Data models

These models are just to help you get started. Feel free to modify or add any new models or fields as you please. The models are purposefully basic to avoid bias and encourage creative solutions.

Placeholder data is used for the examples.

User

Represents the users of the app.

Structure:

```
{
  "id": UUID,
  "name": String,
  "address": String,
  "country": String,
  "house_hold_size": Integer, // Number of family members
}
```

Example:

```
{
  "id": "a1b2c3d4-e5f6-7890-1234-567890abcdef",
  "name": "Alex Doe",
  "address": "123 Main St, Exampletown",
  "country": "Germany",
  "house_hold_size": 4
}
```

Recipe

Contains information about recipes, including ingredients and instructions.

Structure:

```
{
  "id": UUID,
  "name": String,
  "ingredients": [
    "ingredient_name": String,
    "quantity": Number,
  ],
  "portion_quantity": Integer, // E.g., serves 2 persons
  "instructions": Array[String], // Step by step instructions
}
```

Example:

```
{
  "id": "b2c3d4e5-f6a7-8901-2345-67890abcdef0",
  "name": "Simple Salad",
  "ingredients": [
    { "ingredient_name": "Lettuce", "quantity": 1 },
    { "ingredient_name": "Tomato", "quantity": 2 },
    { "ingredient_name": "Cucumber", "quantity": 0.5 }
  ],
  "portion_quantity": 2,
  "instructions": [
    "Wash vegetables.",
    "Chop vegetables.",
    "Mix in a bowl."
  ]
}
```

Article

Represents the individual items sold in the store.

Structure:

```
{
  "id": UUID,
  "name": String,
  "sku": String, // Stock Keeping Unit
  "ingredient_name": String,
  "category": String,
  "nutrition_table": String,
  "nutriscore": String,
  "carbon_footprint": Number, // Assuming a numerical value
  "is_biological": Boolean,
  "description": String,
  "allergy_labels": Array[String], // Allergy warnings
  "image_url": String, // URL to product image
  "is_available": Boolean // Inferred from relationship to Stock
}
```

Example:

```
{
  "id": "c3d4e5f6-a7b8-9012-3456-7890abcdef01",
  "name": "Romaine lettuce",
  "sku": "VEG-LET-001",
  "ingredient_name": "Lettuce",
  "category": "Vegetables",
  "nutrition_table": "Calories: 15, Fat: 0.2g, Carbs: 2.9g, Protein: 1.4g (per 100g)",
  "nutriscore": "A",
  "carbon_footprint": 0.5, // Example value (e.g., kg CO2e)
  "is_biological": true,
  "description": "Crisp iceberg lettuce, perfect for salads and sandwiches.",
```

```
    "allergy_labels": [],
    "image_url": "https://placehold.co/300×300/a2d9a1/333333?text=Lettuce",
    "is_available": true
  }
```

Delivery

Represents customer delivery orders and associated details.

Structure:

```
  {
    "delivery_id": UUID, // Main ID for the delivery
    "customer": UUID, // Links to the User
    "recipes": Array[UUID], // Optional: Links to any recipes ordered
    "timeslot": String, // e.g., "2025-04-23 10:00-12:00"
    "trip_id": UUID, // Groups deliveries that are delivered within the same trip
    "hub_id": String, // Local distribution hub
    "fc_id": String, // Fulfillment Center
    "orderline_ids": UUID[] // Links to the specific items in the order
  }
```

Example:

```
  {
    "delivery_id": "d4e5f6a7-b8c9-0123-4567-890abcdef012",
    "customer": "a1b2c3d4-e5f6-7890-1234-567890abcdef", // Alex Doe's ID
    "recipes": ["b2c3d4e5-f6a7-8901-2345-67890abcdef0"], // Simple Salad ID
    "timeslot": "2025-04-24 14:00-16:00",
    "trip_id": "f6a7b8c9-d0e1-2345-6789-0abcdef01234"
    "hub_id": "HUB-VIE-01",
    "fc_id": "FC-FRA-01",
    "orderline_ids": [
      "e5f6a7b8-c9d0-1234-5678-90abcdef0123" // Orderline for Lettuce
    ]
  }
```

Orderline

Represents a single item line within an order.

Structure:

```
  {
    "id": UUID,
    "delivery_id": UUID, // Links back to the Delivery
    "sku": String, // Links to the Article ordered
    "quantity": Integer
  }
```

Example:

```
{
  "id": "e5f6a7b8-c9d0-1234-5678-90abcdef0123",
  "delivery_id": "d4e5f6a7-b8c9-0123-4567-890abcdef012", // Links to Alex's delivery
  "sku": "VEG-LET-001", // Lettuce ID
  "quantity": 2
}
```

Stock

Represents the stock level of articles at a specific fulfillment center.

Structure:

```
{
  "sku": String, // Links to Article's SKU
  "fc_id": String, // FC where stock is located
  "stock_location": String, // Specific location inside FC
  "quantity": Integer, // Current stock quantity
  "last_delivery_timestamp": Optional[Timestamp], // Date + time the product needs to be delivered by w.r.t expiration
  "is_marked_imperfect": Boolean // If the stock is flagged as imperfect, e.g. dented cans
}
```

Example:

```
{
  "sku": "VEG-LET-001", // Lettuce SKU
  "fc_id": "FC-FRA-01",
  "stock_location": "BUFF-3-CHILLED",
  "quantity": 42,
  "last_delivery_timestamp": "2025-04-28T12:00:00Z", // ISO 8601 Format
  "is_marked_imperfect": false
}
```

FC

A fulfillment center is where orders for customers get picked into bags in totes, to be sent off towards hubs.

Structure:

```
{
  "id": String,
  "address": String,
  "country": String,
}
```

Example:

```
{
  "id": "FC-FRA-01",
  "address": "Am Terminal 1, Frankfurt",
```

```
    "country": "Germany"
  }
```

Hubs

Starting at hub locations, electric Picnic vehicles (EPVs) travel to deliver groceries to neighborhoods in short trips.

Structure:

```
  {
   "id": String
   "address": String,
   "country": String,
  }
```

Example:

```
  {
   "id": "HUB-VIE-01",
   "address": "Industriestraße 5, Viernheim",
   "country": "Germany"
  }
```

Diagram

```
classDiagram
   class User {
      +UUID id
      +String name
      +String address
      +String country
      +Integer house_hold_size
   }

   class Recipe {
      +UUID id
      +String name
      +Map~String, Number~ ingredients
      +Integer portion_quantity
      +List~String~ instructions
   }

   class Article {
      +UUID id
```

```
        +String name
        +String sku
        +String ingredient_name
        +String category
        +String nutrition_table
        +String nutriscore
        +Number carbon_footprint
        +Boolean is_biological
        +String description
        +List~String~ allergy_labels
        +String image_url
        +Boolean is_available
    }

    class Delivery {
        +UUID delivery_id
        +UUID customer
        +List~UUID~ recipes
        +String timeslot
        +UUID trip_id
        +String hub_id
        +String fc_id
        +List~UUID~ orderline_ids
    }

    class Orderline {
        +UUID id
        +UUID delivery_id
        +String sku
        +Integer quantity
    }

    class Stock {
        +String sku
        +String fc_id
        +String stock_location
        +Integer quantity
        +Timestamp last_delivery_timestamp
        +Boolean is_marked_imperfect
    }

    class FC {
        +String id
        +String address
        +String country
    }

    class Hub {
        +String id
        +String address
        +String country
```

```
}

User "1" -- "*" Delivery : has
Article "*" -- "*" Recipe : can be used for
Delivery "*" -- "*" Recipe : includes
Delivery "1" -- "*" Orderline : contains
Delivery "*" -- "1" Hub : assigned to
Delivery "*" -- "1" FC : originates from
Orderline "*" -- "1" Article : references
Stock "*" -- "1" Article : holds
Stock "*" -- "1" FC : located in
```