

sembast

- [sembast filter](#)

```
Future<List<Section>> loadSectionList(Pantry pantry) async {
  final finder = Finder(
    filter: Filter.equals(Column.pantryKey, pantry.key),
    sortOrders: [
      SortOrder(Column.order),
    ],
  );

  var recordSnapshots = await _store.find(await _db, finder: finder);

  return recordSnapshots.map((snapshot) {
    var section = Section.fromJson(snapshot.value);
    section = section.copyWith(key: snapshot.key);
    return section;
  }).toList();
}
```

```
Future<List<Fruit>> getAllStoredByName() async {
  final finder = Finder(sortOrders: [
    SortOrder('name'),
  ]);
  final recordSnapshots = await _fruitStore.find(
    await _db,
    finder: finder,
  );

  // Making a List<Fruit> out of List<RecordSnapshot>
  return recordSnapshots.map((snapshot) {
    final fruit = Fruit.fromMap(snapshot.value);
    // An ID is a key of a record from the database.
    fruit.id = snapshot.key;
    return fruit;
  }).toList();
}
```

- [pub.dev](#)

From:

<https://wiki.moro.kr/> - **Various Ways**

Permanent link:

<https://wiki.moro.kr/open/sembast>

Last update: **2023/06/19 06:05**

