Business Onboarding & Smart Configuration

شرح مرفق لمبرمج **Angular** من الصفر إلى التنفيذ

Companion Brief for the Angular Developer

الغرض من الملف

هذا الملف يشرح لمبرمج Angular ما هي ميزة تهيئة النشاط الذكية، لماذا بُنيت، كيف تعمل من أول تسجيل دخول، وما المطلوب تنفيذه في الواجهة دون الحاجة لفهم تفاصيل الباك إند الداخلية.

Purpose of this brief

This file explains the Business Onboarding feature to the Angular developer from zero to implementation: what it does, how the API flow works, what the UI must build, and what must not be done on the frontend.

Prepared for: Shumoul Angular developer

Reference: docs.shumoul.com → Business Onboarding & Smart Configuration

Quick Index / فهرس مختصر

Angular الجزء الأول: الشرح العربي لمبرمج

1. ما هي ميزة تهيئة النشاط الذكية؟

هذه الميزة تظهر للمشترك بعد تفعيل حسابه وتسجيل دخوله لأول مرة. بدل أن يدخل إلى منصة شمول ويجد عشرات الإعدادات والخيارات، نسأله أسئلة بسيطة عن نشاطه، ثم يقوم النظام بتجهيز إعدادات شمول تلقائيًا بما يناسب نشاطه.

مثال: إذا كان نشاطه مطعمًا أو كوفي، سيتم تفعيل إعدادات نقاط البيع، الطاولات، المطبخ، الإضافات، السعرات، الضريبة، ومراكز التكلفة إذا طلبها.

> **Angular** المهم لمبرمج

الواجهة لا تقرر الإعدادات. الواجهة فقط تعرض الأسئلة، تحفظ الإجابات، تعرض التوصية، ثم ترسل طلب تطبيق الإعدادات. القرار الكامل موجود في API.

2. متى يظهر الاستبيان؟

الاستبيان لا يظهر داخل شاشة التسجيل الأولى. السيناريو الصحيح هو:

1. المشترك يسجل بيانات الاشتراك.
2. النظام يرسل OTP أو رابط تحقق حسب النظام الحالي.
3. بعد التفعيل، يسجل المستخدم الدخول.
4. بعد نجاح الدخول، Angular يستدعي endpoint حالة الاستبيان.
5. إذا كان الاستبيان مطلوبًا، يتم توجيه المستخدم إلى شاشة تهيئة النشاط بدل لوحة التحكم.
6. بعد إكمال التهيئة بنجاح، يدخل المستخدم إلى لوحة التحكم.

3. كيف يعمل النظام خلف الكواليس؟

تم تقسيم العمل بين مشروعين:

الجزء	المسؤولية
Shumoul.MultiTenancyApi	يملك الاستبيان، الأسئلة، القواعد، الإجابات، التوصية، وسجل التطبيق.
Shumoul.BackEnd / Shumoul.Api	يطبق الإعدادات فعليًا داخل إعدادات التينانت باستخدام IAppSettingService.
Angular	يعرض الواجهة فقط ويتعامل مع endpoints العامة الخاصة بالاستبيان.

يجب على **Angular** ألا يستدعي أبدًا **endpoint** الداخلي الخاص بالباك إند. **Angular** يتعامل فقط مع **/api/v1/onboarding/***.

4. رحلة **Angular** المطلوبة

1. بعد Login ناجح، خذ الـ token من النظام الحالي.

2. استدعِ GET /api/v1/onboarding/status.

3. إذا isRequired = true، وجّه المستخدم إلى /onboarding/setup.

4. إذا isRequired = false، وجّه المستخدم إلى لوحة التحكم.

5. اعرض الأسئلة ديناميكيًا من GET /api/v1/onboarding/survey?lang=ar.

6. بعد نهاية الأسئلة، أرسل الإجابات إلى POST /api/v1/onboarding/answers.

7. بعد نجاح الحفظ، استدعِ POST /api/v1/onboarding/recommendation.

8. اعرض شاشة مراجعة التوصية.

9. عند ضغط تطبيق الإعدادات، استدعِ POST /api/v1/onboarding/apply.

10. بعد النجاح، اعرض شاشة نجاح ثم وجّه المستخدم إلى لوحة التحكم.

5. **Endpoints** التي سيستخدمها **Angular**

Method	Endpoint	الغرض	الصلاحية
GET	/api/v1/onboarding/status	معرفة هل الاستبيان مطلوب أم لا	View
GET	/api/v1/onboarding/survey?lang=ar	جلب الأسئلة والخطوات والخيارات	View
POST	/api/v1/onboarding/answers	حفظ إجابات المستخدم	Answer - Admin فقط
POST	/api/v1/onboarding/recommendation	توليد التوصية	Answer - Admin فقط
POST	/api/v1/onboarding/apply	تطبيق الإعدادات	Apply - Admin فقط
POST	/api/v1/onboarding/skip	تخطي الاستبيان	Skip - Admin فقط

6. منطق الصلاحيات

الواجهة يجب أن تفهم أن ليس كل مستخدم داخل المنشأة يحق له تطبيق إعدادات الاستبيان. الباك إند يعتمد على JWT عادي، ويعتبر المستخدم Admin إذا كان لديه role claim بقيمة Admin.

الحالة	التصرف المتوقع في **Angular**
مستخدم مصادق داخل التينانت	يمكنه استدعاء status و survey.
مستخدم غير Admin	قد يحصل على 403 عند save/recommend/apply/skip. يجب عرض رسالة واضحة.
مستخدم Admin	يستطيع إكمال الاستبيان وتطبيق الإعدادات.
بدون Token أو Token منتهي	يتم تحويله إلى Login.

> رسالة 403 المقترحة
>
> إكمال تهيئة النشاط متاح فقط لمدير المنشأة.

7. عرض الأسئلة ديناميكيًا

لا تكتب الأسئلة داخل Angular بشكل ثابت. الأسئلة تأتي من API وفيها نوع السؤال والخيارات وقاعدة الظهور VisibleWhen.

أنواع الأسئلة المطلوبة:

- SingleChoice
- MultiChoice
- Boolean
- Number
- Text

مثال VisibleWhen: إذا كان السؤال يظهر فقط للمطاعم والكافيهات:

```
{
  "activityType": "RestaurantCafe"
```

```
}
```

ومثال سؤال يظهر فقط إذا كانت المنشأة تستخدم الضريبة:

```
{
  "usesVat": true
}
```

8. شكل الواجهة المطلوب

الواجهة يجب أن تكون عربية أولًا، RTL، واحترافية، وليست نموذجًا عاديًا مملًا. الأفضل استخدام cards للخيارات، stepper للتقدم، وألوان شمول الأزرق والأبيض.

- صفحة ترحيب: لنجهز شمول بما يناسب نشاطك
- Stepper من خمس خطوات تقريبًا
- بطاقات لاختيار نوع النشاط
- شاشة مراجعة التوصية قبل التطبيق
- حالة Loading أثناء تطبيق الإعدادات
- شاشة نجاح بعد التطبيق
- رسائل خطأ واضحة ومفهومة

9. الأخطاء التي يجب التعامل معها

الكود	المعنى	ما يعرضه **Angular**
401	المستخدم غير مصادق أو التوكن غير صالح	تحويل إلى Login
403	لا يملك صلاحية الإجراء	إظهار أن الإجراء متاح فقط لمدير المنشأة
501	جسر تطبيق الإعدادات غير مفعل	إظهار رسالة تواصل مع الدعم
500	خطأ عام في الخادم	رسالة خطأ ودية مع زر إعادة المحاولة

Network Error	انقطاع اتصال	زر إعادة المحاولة

10. أشياء ممنوعة على Angular

- لا تستدعِ /api/internal/onboarding/apply-settings-patch/.
- لا تخزن أي Internal API Key في Angular.
- لا تقرر الإعدادات من الواجهة.
- لا تكتب الأسئلة بشكل ثابت داخل الكود.
- لا تتجاوز 403 أو تعتبرها bug.
- لا تعتمد على tenant 555001 داخل الكود؛ هذا مجرد تينانت اختبار.
- لا تعدل عقود API من الواجهة.

11. قائمة قبول قبل التسليم

- بعد Login يتم فحص onboarding/status قبل فتح الداشبورد.
- إذا isRequired=true يتم فتح /onboarding/setup.
- الأسئلة تظهر ديناميكيًا من API.
- VisibleWhen يعمل.
- حفظ الإجابات يعمل للمستخدم Admin.
- غير Admin يحصل على رسالة واضحة عند 403.
- التوصية تظهر بشكل مفهوم بدون عرض JSON خام كواجهة رئيسية.
- تطبيق الإعدادات ينجح ويعرض شاشة نجاح.
- بعد Applied لا يظهر الاستبيان مرة أخرى.
- لا توجد مفاتيح أو endpoints داخلية في كود Angular.

Part Two: English Explanation for the Angular Developer

1. What is Business Onboarding?

Business Onboarding is the first-login setup experience for new Shumoul tenants. Instead of forcing the tenant to manually configure many system settings, the UI asks a short business survey, sends the answers to the API, receives a recommendation, and then applies the selected settings through the backend.

The frontend does not decide which settings to enable. The backend owns all recommendation and settings-patch logic. Angular only renders, collects answers, previews the recommendation, and calls the public onboarding endpoints.

> **Key rule**
>
> Angular must never call the internal backend apply endpoint. Use only /api/v1/onboarding/* endpoints.

2. When should it appear?

The onboarding survey should appear after account activation and after the first successful login. It should not be part of the initial registration form.

1. User registers a new subscription.
2. Account is activated through OTP or the current activation flow.
3. User logs in for the first time.
4. Angular calls GET /api/v1/onboarding/status.
5. If onboarding is required, redirect to /onboarding/setup.
6. After successful apply or skip, redirect to the dashboard.

3. Responsibility split

Layer	Responsibility
Shumoul.MultiTenancyApi	Survey definitions, questions, profiles, rules, sessions, recommendation generation, public onboarding APIs.
Shumoul.BackEnd / Shumoul.Api	Real tenant AppSettings persistence through IAppSettingService and the internal apply bridge.
Angular	UI flow, status check, dynamic survey rendering, answer submission, recommendation preview, apply request, and error handling.

4. Required Angular flow

1. After successful login, store the token using the existing auth flow.
2. Call GET /api/v1/onboarding/status.
3. If isRequired is true, route to /onboarding/setup.

4. If isRequired is false, continue to the normal dashboard.
5. Load the survey using GET /api/v1/onboarding/survey?lang=ar.
6. Render questions dynamically from the API response.
7. Save answers using POST /api/v1/onboarding/answers.
8. Generate recommendation using POST /api/v1/onboarding/recommendation.
9. Show recommendation preview.
10. Apply using POST /api/v1/onboarding/apply.
11. Show success screen and redirect to dashboard.

5. Public endpoints used by Angular

Method	Endpoint	Purpose	Permission
GET	/api/v1/onboarding/status	Check onboarding state	View
GET	/api/v1/onboarding/survey?lang=ar	Load survey definition	View
POST	/api/v1/onboarding/answers	Save answers	Answer - Admin only
POST	/api/v1/onboarding/recommendation	Generate recommendation	Answer - Admin only
POST	/api/v1/onboarding/apply	Apply recommended settings	Apply - Admin only
POST	/api/v1/onboarding/skip	Skip onboarding	Skip - Admin only

6. Authorization model

All requests must use the normal Authorization: Bearer <token> header from the existing login flow. The backend detects tenant admin users from the JWT role claim. Non-admin tenant users can read status and survey, but save/recommend/apply/skip will return 403.

Case	Expected Angular behavior
No token / invalid token	Redirect to login.
Authenticated tenant user, not Admin	Can read status/survey; show admin-required message on 403 for sensitive actions.
Tenant Admin	Can complete and apply onboarding.
Internal apply endpoint	Never call it from Angular.

7. Dynamic survey rendering

Angular must render questions based on the survey API response. Do not hard-code the questionnaire. Each question contains a type, labels, options, required flag, and optional VisibleWhen metadata.

Supported question types:

- SingleChoice
- MultiChoice
- Boolean
- Number
- Text

Example visibility rule:

```
{
  "activityType": "RestaurantCafe"
}
```

Show the question only when answers.activityType === "RestaurantCafe".

8. Recommended UI structure

- Welcome screen
- Survey stepper screen
- Question renderer component
- Recommendation preview screen
- Apply loading state
- Success screen
- Error state
- Optional skip confirmation dialog

Use Arabic-first, RTL layout, Shumoul colors, card-based options, clear progress indicator, and a professional SaaS onboarding style.

9. Error handling

HTTP	Meaning	UI behavior
401	Unauthorized or expired token	Redirect to login
403	User lacks required permission	Show admin-required message
501	Apply bridge is disabled	Show support/contact message
500	Server error	Show friendly error + retry
Network	Connection issue	Show retry option

10. Acceptance checklist

- Onboarding status is checked after login before dashboard navigation.
- Survey is rendered dynamically from the API.
- VisibleWhen logic works.
- Admin users can save answers, generate recommendation, and apply.

- Non-admin users receive a clear 403 message for sensitive actions.
- Recommendation preview is user-friendly and does not expose raw JSON as the main UI.
- Apply success redirects to dashboard.
- Applied/Skipped status no longer opens onboarding.
- Angular never calls internal endpoints and never stores internal API keys.

API Examples / أمثلة API

Status response


```
{
  "isRequired": true,
  "status": "NotStarted",
  "currentStep": 1,
  "canSkip": true,
  "completedOn": null,
  "appliedOn": null,
  "appliedProfileCode": null,
  "appliedProfileNameAr": null,
  "appliedProfileNameEn": null
}
```


Save answers request


```
{
  "answers": {
    "activityType": "RestaurantCafe",
    "branchesCount": "One",
    "usersCount": "OneToThree",
    "hasPos": true,
    "paymentMethods": ["Cash", "Card"],
    "requireCustomerBeforeClosing": "OnlyReturnsAndCoupons",
    "preventOutOfStockSales": true,
    "hasTables": true,
    "hasKitchen": true,
    "usesModifiers": true,
    "showCalories": true,
    "hasInventory": true,
    "hasExpiryProducts": true,
    "usesBatchNumber": false,
```

```
    "usesSerialNumbers": false,
    "usesScale": false,
    "usesMultiUnits": true,
    "usesVat": true,
    "pricesIncludeVat": true,
    "needsCostCenters": "Mandatory",
    "needsProjects": false,
    "hasSalesmen": false,
    "hasMarketers": false
  }
}
```

Apply request

```
{
  "sessionId": "<session id from recommendation>",
  "acceptAll": true,
  "overrides": null
}
```

Appendix: Ready Prompt for Angular Claude Code

The Angular developer can copy this prompt into Claude Code inside the Angular project. It is intentionally written in English because the codebase, APIs, routes, and DTOs use English names.

```
# Task: Phase 2 — Angular First Login Onboarding UI



We have completed and secured the backend for Business Onboarding & Smart Configuration.



Backend status:

- Onboarding backend flow is live-tested.

- /apply changes tenant AppSettings successfully.

- CostCenterSettings is supported.

- JWT authentication works.

- [MustHavePermission] is enforced.

- Onboarding View is available to authenticated tenant users.

- Onboarding Answer / Recommendation / Apply / Skip require tenant Admin role.

- Angular must use the normal Authorization: Bearer <token> header.

- Angular must never call the internal BackEnd endpoint directly.



Do not implement backend changes.

Do not bypass authorization.

Do not hard-code tenant IDs.

Do not hard-code survey questions.

Do not call /api/internal/onboarding/apply-settings-patch.

Do not expose any internal API key.

Do not implement registration/OTP/WhatsApp in this phase.



## Backend Endpoints



Use these endpoints:



GET  /api/v1/onboarding/status

GET  /api/v1/onboarding/survey?lang=ar

POST /api/v1/onboarding/answers
```

POST /api/v1/onboarding/recommendation

POST /api/v1/onboarding/apply

POST /api/v1/onboarding/skip

All requests must use the existing authenticated HTTP client/interceptor.

Main User Journey

After successful login:

1. Store token using the existing auth flow.

2. Call GET /api/v1/onboarding/status.

3. If isRequired = true, redirect user to /onboarding/setup.

4. If isRequired = false, continue to the normal dashboard.

Do not show the main dashboard before checking onboarding status.

Avoid redirect loops and never call the internal apply endpoint from Angular.

Required Routes

Add routes:

- /onboarding/setup

- /onboarding/recommendation

- /onboarding/success

Use existing auth guards.

Angular Service

Create OnboardingService with these methods:

- getStatus()

- getSurvey(lang: string)

- saveAnswers(answers: Record<string, any>)

- generateRecommendation()

- applyRecommendation(sessionId: string)

- skip()

Use the existing API base URL and HTTP interceptor pattern.

Survey Rendering

Render questions dynamically from the survey API response.

Supported question types:

- SingleChoice

- MultiChoice

- Boolean

- Number

- Text

Implement simple VisibleWhen logic:

- If null or empty, question is visible.

- If { "activityType": "RestaurantCafe" }, show only when answers.activityType === "RestaurantCafe".

- If { "usesVat": true }, show only when answers.usesVat === true.

Do not hard-code the questionnaire.

Recommendation Preview

Show:

- Profile name

- Confidence percentage

- Summary

- Settings groups included

- Warnings

- Next actions

Do not show raw JSON as the main UI.

Apply Flow

When user clicks "تطبيق الإعدادات", call POST /api/v1/onboarding/apply:

```
{
  "sessionId": "<session id>",
  "acceptAll": true,
  "overrides": null
}
```

Show loading: "جاري تطبيق الإعدادات..."

On success, show success screen and redirect to dashboard.

On failure, stay on recommendation screen and show a clear error.

Admin Permission UX

Backend restricts sensitive actions to tenant Admin.

- status/survey may work for normal tenant users.

- answers/recommendation/apply/skip may return 403 for non-admin users.

If 403 occurs, show:

"إكمال تهيئة النشاط متاح فقط لمدير المنشأة."

Do not treat 403 as a backend bug.

Error Handling

401: redirect to login.

403: show admin-required permission message.

501: show "تطبيق الإعدادات غير مفعل حاليًا. تواصل مع الدعم."

500: show friendly generic error.

Network error: show retry option.

UI Requirements

Arabic first.

RTL.

Use Shumoul colors.

Professional SaaS onboarding style.

Use card-based options, stepper progress, and responsive layout.

Suggested UI copy:

- Welcome title: "لنجهز شمول بما يناسب نشاطك"

- Subtitle: "أجب عن بعض الأسئلة البسيطة وسنقوم بضبط إعدادات النظام تلقائيًا."

- Survey title: "تهيئة نشاطك"

- Recommendation title: "الإعدادات المقترحة لمنشأتك"

- Apply button: "تطبيق الإعدادات"

- Skip button: "تخطي الآن"

- Success title: "تمت تهيئة نشاطك بنجاح"

Testing

Use tenant 555001 / مطاعم شواطئ عدن only as a dev fixture. Do not hard-code it.

Test:

1. No token -> redirect to login.

2. Valid tenant Admin token -> full flow succeeds.

3. Valid normal tenant user -> status/survey may work, sensitive actions return 403.

4. Applied status -> no onboarding redirect.

5. Skipped status -> dashboard opens normally.

6. API 501 -> show support message.

7. Network error -> show retry.

Final Report Required

At the end provide:

- Files created and modified.

- Routes added.

- Components created.

- Service methods added.

- Guards/hooks added.

- How status is checked after login.

- How dynamic survey rendering works.

- How VisibleWhen works.

- How recommendation preview works.

- How apply works.

- How 403 admin-only behavior is handled.

- Test results with tenant 555001.

- Screenshots or UI description.

- Remaining limitations.

Final note / ملاحظة نهائية

Use docs.shumoul.com as the official reference. This file is a practical companion for the Angular implementation and should be shared with the frontend developer together with the documentation link.