を詰めてください。名寄せは今後想定しないようなパターン（特に住所）が登場する可能性もあります

（イ）API のセキュリティの実現方法ついて議論してください。プロトタイプでどこまで実装するかも決めてください

**プロトタイプ**

ディスカッションで決まった内容をもとに、実際に動くプロトタイプ（サンプルデータ作成を含む）を作成してください。以下プロトタイプの要件となります

1. 代理店契約 DB （A、B、C)のデータを名寄せし、顧客契約 DB と顧客 DB を作成するプログラムを用意してください
2. 2を実行後、顧客 ID をキーとして契約情報一覧が取得できる API を実装してください（可能な範囲でセキュリティ要件を実装してください）
   （ア）Input: 顧客 ID
   （イ）Output: 商品一覧＋代理店情報
      ① 商品 ID、商品名、代理店 ID
3. 実装には以下のテクノロジーの何かを利用して下さい
   （ア）Java
   （イ）Python
   （ウ）Javascript (Node.js)
   （エ）Rust
   （オ）Go
   （カ）Kotllin
4. REST API の実装には、既存のメジャーな Framework を使うこと（可能であれば GraphQL を推奨）
5. PostgreSQL v14 がローカルで実行されていることを想定して下さい
6. 名寄せロジックの Unit Test を実装してください
7. API の Unit Test を実装してください

上記条件を満たすプログラムを作成し、github で公開してください。Readme に、プログラムを実行するための各種前提条件と、実行方法を詳細に記述してください。Mac 環境で検証します。

---

For 面接官

**技術的検証ポイント**

- 名寄せはある程度のロジック能力チェック
- API だけじゃなくて one shot のプログラムもかけるか
- DB の基本的知識
- セキュリティのところ
- API 作成
- サンプルデータ作成と、Unit Test で品質への対応が見える

**ディスカッション検証**

- 名寄せの調整
  - 正規化
    - 前後スペースの削除
    - 全角、半角とか
    - 各種区切り文字 / とか -
    - 電話番号は(090)の括弧とか
    - 3番地二上目 -> 3-2 とか
  - 住所の名寄せ
    - 正規化した上で
    - 完全一致、前方一致、後方一致
    - プラガブルにして欲しい
  - 名寄せした上で何を正しいデータとするか
    - 電話、住所とも長い方を優先
- API セキュリティのところ
  - 認証・認可の話がわかってるか。よりセキュアにするにはどうするか。
  - PostgreSQL なので Row Level Security とかも使える。知ってるか。