<<Interface>>
MealerSerializable
+toMap() : Map<String, Object>
+getId() : String
+setId(String id) : void

<<@Interface>>
MealerSerializableElement
+key() : String

MealerOrder
id : String
chefId : String
clientId : String
recipeId : String
+getStatusStep() : int

MealerComplaint
id : String
chefId: String
userId: String
description : String

MealerUser
id : String
firstName : String
lastName : String
email : String
address : String
creditCard : String
profilePictureUrl : String
voidCheckUrl : String
description : String
menuId : String
ratings : Map<String, List<String>>
totalSales : Integer
isSuspended : Boolean
suspendedUntil : String
availableViews: Map<MealerRole, List<Integer>>
orderIds : List<String>
messageToken : String
+MealerUser(String id, MealerRole role, String firstName, String lastName, String email, String address, String creditCard, String description, String voidCheckUrl)
+isSuspended(DatabaseClient client) : boolean
+suspend(Date suspensionEndDate) : void
+getAvailableViews() : List<Integer>
+rate(int r, String userId) : void
+averageRating() : float

MealerMenu
id : String
chefId : String
recipeIds : List<String>
+addRecipe(MealerRecipe r) : void

<<Enumeration>>
MealerOrderStatus
WAITING
ACCEPTED
COMPLETED
REJECTED

MealerOrderStatusUtils
-hiddenOrderStatus : List<String>
+toStep(MealerOrderStatus s) : int
+getMaxStatusStep() : int
+getPrettyStatus(MealerOrderStatus s, Context c) : String
+getHiddenOrderStatus() : List<String>

<<Enumeration>>
MealerRole
USER
CHEF
ADMIN

MealerClient

MealerAdmin

MealerChef

RatingUtils
+averageRating(Map<String, List<String> r) : float
+rate(Map<String, List<String> r, int r, String userId) : void

StringUtils
+capitalize(String input) : String

DateUtils
-format : DateFormat
+toDate(String isoString) : Date
+toString(Date date) : String
+toPrettyString(String isoString) : String
+nowPrettyString() : String
+isPassed(String isoString) : boolean

FieldValidator
-emailRegex : String
-context : Context
+FieldValidator(Context context)
+required(EditText text) : boolean
+required(ChipGroup c, EditText t) : boolean
+password(EditText text) : boolean
+email(EditText text) : boolean
+termsAndConditions(EditText text) : boolean
+creditCard(EditText text) : boolean
+aggregateChips(ChipGroup g) : List<String>

MealerRecipe
id : String
name : String
course : String
categories : List<String>
ingredients : List<String>
allergens : List<String>
price : Float
description : String
isOffered : Boolean
chefId : String
ratings : Map<String, List<String>>
+MealerRecipe(String id, String name, String course, List<String> categories, List<String> ingredients, List<String> allergens, float price, String description, boolean isOffered, String chefId)
+rate(int r, String userId) : void
+averageRating() : float

<<Singleton>>
Services
-database : DatabaseClient
-storage : StorageClient
+getDatabaseClient() : DatabaseClient
+getStorageClient() : StorageClient

<<Interface>>
DatabaseClient
+getUser() : Future<MealerUser>
+getUser(DatabaseCompletionCallback callback) : Future<MealerUser>
+getUser(String id) : Future<MealerUser>
+getUser(String id, DatabaseCompletionCallback callback) : Future<MealerUser>
+getUser(String id, Class<T> clazz, DatabaseCompletionCallback callback) : Future<MealerUser>
+getUserMenu(DatabaseCompletionCallback callback): Future<MealerMenu>
+getUserMenu(String id, DatabaseCompletionCallback callback): Future<MealerMenu>
+getOrder(String id, DatabaseCompletionCallback callback) : Future<MealerOrder>
+getMenu(String id, DatabaseCompletionCallback callback) : Future<MealerMenu>
+listenForUserRecipes(Activity activity, String userId, boolean offeredRecipesOnly, DatabaseCompletionCallback<List<T>> callback) : void
+getRecipe(String id, DatabaseCompletionCallback callback) : Future<MealerRecipe>
+searchRecipesByName(String name, int limit, DatabaseCompletionCallback callback) : Future<List<MealerRecipe>>
+searchRecipes(int limit, DatabaseFilterCallback filter, DatabaseCompletionCallback callback) : Future<List<MealerRecipe>>
+updateRecipe(MealerRecipe recipe, DatabaseSetCallback callback) : Future<Boolean>
+updateMenu(MealerMenu menu) : Future<Boolean>
+updateMenu(MealerMenu menu, DatabaseSetCallback callback) : Future<Boolean>
+updateUser(MealerUser user) : Future<Boolean>
+updateUser(MealerUser user, DatabaseSetCallback callback) : Future<Boolean>
+updateOrder(MealerOrder order, DatabaseSetCallback callback) : Future<Boolean>
+orderRecipe(MealerRecipe recipe, DatabaseCompletionCallback callback) : Future<MealerOrder>
+updateComplaint(MealerComplaint complaint) : Future<Boolean>
+updateComplaint(MealerComplaint complaint, DatabaseSetCallback callback) : Future<Boolean>
+deleteRecipe(String id, DatabaseSetCallback callback) : Future<Boolean>
+deleteOrder(String id, DatabaseSetCallback callback) : Future<Boolean>
+deleteComplaint(String id, DatabaseSetCallback callback) : Future<Boolean>
+updateUserToken() : Future<Boolean>
+rejectOrder(MealerOrder order, Context c, DatabaseSetCallback callback) : Future<Boolean>
+listenForModel(Activity activity, String collectionId, String documentId, Class<T> clazz, DatabaseCompletionCallback<T> callback) : DatabaseListener
+listenForModels(Activity activity, String collectionId, Class<T> clazz, DatabaseCompletionCallback<List<T>> callback) : DatabaseListener
+listenForModels(Activity activity, String collectionId, Class<T> clazz, DatabaseFilterCallback filter, DatabaseCompletionCallback<List<T>> callback) : DatabaseListen

FirebaseDatabaseClient
-getModel(String collectionId, String documentId, Class<T> clazz, DatabaseCompletionCallback<T> callback) : Future<T>)
-getModels(String collectionId,Class<T> clazz, int limit, DatabaseFilterCallback filter) : Future<List<T>>)
-saveModel(String collectionId, String documentId, Object objToSave, DatabaseSetCallback callback) : Future<Boolean>)
-deleteModel(String collectionId, String documentId, DatabaseSetCallback callback) : Future<Boolean>)

<<Interface>>
StorageClient
+uploadFile(byte[] bytes, String remotePath, StorageProgressCallback callback) : Future<Boolean>

FirebaseStorageClient
-storage : FirebaseStorage
-storageRef : StorageReference

com.google.firebase.messaging.FirebaseMessagingService

MealerMessagingService
-orderStatusType : String
+onNewToken(String token)
+showOrderNotification(String orderId)

MealerMessager
-SCOPES : String[]
-context : Context
+MealerMessager(Context context)
+notifyClientOrder(String orderId, String clientToken)
-sendMessage(String recipient, String title, String body, Map<String, String> data

android.os.AsyncTask

SendFCM
-FCM_MESSAGE_URL : String
-fcm : Map<String, Object>
+SendFCM(Map<String, Object> fcm)

<<Interface>>
DatabaseCompletionCallback<T>
+onComplete(T object) : void

<<Interface>>
DatabaseFilterCallback
+applyFilter(CollectionReference reference) : Query

<<Interface>>
DatabaseSetCallback
+onComplete(Boolean object) : void

<<Interface>>
StorageProgressCallback
+onProgress(long processed, long total) : void

DatabaseListener
listenerRegistrations : List<ListenerRegistration>
+DatabaseListener()
+DatabaseListener(ListenerRegistration r)
+remove() : void
+addRegistration(DatabaseListener l) : void