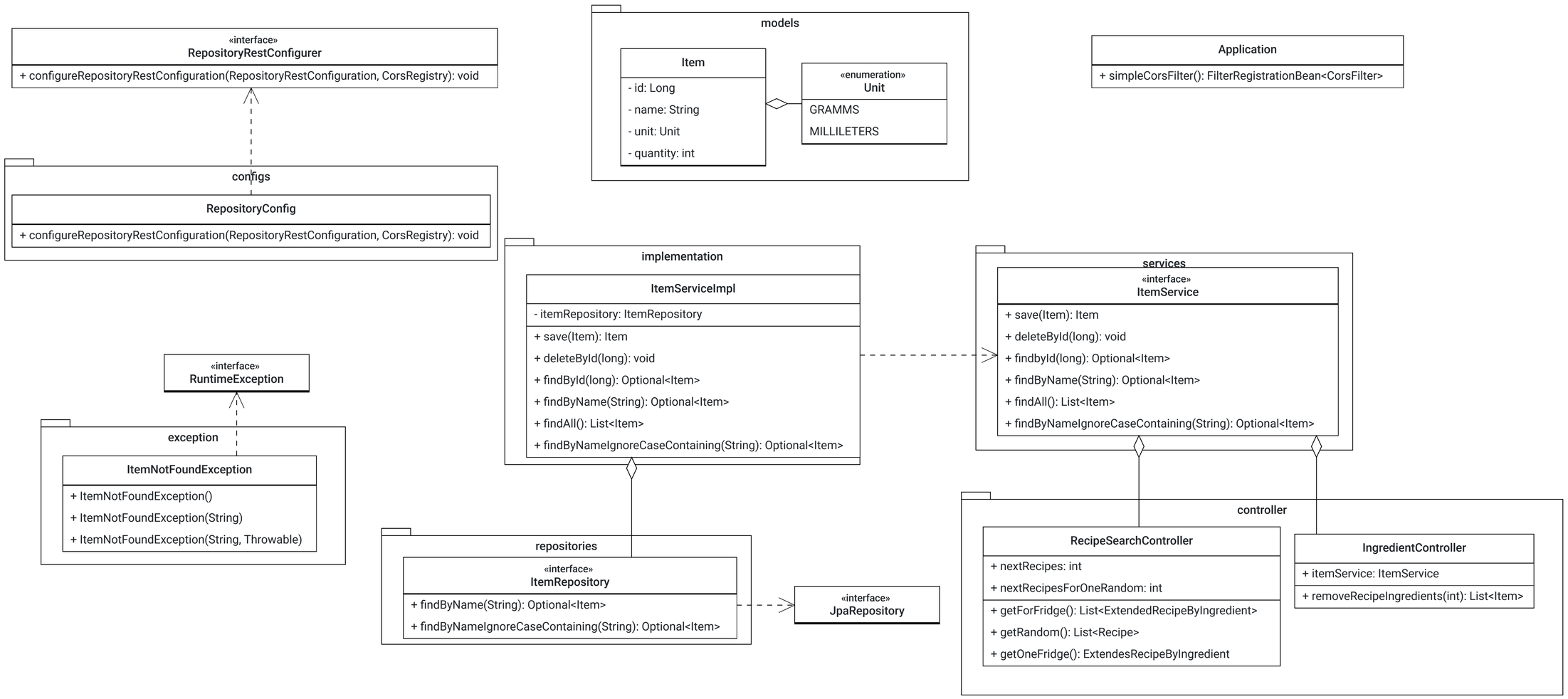
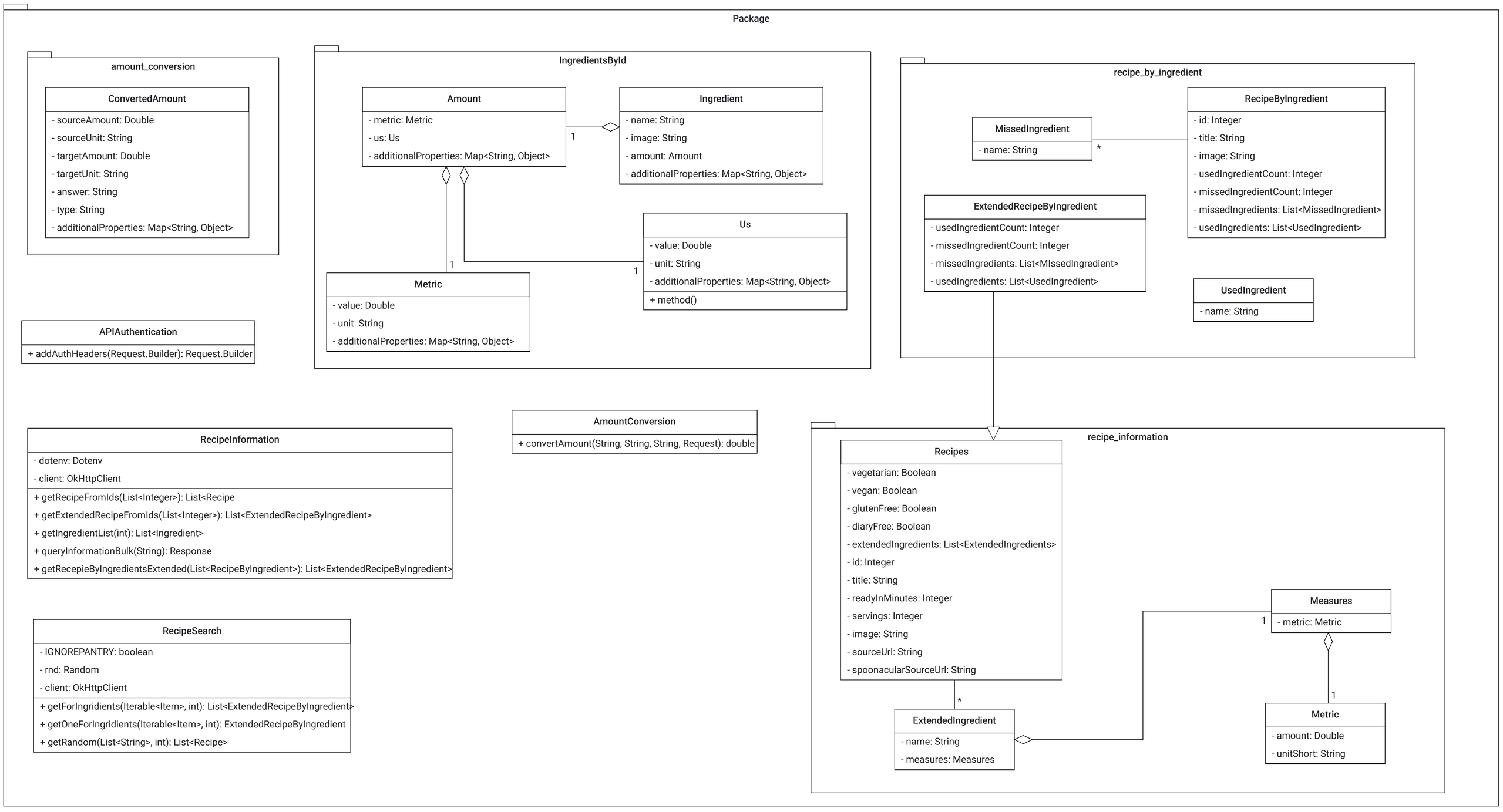

«interface»
RepositoryRestConfigurer
+ configureRepositoryRestConfiguration(RepositoryRestConfiguration, CorsRegistry): void
models
Item
- id: Long
- name: String
- unit: Unit
- quantity: int
«enumeration»
Unit
GRAMMS
MILLILETERS
Application
+ simpleCorsFilter(): FilterRegistrationBean<CorsFilter>
configs
RepositoryConfig
+ configureRepositoryRestConfiguration(RepositoryRestConfiguration, CorsRegistry): void
implementation
ItemServiceImpl
- itemRepository: ItemRepository
+ save(Item): Item
+ deleteById(long): void
+ findById(long): Optional<Item>
+ findByName(String): Optional<Item>
+ findAll(): List<Item>
+ findByNameIgnoreCaseContaining(String): Optional<Item>
services
«interface»
ItemService
+ save(Item): Item
+ deleteById(long): void
+ findbyId(long): Optional<Item>
+ findByName(String): Optional<Item>
+ findAll(): List<Item>
+ findByNameIgnoreCaseContaining(String): Optional<Item>
«interface»
RuntimeException
exception
ItemNotFoundException
+ ItemNotFoundException()
+ ItemNotFoundException(String)
+ ItemNotFoundException(String, Throwable)
repositories
«interface»
ItemRepository
+ findByName(String): Optional<Item>
+ findByNameIgnoreCaseContaining(String): Optional<Item>
«interface»
JpaRepository
controller
RecipeSearchController
+ nextRecipes: int
+ nextRecipesForOneRandom: int
+ getForFridge(): List<ExtendedRecipeByIngredient>
+ getRandom(): List<Recipe>
+ getOneFridge(): ExtendesRecipeByIngredient
IngredientController
+ itemService: ItemService
+ removeRecipeIngredients(int): List<Item>


Package
amount_conversion
ConvertedAmount
- sourceAmount: Double
- sourceUnit: String
- targetAmount: Double
- targetUnit: String
- answer: String
- type: String
- additionalProperties: Map<String, Object>
IngredientsById
Amount
- metric: Metric
- us: Us
- additionalProperties: Map<String, Object>
1
Ingredient
- name: String
- image: String
- amount: Amount
- additionalProperties: Map<String, Object>
Us
- value: Double
- unit: String
- additionalProperties: Map<String, Object>
+ method()
1
1
Metric
- value: Double
- unit: String
- additionalProperties: Map<String, Object>
recipe_by_ingredient
MissedIngredient
- name: String
*
RecipeByIngredient
- id: Integer
- title: String
- image: String
- usedIngredientCount: Integer
- missedIngredientCount: Integer
- missedIngredients: List<MissedIngredient>
- usedIngredients: List<UsedIngredient>
ExtendedRecipeByIngredient
- usedIngredientCount: Integer
- missedIngredientCount: Integer
- missedIngredients: List<MIssedIngredient>
- usedIngredients: List<UsedIngredient>
UsedIngredient
- name: String
APIAuthentication
+ addAuthHeaders(Request.Builder): Request.Builder
AmountConversion
+ convertAmount(String, String, String, Request): double
RecipeInformation
- dotenv: Dotenv
- client: OkHttpClient
+ getRecipeFromIds(List<Integer>): List<Recipe
+ getExtendedRecipeFromIds(List<Integer>): List<ExtendedRecipeByIngredient>
+ getIngredientList(int): List<Ingredient>
+ queryInformationBulk(String): Response
+ getRecepieByIngredientsExtended(List<RecipeByIngredient>): List<ExtendedRecipeByIngredient>
RecipeSearch
- IGNOREPANTRY: boolean
- rnd: Random
- client: OkHttpClient
+ getForIngridients(Iterable<Item>, int): List<ExtendedRecipeByIngredient>
+ getOneForIngridients(Iterable<Item>, int): ExtendedRecipeByIngredient
+ getRandom(List<String>, int): List<Recipe>
recipe_information
Recipes
- vegetarian: Boolean
- vegan: Boolean
- glutenFree: Boolean
- diaryFree: Boolean
- extendedIngredients: List<ExtendedIngredients>
- id: Integer
- title: String
- readyInMinutes: Integer
- servings: Integer
- image: String
- sourceUrl: String
- spoonacularSourceUrl: String
*
ExtendedIngredient
- name: String
- measures: Measures
Measures
- metric: Metric
1
1
Metric
- amount: Double
- unitShort: String