**تقرير توثيقي للوظائف الأساسية في مشروع “المحاسبي”**

هذا التقرير يقدّم نظرة شاملة على الهيكل البرمجي لوحدات التطبيق الثلاثة الرئيسية (المعاملات الشخصية، الفئات، القوالب) في جزأين لكل وحدة: **الخدمات (Services)** و**الموفر (Provider)**. الشرح موجّه لقرّاء غير تقنيّين مع تبسيط المصطلحات، ليكون مرجعًا متكاملاً لبقية فريق العمل.

**1. وحدة المعاملات الشخصية (Transactions)**

**1.1 خدمات المعاملات (TransactionService)**

**المتغيرات الداخلية (State & Helpers)**

| **المتغيّر** | **نوعه** | **وظيفته** |
| --- | --- | --- |
| \_txnBox | Box<TransactionModel> | “صندوق” محلي (Hive) لحفظ جميع بيانات المعاملات على جهاز المستخدم. |
| \_queueBox | Box<SyncRecord> | صندوق انتظار لتغييرات المعاملات التي لم تُرسل بعد إلى الخادم (عند الاتصال). |
| \_firestore | FirebaseFirestore | الواجهة للتواصل مع قاعدة بيانات Firestore على الإنترنت. |
| \_userId | String | رقم تعريف المستخدم في نظام المصادقة (Firebase Auth). |
| \_prefs | SharedPreferences | تخزين مُفتاح lastPulledAt لمعرفة آخر وقت جلبنا فيه البيانات من الخادم. |
| \_kLastPulledKey | String | مفتاح ثابت لتخزين واسترجاع زمن آخر جلب للبيانات. |

**الدوال الأساسية**

1. **إضافة معاملة (addTransaction)**
   * تحفظ المعاملة فورًا محليًا في \_txnBox.
   * تضيف سجل انتظار (SyncRecord.create) إلى \_queueBox ليتم مزامنته لاحقًا إلى الخادم.
2. **تعديل معاملة (updateTransaction)**
   * نفس عملية الإضافة، لكن مع علامة “تحديث” في سجل الانتظار.
3. **حذف معاملة (deleteTransaction)**
   * تحذف المعاملة محليًا وتضع سجل “حذف” في \_queueBox.
4. **قراءة جميع المعاملات مرة واحدة (fetchAll)**
   * تجلب البيانات المحلية مرتبة حسب التاريخ من الأحدث للأقدم.
5. **مراقبة التغييرات في المعاملات (watchAll)**
   * تيّار معلوماتي (Stream) ينبّه الواجهة فور أي تغيير محلي في المعاملات.
6. **البحث في الوصف (searchByDescription)**
   * يفلتر نتائج fetchAll حسب نص الوصف.
7. **الوصفّات المتكررة (recentDescriptions)**
   * يخرج قائمة بأكثر الوصفّات تكرارًا لأغراض الإكمال التلقائي.
8. **فلترة حسب نطاق زمني (fetchByDateRange)**
   * يختار المعاملات بين تاريخين محدّدين.
9. **ملخّص المصروفات والإيرادات (getSummary)**
   * يحسب إجمالي المصروفات والإيرادات في فترة محدّدة.
10. **إجماليات حسب الفئة (getCategoryTotals)**
    * يحسب مجموع المبالغ لكل فئة خلال فترة معينة.

**مزامنة مع الخادم**

1. **دفع التغييرات (syncUpstream)**
   * يقرأ سجلات الانتظار في \_queueBox وينفذها على Firestore ثم يحذف السجل بعد النجاح.
2. **جلب التحديثات (syncDownstream)**
   * يقرأ زمن آخر جلب (lastPulledAt) ويستعلم Firestore عن المستندات المُعدّلة بعده، ثم يدمجها محليًا.
3. **المزامنة الكاملة (synchronize)**
   * يجمع بين syncUpstream و syncDownstream.
4. **الجلب الأوّلي (initialSeed)**
   * إذا كان الصندوق المحلي فارغًا، يجلب جميع المعاملات دفعةً واحدةً (بتمرير صفحات بحجم 100) ويرتب آخر جلب.

**أدوات مساعدة**

1. **مسح جميع المعاملات (clearAll)**
   * يفرغ الصندوقين المحليّين (\_txnBox, \_queueBox) ويعيد ضبط آخر زمن جلب.

**1.2 مزوّد المعاملات (TransactionProvider)**

**المتغيرات والحالة**

| **المتغير** | **وظيفته** |
| --- | --- |
| \_transactions | قائمة المعاملات الحالية التي تعرضها الواجهة. |
| \_isLoading | علم “جارٍ التحميل” أثناء الجلب الأوّلي أو إعادة المزامنة. |
| \_isProcessing | علم “جارٍ المعالجة” أثناء إضافة/تعديل/حذف معاملة. |
| \_isSyncing | علم “جارٍ المزامنة” أثناء استدعاء synchronize(). |
| \_error | يحتوي رسالة الخطأ الأخيرة إن وجدت لعرضها للمستخدم. |
| \_sub | اشتراك في تيّار watchAll() لتحديث \_transactions تلقائيًا. |

**دورة الحياة**

* **التهيئة (ctor → \_initialize)**
  1. ضبط \_isLoading = true.
  2. استدعاء initialSeed لجلب البيانات الأوّلية.
  3. الاشتراك في watchAll() لتلقّي التحديثات المحلية.
  4. استدعاء synchronize() لمزامنة أي تغييرات معلّقة.
  5. ضبط \_isLoading = false.
* **dispose()**
  1. إلغاء الاشتراك \_sub.cancel().

**الدوال المتاحة للواجهة**

1. **addTransaction(txn)**
2. **updateTransaction(txn)**
3. **deleteTransaction(id)**
   * جميعها تغلف العمليات بـ \_runProcessing ثم تستدعي مزامنة فورية لـ Firestore.
4. **synchronize()**
   * يدير علم \_isSyncing ويستدعي syncUpstream() ثم syncDownstream().
5. **search(query)**، **recentDescriptions(limit)**، **fetchByDateRange(start,end)**
   * استعلامات محلية على Hive مع معالجة الأخطاء.
6. **getSummary(start,end)**، **getCategoryTotals(start,end)**
   * ملخّصات وتقارير محلية.
7. **clearAll()**
   * مسح البيانات المحلية وإعادة تحميل الواجهة.

**الدوال المساعدة الداخلية**

* **\_runProcessing(fn)**
  + يضبط \_isProcessing، ينفّذ fn، يعالج الخطأ إن وجد، ثم يعيد ضبط العلم.
* **\_setLoading(bool)**، **\_setSyncing(bool)**، **\_setError(e)**
  + تحديث الحالة وإشعار الواجهة.

**2. وحدة الفئات (Categories)**

**2.1 خدمات الفئات (CategoryService)**

**المتغيرات الداخلية**

| **المتغير** | **وظيفته** |
| --- | --- |
| \_box | صندوق Hive لحفظ كل الفئات محليًا. |
| \_firestore | واجهة Firestore للتخزين والسحب من الإنترنت. |
| \_userId | رقم تعريف المستخدم الحالي. |
| \_prefs | لتخزين واسترجاع زمن آخر جلب (lastPulledAt). |
| \_kLastPulledKey | مفتاح ثابت لـ SharedPreferences. |

**الدوال الأساسية**

1. **seedDefaults()**
   * يملأ الصندوق المحلي بقائمة مُثلى من 5 فئات مضمّنة (طعام، مواصلات، شرائية، راتب، أخرى) عند التشغيل الأوّل.
2. **initialSeed()**
   * إذا كان الصندوق المحلي فارغًا، يجلب جميع الفئات من Firestore ويرتبها محليًا ثم يحفظ زمن الجلب.
3. **fetchAll()**
   * يقرأ كل الفئات محليًا ويرتبها حسب الاسم.
4. **watchAll()**
   * تيّار Hive ينبّه الواجهة فور أي تغيير.
5. **getById(id)**
   * استرجاع فئة واحدة حسب رقمها.
6. **fetchExpenses()**
   * فقط الفئات التي isExpense == true.
7. **fetchIncome()**
   * فقط الفئات التي isExpense == false.
8. **searchByName(query)**
   * فلترة حسب وجود نص في اسم الفئة.
9. **addCategory(cat)**
10. **updateCategory(cat)**
11. **deleteCategory(id)**
    * حفظ/تحديث/حذف محليًا ثم فوريًا على Firestore.
12. **syncDownstream()**
    * يجلب التغييرات من Firestore منذ آخر جلب ويحدّث الصندوق المحلي.
13. **synchronize()**
    * مجرد اسم مستعار لـ syncDownstream().
14. **clearAll()**
    * يمسح كل الفئات محليًا ويعيد ضبط زمن الجلب.

**2.2 موفّر الفئات (CategoryProvider)**

**الحالة والمتغيرات**

| **المتغير** | **وظيفته** |
| --- | --- |
| \_categories | قائمة الفئات المتاحة للواجهة. |
| \_loading | علم أثناء التهيئة والجلب الأوّلي. |
| \_processing | علم أثناء إضافة/تعديل/حذف فئة. |
| \_syncing | علم أثناء جلب التحديثات من الخادم. |
| \_error | رسالة الخطأ الأخيرة إن وجدت. |
| \_subscription | الاشتراك في تيّار watchAll() لتحديث \_categories. |

**دورة الحياة**

* **\_initialize()**
  1. \_setLoading(true)
  2. await service.seedDefaults()
  3. await service.initialSeed()
  4. الاشتراك في watchAll()
  5. \_setLoading(false)
* **dispose()** → إلغاء الاشتراك.

**الدوال العامة**

1. **addCategory(cat)**، **updateCategory(cat)**، **deleteCategory(id)**
   * جميعها تغلف \_runProcessing.
2. **getById(id)**، **fetchExpenses()**، **fetchIncome()**، **searchByName(query)**
   * تتعامل مع استعلامات محلية + معالجة الأخطاء.
3. **syncDownstream()**، **synchronize()**
   * لجلب التحديثات من الخادم يدوياً.
4. **clearAll()**
   * لمسح كل الفئات وإعادة التهيئة.

**الدوال المساعدة**

* **\_runProcessing(fn)**
* **\_setLoading(bool)**، **\_setSyncing(bool)**، **\_setError(e)**

**3. وحدة القوالب (Templates)**

**3.1 خدمات القوالب (TemplateService)**

**المتغيرات الداخلية**

| **المتغير** | **وظيفته** |
| --- | --- |
| \_box | صندوق Hive لحفظ القوالب محليًا. |
| \_firestore | واجهة Firestore للتخزين والسحب. |
| \_userId | معرّف المستخدم الحالي. |
| \_prefs | لحفظ زمن آخر جلب (lastPulledAt). |
| \_kLastPulledKey | مفتاح ثابت لتخزينه في SharedPreferences. |

**الدوال الرئيسية**

1. **initialSeed()**
   * إذا كان الصندوق فارغًا، يجلب جميع القوالب من Firestore ويرتبها حسب nextRun.
2. **fetchAll()**
   * قراءة كل القوالب محليًا وترتيبها حسب nextRun.
3. **watchAll()**
   * تيّار Hive للتحديثات الفورية.
4. **getById(id)**
   * استرجاع قالب واحد.
5. **addTemplate(tpl)**، **updateTemplate(tpl)**، **deleteTemplate(id)**
   * CRUD محلي وفوري على Firestore.
6. **fetchDueTemplates({now})**
   * يختار القوالب التي يجب تشغيلها (autoAdd == true و nextRun ≤ الآن).
7. **bumpNextRun(tpl)**
   * يحسب تاريخ التشغيل التالي بناءً على التكرار (يومي/أسبوعي/شهري/مخصص) ويحدثه محليًا وعلى الخادم.
8. **syncDownstream()**
   * يجلب تغييرات Firestore اللاحقة للـ updatedAt.
9. **synchronize()** → مرادف لـ syncDownstream().
10. **clearAll()**
    * يمسح كل القوالب ويعيد ضبط زمن الجلب.

**3.2 موفّر القوالب (TemplateProvider)**

**الحالة والمتغيرات**

| **المتغير** | **وظيفته** |
| --- | --- |
| \_templates | قائمة القوالب المحملة في الذاكرة. |
| \_isLoading | أثناء الجلب الأولي للقوالب من Firestore. |
| \_isProcessing | أثناء إضافة/تعديل/حذف قالب. |
| \_isSyncing | أثناء جلب التحديثات اللاحقة من الخادم. |
| \_error | رسالة الخطأ الأخيرة إن وجدت. |
| \_sub | اشتراك في تيّار watchAll() لتحديث \_templates فورًا. |

**دورة الحياة**

* **\_initialize()**
  1. \_setLoading(true)
  2. await service.initialSeed()
  3. اشتراك في watchAll()
  4. \_setLoading(false)
* **dispose()** → إلغاء الاشتراك.

**الدوال العامة**

1. **addTemplate(tpl)**، **updateTemplate(tpl)**، **deleteTemplate(id)**
   * تنفّذ عبر \_runProcessing.
2. **fetchDueTemplates({now})**
   * جلب ما يجب تشغيله محليًا.
3. **bumpNextRun(tpl)**
   * تحديث تاريخ التشغيل التالي.
4. **syncDownstream()**، **synchronize()**
   * جلب التحديثات من الخادم.
5. **clearAll()**
   * مسح جميع القوالب.

**الدوال المساعدة**

* **\_runProcessing(fn)**، **\_setLoading(bool)**، **\_setSyncing(bool)**، **\_setError(e)**