Meal Planner API Contract (v0.1)

This document summarises the initial public API contract for the meal planner app. It focuses on the core user loop (Discover -> Plan -> Shop -> Home -> Insights). Auth/admin endpoints are out of scope for this version.

1. Discover

1.1 GET /api/v1/discover

Returns personalised discovery content for the current user.

Response (example shape):

```
{
  "dailyBites": [ {recipeSummary...}, ... ],
  "carousel":   [ {recipeSummary...}, ... ],
  "likedRecipeIds": [1,2,3]
}
```

1.2 GET /api/v1/recipes/:id

Returns full details for a given recipe, used across Discover and Plan.

1.3 POST /api/v1/interactions

Records a user interaction with a recipe.

Body:

```
{ "recipeId": number,
  "interactionType": "like"|"dislike"|"view"|"cooked",
  "source": "discover"|"plan"|"home"|"insights" }
```

1.4 POST /api/v1/me/likes

Add a recipe to the user's liked list.

Body: { "recipeId": number }

1.5 DELETE /api/v1/me/likes/:recipeId

Remove a recipe from the user's liked list.

2. Plan

2.1 POST /api/v1/meal-plans

Generate or regenerate a weekly plan for the user.

Body: { "startDate": "YYYY-MM-DD", "forceRegenerate"?: boolean }

Response: full meal plan including meals per day.

2.2 GET /api/v1/meal-plans/current

Returns the meal plan covering today (if any).

2.3 GET /api/v1/meal-plans

Query params: ?from=YYYY-MM-DD&to=YYYY-MM-DD

Returns a list/summary of plans in the given range.

2.4 GET /api/v1/meal-plans/:id

Returns full details of a specific plan.

2.5 POST /api/v1/meal-plans/:id/replace-meal

Replace a single meal in a plan using the reco engine.

Body: { "planMealId": number } or { "date": "YYYY-MM-DD", "mealType": "dinner" }

3. Preferences

3.1 GET /api/v1/users/me/preferences

Fetch the user's saved food preferences/settings.

3.2 PUT /api/v1/users/me/preferences

Create or replace the user's preferences.

Body (example):

```
{
  "dietType": "omnivore"|"vegetarian"|...,
  "caloriesTarget": number,
  "excludeIngredients": [ingredientId...],
  "cuisineLikes": ["thai","mediterranean"],
  "cuisineDislikes": ["british"],
  "effortLevel": "low"|"medium"|"high"
}
```

4. Home

4.1 GET /api/v1/home

Aggregated call for the home screen.

Response (example):

```
{
  "preferencesSummary": { ... },
  "upcomingMeal": { ... },
  "stats": { ... },
  "fact": { "id": string, "text": string }
}
```

5. Shop

5.1 GET /api/v1/shop

Returns the aggregated shopping list for a plan plus a fact.

Query params: ?planId=optional (if omitted, use current plan).

Response (example):

```
{
  "planId": number,
  "categories": [
    { "name": "Vegetables", "items": [ {"ingredientId", "name", "quantity", "unit"},
... ] },
    ...
  ],
  "fact": { "id": string, "text": string }
}
```

6. Insights

6.1 GET /api/v1/insights/weekly

Returns a weekly insights overview for the user.

Query params (optional): ?weekStart=YYYY-MM-DD

Response (example):

```
{
  "fact": { ... },
  "time": { ... },
  "money": { ... },
  "cuisine": { ... },
  "images": [url1, url2, url3]
}
```

6.2 POST /api/v1/insights/reviews

Submit a review of a plan or specific recipes.

Body (example): { "planId"?: number, "recipeId"?: number, "rating": 1-5, "comment"?: string }

6.3 POST /api/v1/insights/suggestions

Submit free-form suggestions/feedback.

Body: { "message": string, "context"?: string }