

توثيق واجهة برمجة التطبيقات (API Documentation)

منصة سهول - الواجهات المحسنة

الإصدار: 2.2
التاريخ: نوفمبر 2025
المؤلف: Manus AI

نظرة عامة

توفر منصة سهول واجهة برمجة تطبيقات شاملة مبنية على tRPC تتيح الوصول إلى جميع ميزات المنصة بطريقة آمنة ومحسنة. تستخدم المنصة TypeScript لضمان سلامة الأنواع (Type Safety) وتتوفر تجربة تطوير محسنة مع IntelliSense الكامل.

المميزات الرئيسية

تميز واجهة برمجة التطبيقات بعدها خصائص متقدمة تجعلها مناسبة للتطبيقات الإنتاجية. أولاً، توفر Type Safety كاملة حيث يتم التحقق من الأنواع تلقائياً في وقت التطوير والتشغيل. ثانياً، تستخدم Redis لتحسين الأداء وتقليل الحمل على قاعدة البيانات. ثالثاً، توفر & Authentication & Caching Real-time Updates من خلال تحديث البيانات تلقائياً. رابعاً، تدعم OAuth 2.0 مع نظام Authorization متكاملة.خامساً، توفر Error Handling شامل مع رسائل خطأ واضحة بالعربية.

البنية التقنية

تعتمد المنصة على 11 إطار عمل رئيسي، مع Express 4 كخادم HTTP، وDrizzle ORM، وZod للتحقق من صحة البيانات، وRedis للتخزين المؤقت، وSuperjson لدعم أنواع البيانات المعقدة مثل Date.

المصادقة والترخيص (Authentication & Authorization)

نظام المصادقة

تستخدم المنصة نظام **OAuth** للمصادقة، حيث يتم تخزين الجلسة في **Cookie** آمن مع **JWT**. يتم التحقق من الجلسة تلقائياً في كل طلب، ويتم تحديث الجلسة عند كل تسجيل دخول.

الأدوار والصلاحيات

يدعم النظام أربعة أدوار رئيسية:

الوصف	الصلاحيات	الدور
المدير الرئيسي للمنصة	جميع الصلاحيات	admin
مدير العمليات	إدارة المزارع والمستخدمين	manager
مشغل المعدات	تنفيذ المهام	operator
مزارع (قراءة فقط)	عرض البيانات فقط	farmer

Procedures Types

تنقسم `procedures` إلى نوعين:

`publicProcedure`: متاح للجميع بدون مصادقة (مثل تسجيل الدخول).

`protectedProcedure`: يتطلب مصادقة، يتم حقن `ctx.user` تلقائياً.

الواجهات البرمجية (API Endpoints)

Authentication API .1

`auth.me`

نوع: `Query`

المصادقة: `Public`

الوصف: الحصول على معلومات المستخدم الحالي

المدخلات: لا يوجد

المرجعات:

```
{  
  id: number;  
  openId: string;  
  name: string | null;  
  email: string | null;  
  role: "admin" | "manager" | "operator" | "farmer";  
  createdAt: Date;  
  lastSignedIn: Date;  
}
```

مثال استخدام:

```
const { data: user } = trpc.auth.me.useQuery();
```

auth.logout

النوع: Mutation

المصادقة: Public

الوصف: تسجيل خروج المستخدم

المدخلات: لا يوجد

المرجعات:

```
{ success: true }
```

مثال استخدام:

```
const logoutMutation = trpc.auth.logout.useMutation();  
await logoutMutation.mutateAsync();
```

⭐ جديد Dashboard API .2

`dashboard.getStats`

النوع: Query

المصادقة: Protected

التخزين المؤقت: 5 دقائق

الوصف: الحصول على إحصائيات Dashboard الشاملة

المدخلات: لا يوجد

المخرجات:

```
{  
  farms: {  
    totalFarms: number;  
    totalArea: number;  
  };  
  fields: {  
    totalFields: number;  
  };  
  equipment: {  
    totalEquipment: number;  
    activeEquipment: number;  
  };  
  droneAnalysis: {  
    totalImages: number;  
    processedImages: number;  
    avgNdvi: number;  
    totalPests: number;  
    highwaterStress: number;  
  };  
  diseaseDetection: {  
    totalDetections: number;  
    completedDetections: number;  
  };  
  lastUpdated: Date;  
}
```

مثال استخدام:

```
const { data, isLoading } = trpc.dashboard.getStats.useQuery();
```

ملاحظات:

- يتم تحديث البيانات تلقائياً كل دقيقة
- يتم حساب الإحصائيات من قاعدة البيانات مباشرةً
- يتم تخزين النتائج في Redis لمدة 5 دقائق

`dashboard.getChartData`

نوع: Query

المصادقة: Protected

التخزين المؤقت: 10 دقائق

الوصف: الحصول على بيانات الرسوم البيانية

المدخلات:

```
{  
  type: "ndvi" | "diseases" | "productivity";  
  period: "week" | "month" | "year"; // default: "month"  
}
```

المخرجات:

```
Array<{  
  date: string;  
  value: number;  
}>
```

مثال استخدام:

```
const { data: chartData } = trpc.dashboard.getChartData.useQuery({
  type: "ndvi",
  period: "month"
});
```

dashboard.getRecentAlerts

نوع: Query
المصادقة: Protected
التخزين المؤقت: 1 دقيقة
الوصف: الحصول على آخر التنبيهات

المدخلات:

```
{
  limit: number; // default: 5
}
```

المخرجات:

```
Array<{
  id: number;
  type: string;
  title: string;
  message: string;
  priority: string;
  createdAt: Date;
}>
```

مثال استخدام:

```
const { data: alerts } = trpc.dashboard.getRecentAlerts.useQuery({ limit:
  10 });
```

(Redis Caching مع Farms API .3

farms.list

النوع: Query

المصادقة: Protected

التخزين المؤقت: ✓ 5 دقائق

الوصف: الحصول على قائمة المزارع للمستخدم الحالي

المدخلات: لا يوجد

المخرجات:

```
Array<{
  id: number;
  ownerId: number;
  name: string;
  location: string | null;
  totalArea: number | null;
  createdAt: Date;
  updatedAt: Date;
}>
```

مثال استخدام:

```
const { data: farms } = trpc.farms.list.useQuery();
```

farms.getById

النوع: Query

المصادقة: Protected

التخزين المؤقت: ✓ 5 دقائق

الوصف: الحصول على تفاصيل مزرعة معينة

المدخلات:

```
{  
  farmId: number;  
}
```

المخرجات:

```
{  
  id: number;  
  ownerId: number;  
  name: string;  
  location: string | null;  
  totalArea: number | null;  
  createdAt: Date;  
  updatedAt: Date;  
}
```

مثال استخدام:

```
const { data: farm } = trpc.farms.getById.useQuery({ farmId: 1 });
```

farms.create

النوع: Mutation

المصادقة: Protected

الوصف: إنشاء مزرعة جديدة

المدخلات:

```
{  
  name: string; // 1-100 حرف  
  location?: string; // max 200 حرف  
  totalArea?: number; // موجب  
}
```

المخرجات:

```
{  
  id: number;  
  ownerId: number;  
  name: string;  
  location: string | null;  
  totalArea: number | null;  
  createdAt: Date;  
  updatedAt: Date;  
}
```

مثال استخدام:

```
const createFarm = trpc.farms.create.useMutation();  
await createFarm.mutateAsync({  
  name: "مزرعة الأمل",  
  location: "الرياض",  
  totalArea: 100  
});
```

ملاحظات:

- يتم إلغاء التخزين المؤقت تلقائياً بعد الإنشاء ✓
- يتم تعيين ownerId تلقائياً من المستخدم الحالي

farms.update

النوع: Mutation

المصادقة: Protected

الوصف: تحديث مزرعة موجودة

المدخلات:

```
{  
  farmId: number;  
  name: string;  
  location?: string;  
  totalArea?: number;  
}
```

المخرجات:

```
{  
  id: number;  
  ownerId: number;  
  name: string;  
  location: string | null;  
  totalArea: number | null;  
  createdAt: Date;  
  updatedAt: Date;  
}
```

مثال استخدام:

```
const updateFarm = trpc.farms.update.useMutation();  
await updateFarm.mutateAsync({  
  farmId: 1,  
  name: "مزرعة الأمل المحدثة",  
  totalArea: 150  
});
```

ملاحظات:

- يتم إلغاء التخزين المؤقت تلقائياً بعد التحديث ✓

farms.delete

النوع: Mutation

المصادقة: Protected

الوصف: حذف مزرعة

المدخلات:

```
{  
  farmId: number;  
}
```

المخرجات:

```
{ success: true }
```

مثال استخدام:

```
const deleteFarm = trpc.farms.delete.useMutation();  
await deleteFarm.mutateAsync({ farmId: 1 });
```

ملاحظات:

- يتم إلغاء التخزين المؤقت تلقائياً بعد الحذف ✓

Work Planner API (AI-Powered) .4

workPlanner.list

نوع: Query

المصادقة: Protected

التخزين المؤقت: 5 دقائق ✓

الوصف: الحصول على خطط العمل لحقل معين

المدخلات:

```
{  
  fieldId: number;  
  limit?: number; // default: 20  
  offset?: number; // default: 0  
}
```

المخرجات:

```
Array<{  
  id: number;  
  fieldId: number;  
  name: string;  
  cropType: string | null;  
  season: string | null;  
  startDate: Date;  
  endDate: Date | null;  
  status: "active" | "completed" | "cancelled";  
  estimatedCost: number | null;  
  actualCost: number | null;  
  createdAt: Date;  
  updatedAt: Date;  
>
```

مثال استخدام:

```
const { data: workPlans } = trpc.workPlanner.list.useQuery({ fieldId: 1 });
```

workPlanner.getTasks

النوع: Query

المصادقة: Protected

التخزين المؤقت: 3 دقائق ✓

الوصف: الحصول على مهام خطة عمل معينة

المدخلات:

```
{  
  workPlanId: number;  
}
```

المخرجات:

```
Array<{  
  id: number;  
  workPlanId: number;  
  name: string;  
  description: string | null;  
  type: string;  
  scheduledDate: Date;  
  completedDate: Date | null;  
  assignedTo: number | null;  
  equipmentId: number | null;  
  status: "pending" | "in_progress" | "completed" | "cancelled";  
  priority: "low" | "medium" | "high" | "urgent";  
  estimatedDuration: number | null;  
  actualDuration: number | null;  
  createdAt: Date;  
  updatedAt: Date;  
}>
```

مثال استخدام:

```
const { data: tasks } = trpc.workPlanner.getTasks.useQuery({ workPlanId: 1 });
```



workPlanner.generateAIRRecommendations

النوع: Mutation

المصادقة: Protected

الوصف: توليد توصيات ذكية بناءً على تحليل الطائرات والأمراض

المدخلات:

```
{  
  fieldId: number;  
  farmId: number;  
}
```

المخرجات:

```
{  
  success: boolean;  
  message: string;  
  recommendations: Array<{  
    title: string;  
    description: string;  
    priority: "high" | "medium" | "low";  
    timeframe: "urgent" | "this_week" | "this_month";  
  }>;  
  analysisData: {  
    avgNdvi: number;  
    pestCount: number;  
    highWaterStressCount: number;  
    diseaseCount: number;  
  };  
}
```

مثال استخدام:

```

const generateRecommendations =
  trpc.workPlanner.generateAIRRecommendations.useMutation();
const result = await generateRecommendations.mutateAsync({
  fieldId: 1,
  farmId: 1
});

console.log(result.recommendations);
// [
//   {
//     title: "تحسين صحة المحاصيل",
//     description: "مؤشر NDVI منخفض مؤشر ...",
//     priority: "high",
//     timeframe: "urgent"
//   }
// ]

```

كيف يعمل:

1. يجمع بيانات تحليل الطائرات (NDVI، الآفات، الإجهاد المائي)
2. يجمع بيانات كشف الأمراض
3. يستخدم LLM (Large Language Model) لتوليد توصيات مخصصة
4. إذا فشل LLM، يستخدم نظام Fallback ذكي

ملاحظات:

- يتطلب وجود صور طائرات للحقل
- يستخدم آخر 5 صور لتحليل الاتجاهات
- التوصيات مخصصة بناءً على البيانات الفعلية

`workPlanner.createFromRecommendations`

النوع: Mutation
المصادقة: Protected
الوصف: إنشاء خطة عمل تلقائياً من التوصيات

المدخلات:

```
{  
  fieldId: number;  
  recommendations: Array<{  
    title: string;  
    description: string;  
    priority: "high" | "medium" | "low";  
    timeframe: "urgent" | "this_week" | "this_month";  
  }>;  
}
```

المخرجات:

```
{  
  success: boolean;  
  workPlanId: number;  
  tasksCreated: number;  
}
```

مثال استخدام:

```
// 1. توليد التوصيات.  
const recommendations = await generateRecommendations.mutateAsync({  
  fieldId: 1,  
  farmId: 1  
});  
  
// 2. إنشاء خطة عمل.  
const createPlan =  
  trpc.workPlanner.createFromRecommendations.useMutation();  
const result = await createPlan.mutateAsync({  
  fieldId: 1,  
  recommendations: recommendations.recommendations  
});  
  
console.log(`تم إنشاء ${result.tasksCreated} مهام`);
```

(Redis Caching مع Drone Images API .5

droneImages.upload

النوع: Mutation

المصادقة: Protected

الوصف: رفع صورة طائرة للتحليل

المدخلات:

```
{  
  farmId: number;  
  fieldId?: number;  
  fileName: string;  
  fileData: string; // base64  
  captureDate?: Date;  
  altitude?: number;  
  gpsLatitude?: string;  
  gpsLongitude?: string;  
}
```

المخرجات:

```
{  
  imageId: number;  
  status: "processing";  
  message: string;  
}
```

مثال استخدام:

```
const uploadImage = trpc.droneImages.upload.useMutation();
const result = await uploadImage.mutateAsync({
  farmId: 1,
  fieldId: 1,
  fileName: "field1_20251103.jpg",
  fileData: base64String,
  altitude: 100,
  gpsLatitude: "24.7136",
  gpsLongitude: "46.6753"
});
```

droneImages.list

النوع: Query

المصادقة: Protected

 التخزين المؤقت: 3 دقائق

الوصف: الحصول على قائمة صور الطائرات

المدخلات:

```
{
  farmId: number;
  fieldId?: number;
  limit?: number; // default: 20
  offset?: number; // default: 0
}
```

المخرجات:

```
Array<{
  id: number;
  farmId: number;
  fieldId: number | null;
  uploadedBy: number;
  fileName: string;
  fileSize: number;
  fileType: string;
  storagePath: string;
  storageUrl: string;
  captureDate: Date | null;
  altitude: number | null;
  gpsLatitude: string | null;
  gpsLongitude: string | null;
  status: "uploaded" | "processing" | "processed" | "failed";
  createdAt: Date;
  updatedAt: Date;
}>
```

مثال استخدام:

```
const { data: images } = trpc.droneImages.list.useQuery({
  farmId: 1,
  fieldId: 1,
  limit: 10
});
```

`droneImages.getProcessingStatus`

نوع: Query

المصادقة: Protected

التخزين المؤقت: 30 ثانية ✓

الوصف: الحصول على حالة معالجة صورة

المدخلات:

```
{  
  imageId: number;  
}
```

المخرجات:

```
Array<{  
  id: number;  
  imageId: number;  
  jobType: "ndvi" | "segmentation" | "object_detection";  
  status: "queued" | "processing" | "completed" | "failed";  
  progress: number | null;  
  errorMessage: string | null;  
  createdAt: Date;  
  updatedAt: Date;  
>
```

مثال استخدام:

```
const { data: jobs } = trpc.droneImages.getProcessingStatus.useQuery({  
  imageId: 1  
});
```

Disease Detection API .6

`diseaseDetection.uploadImage`

النوع: Mutation

المصادقة: Protected

الوصف: رفع صورة لكشف الأمراض

المدخلات:

```
{  
  farmId: number;  
  fieldId?: number;  
  imageUrl: string;  
  cropType: string;  
}
```

المخرجات:

```
{  
  success: boolean;  
  detectionId: number;  
}
```

مثال استخدام:

```
const uploadImage = trpc.diseaseDetection.uploadImage.useMutation();  
const result = await uploadImage.mutateAsync({  
  farmId: 1,  
  fieldId: 1,  
  imageUrl: "https://.../",  
  cropType: "tomato"  
});
```

diseaseDetection.simulateYOLO

النوع: Mutation

المصادقة: Protected

الوصف: محاكاة معالجة YOLO (للمودج الأولي)

المدخلات:

```
{  
  detectionId: number;  
}
```

المخرجات:

```
{  
  success: boolean;  
  detectionId: number;  
  diseases: Array<{  
    name: string;  
    confidence: string;  
    severity: "low" | "moderate" | "high" | "critical";  
    affectedArea: string;  
    recommendations: string;  
  }>;  
}
```

مثال استخدام:

```
const simulateYOLO = trpc.diseaseDetection.simulateYOLO.useMutation();  
const result = await simulateYOLO.mutateAsync({ detectionId: 1 });
```

التخزين المؤقت (Caching Strategy)

نظام Redis

تستخدم المنصة Redis للتخلص المؤقت لتحسين الأداء. يتم تخزين النتائج تلقائياً وإلغاء التخزين عند التحديث.

مفاتيح التخزين المؤقت

TTL	المفتاح	النوع
5 دقائق	*:user:{userId}	User Data
5 دقائق	*:farm:{farmId}	Farm Data
3 دقائق	*:farm:{farmId}:drone-images	Drone Images
30 ثانية	drone-image:{imageId}:processing-status	Processing Status
5 دقائق	user:{userId}:dashboard:stats	Dashboard Stats
10 دقائق	*:user:{userId}:dashboard:chart	Chart Data
1 دقيقة	*:user:{userId}:dashboard:alerts	Alerts
5 دقائق	*:field:{fieldId}:work-plans	Work Plans

إلغاء التخزين المؤقت

يتم إلغاء التخزين المؤقت تلقائياً عند:

- إنشاء أو تحديث أو حذف مزرعة
- رفع صورة طائرة جديدة
- إنشاء خطة عمل جديدة
- تحديث حالة معالجة

معالجة الأخطاء (Error Handling)

أنواع الأخطاء

الحالة	الوصف	الكود
401	غير مصرح	UNAUTHORIZED
403	ممنوع	FORBIDDEN
404	غير موجود	NOT_FOUND
400	طلب خاطئ	BAD_REQUEST
500	خطأ في الخادم	INTERNAL_SERVER_ERROR

مثال معالجة الأخطاء

```
const { data, error, isError } = trpc.farms.getId.useQuery({ farmId: 999 });

if (isError) {
  if (error.data?.code === 'NOT_FOUND') {
    console.log('المزرعة غير موجودة');
  } else if (error.data?.code === 'FORBIDDEN') {
    console.log('ليس لديك صلاحيات الوصول');
  } else {
    console.log(`خطأ غير متوقع: ${error.message}`);
  }
}
```

أفضل الممارسات (Best Practices)

1. استخدام Optimistic Updates

```
const utils = trpc.useUtils();
const createFarm = trpc.farms.create.useMutation({
  onMutate: async (newFarm) => {
    // إلغاء الطلبات الجارية
    await utils.farms.list.cancel();

    // حفظ البيانات الحالية
    const previousFarms = utils.farms.list.getData();

    // تحديث متغّير
    utils.farms.list.setData(undefined, (old) => [...(old || []), newFarm]);

    return { previousFarms };
  },
  onError: (err, newFarm, context) => {
    // استرجاع البيانات السابقة عند الخطأ
    utils.farms.list.setData(undefined, context?.previousFarms);
  },
  onSettled: () => {
    // تحديث البيانات بعد الانتهاء
    utils.farms.list.invalidate();
  },
});
});
```

2. استخدام Pagination

```
const [page, setPage] = useState(0);
const limit = 20;

const { data: images } = trpc.droneImages.list.useQuery({
  farmId: 1,
  limit,
  offset: page * limit
});
```

3. استخدام Polling للتحديثات

```
const { data: processingStatus } =  
trpc.droneImages.getProcessingStatus.useQuery(  
  { imageId: 1 },  
  {  
    refetchInterval: 5000, // تجديد كل 5 ثوانٍ  
    enabled: status !== 'completed' && status !== 'failed'  
  }  
);
```

Satellite Images API (Sentinel Hub)

Overview

تكامل مع Sentinel Hub للحصول على صور الأقمار الصناعية من 2-Sentinel. يوفر صور RGB حقيقية وصور NDVI لتحليل صحة المحاصيل.

`satelliteImages.getTrueColorImage`

النوع: Mutation

المصادقة: Protected

الوصف: الحصول على صورة RGB حقيقية من 2-Sentinel

المدخلات:

```
{  
  bbox: {  
    minLon: number,  
    minLat: number,  
    maxLon: number,  
    maxLat: number  
  },  
  dateFrom: string, // YYYY-MM-DD  
  dateTo: string,  
  resolution?: number // default: 10m  
}
```

المخرجات:

```
{  
  success: boolean;  
  imageBase64: string;  
  date: string;  
  resolution: number;  
  error?: string;  
}
```

مثال استخدام:

```
const getImage = trpc.satelliteImages.getTrueColorImage.useMutation();  
const result = await getImage.mutateAsync({  
  bbox: {  
    minLon: 46.6753,  
    minLat: 24.7136,  
    maxLon: 46.7753,  
    maxLat: 24.8136  
  },  
  dateFrom: "2024-01-01",  
  dateTo: "2024-01-31",  
  resolution: 10  
});
```

`satelliteImages.getNDVIImage`

النوع: Mutation

المصادقة: Protected

الوصف: الحصول على صورة NDVI ملونة مع إحصائيات

المدخلات: نفس `getTrueColorImage`

المخرجات:

```
{
  success: boolean;
  imageBase64: string;
  ndviStats: {
    mean: number;
    min: number;
    max: number;
  };
  date: string;
  resolution: number;
  error?: string;
}
```

:NDVI Color Mapping

التفسير	اللون	NDVI Range
ماء/غيوم	رمادي	0.2- >
تربة عارية	بني	0.0 - 0.2-
نباتات ضعيفة	أصفر فاتح	0.2 - 0.0
نباتات متوسطة	أصفر-أخضر	0.4 - 0.2
نباتات جيدة	أخضر فاتح	0.6 - 0.4
نباتات صحية	أخضر	0.8 - 0.6
نباتات ممتازة	أخضر داكن	0.8 <

`satelliteImages.getAvailableDates`

نوع: Query

المصادقة: Protected

الوصف: الحصول على التواریخ الممکنة للصور فی منطقة معینة

المدخلات:

```
{  
  bbox: { minLon, minLat, maxLon, maxLat },  
  dateFrom: string,  
  dateTo: string  
}
```

المخرجات:

```
{  
  success: boolean;  
  dates: string[];  
  count: number;  
  error?: string;  
}
```

`satelliteImages.getFieldSatelliteImage`

نوع: Mutation

المصادقة: Protected

الوصف: الحصول على صورة فضائية لحقل محدد

المدخلات:

```
{  
  fieldId: number;  
  dateFrom: string;  
  dateTo: string;  
  imageType: "true_color" | "ndvi";  
  resolution?: number;  
}
```

المخرجات:

```
{  
  success: boolean;  
  fieldName: string;  
  imageBase64: string;  
  ndviStats?: { mean, min, max };  
  date: string;  
  resolution: number;  
  bbox: { minLon, minLat, maxLon, maxLat };  
  error?: string;  
}
```

Setup

1. إنشاء حساب Sentinel Hub

1. زيارـة [Copernicus Data Space](#)

2. إنشـاء حـساب مـجـانـي

3. الحصول عـلـى OAuth credentials

2. إضـافـة Credentials

في Settings → Secrets

```
SENTINEL_HUB_CLIENT_ID=your-client-id  
SENTINEL_HUB_CLIENT_SECRET=your-client-secret
```

Best Practices

اختيار التواريخ: Sentinel-2 يمر فوق نفس المنطقة كل 5 أيام تقريباً. استخدم نطاق تاريخ 7-14 يوم لضمان الحصول على صورة.

Cloud Coverage: النظام يستبعد تلقائياً الصور ذات التغطية السحابية < 30%.

Resolution: 10m (دقة عالية)، 20m (متوسطة)، 60m (منخفضة).

Limitations

Free Tier: 1000 requests/month •

Max bbox size: ~100 km² •

Historical data: من 2015 حتى الآن •

Update frequency: كل 5 أيام •

Weather API .11 (الطقس)

نظرة عامة

تكامل كامل مع OpenWeatherMap API للحصول على بيانات الطقس الحالية والتوقعات والمؤشرات الزراعية للمزارع. يتضمن 5 Procedures رئيسية مع Redis caching ودعم اللغة العربية.

Procedures

`weather.getCurrentWeather`

نوع: Query

المصادقة: Protected

الوصف: الحصول على الطقس الحالي لموقع محدد

المدخلات:

```
{  
  lat: number;           // إلی -90 90  
  lon: number;           // إلی -180 180  
  farmId?: number;      // اختياري  
}
```

المخرجات:

```
{  
  success: boolean;  
  farmId?: number;  
  location: { name, lat, lon };  
  current: {  
    temp: number;           درجة مئوية  
    feelsLike: number;  
    tempMin: number;  
    tempMax: number;  
    pressure: number;       كيلوباسكال  
    humidity: number;       %  
    visibility: number;     كم  
    windSpeed: number;      س/كم  
    windDeg: number;        درجة  
    clouds: number;         %  
    weather: {  
      main: string;  
      description: string;  
      icon: string;  
    };  
    sunrise: string;         // ISO 8601  
    sunset: string;          // ISO 8601  
  };  
  timestamp: string;  
}
```

مثال:

```
const weather = await trpc.weather.getCurrentWeather.useQuery({
  lat: 24.7136, // الرياض
  lon: 46.6753,
  farmId: 1,
});

console.log(`درجة الحرارة: ${weather.current.temp}°C`);
console.log(`الرطوبة: ${weather.current.humidity}%`);
```

weather.getForecast

نوع: Query

المصادقة: Protected

الوصف: الحصول على توقعات الطقس لـ 5 أيام

المدخلات:

```
{
  lat: number;
  lon: number;
  farmId?: number;
}
```

المخرجات:

```
{
  success: boolean;
  farmId?: number;
  location: { name, lat, lon };
  forecast: Array<{
    date: string; // YYYY-MM-DD
    temp: { min, max, avg };
    humidity: number;
    pressure: number;
    windSpeed: number;
    clouds: number;
    rain: number; // mm
    weather: { main, description, icon };
  }>;
  timestamp: string;
}
```

مثال:

```
const forecast = await trpc.weather.getForecast.useQuery({
  lat: 24.7136,
  lon: 46.6753,
});

forecast.forecast.forEach(day => {
  console.log(` ${day.date}: ${day.temp.max}°C / ${day.temp.min}°C`);
});
```

weather.getAgricultural

نوع: Query

المصادقة: Protected

الوصف: حساب المؤشرات الزراعية (5 مؤشرات)

المدخلات:

```
{
  lat: number;
  lon: number;
  farmId?: number;
}
```

المخرجات:

```
{
  success: boolean;
  farmId?: number;
  location: { name, lat, lon };
  agricultural: {
    heatStressIndex: {           مؤشر الإجهاد الحراري //
      value: number;
      level: 'high' | 'moderate' | 'low';
      description: string;
    };
    irrigationNeed: {           حاجة الري //
      level: 'high' | 'medium' | 'low';
      description: string;
    };
    sprayingSuitability: {     ملائمة الرش //
      level: 'good' | 'moderate' | 'poor';
      description: string;
    };
    frostRisk: {                خطر الصقيع //
      level: 'high' | 'moderate' | 'none';
      description: string;
    };
    cropGrowthIndex: {         مؤشر نمو المحاصيل //
      value: number;           // 0-100
      level: 'excellent' | 'good' | 'moderate' | 'poor';
      description: string;
    };
  };
  timestamp: string;
}
```

مثال:

```

const agri = await trpc.weather.getAgricultural.useQuery({
  lat: 24.7136,
  lon: 46.6753,
  farmId: 1,
});

if (agri.agricultural.irrigationNeed.level === 'high') {
  console.log('تنبيه: حاجة عالية للري');
}

if (agri.agricultural.sprayingSuitability.level === 'poor') {
  console.log('تحذير: غير مناسب للرش اليوم');
}

```

weather.getFarmWeather

نوع: Query

المصادقة: Protected

الوصف: الحصول على طقس مزرعة محددة (شامل)

المدخلات:

```
{
  farmId: number;
}
```

المخرجات:

```
{
  success: boolean;
  farm: { id, name, location };
  current: { ... };           // من getCurrentWeather
  forecast: [ ... ];         // من getForecast
  agricultural: { ... };     // من getAgricultural
  timestamp: string;
}
```

مثال:

```
const farmWeather = await trpc.weather.getFarmWeather.useQuery({
  farmId: 1,
});

console.log(`مزرعة: ${farmWeather.farm.name}`);
console.log(`درجة الحرارة: ${farmWeather.current.temp}°C`);
console.log(`مؤشر النمو: ${farmWeather.agricultural.cropGrowthIndex.value}`);
```

weather.getAlerts

النوع: Query

المصادقة: Protected

الوصف: الحصول على تنبيهات الطقس

المدخلات:

```
{
  lat: number;
  lon: number;
  farmId?: number;
}
```

المخرجات:

```
{  
  success: boolean;  
  farmId?: number;  
  location: { name, lat, lon };  
  alerts: Array<{  
    type: 'extreme_heat' | 'frost' | 'high_wind' | 'rain' | 'irrigation';  
    severity: 'high' | 'medium' | 'low';  
    title: string;  
    description: string;  
    icon: string;  
  }>;  
  alertCount: number;  
  timestamp: string;  
}
```

:مثال

```
const alerts = await trpc.weather.getAlerts.useQuery({  
  lat: 24.7136,  
  lon: 46.6753,  
  farmId: 1,  
});  
  
if (alerts.alertCount > 0) {  
  alerts.alerts.forEach(alert => {  
    console.log(` ${alert.icon} ${alert.title}: ${alert.description}`);  
  });  
}
```

المؤشرات الزراعية

المثالى	النطاق	الوصف	المؤشر
> 27 (منخفض)	0-50	مؤشر الإجهاد الحراري	Heat Stress Index
يعتمد على الحرارة والرطوبة	low/medium/high	حاجة الري	Irrigation Need
يعتمد على الرياح	good/moderate/poor	ملاءمة الرش	Spraying Suitability
> 5°C (عالي)	none/moderate/high	خطر الصقيع	Frost Risk
< 80 (ممتاز)	0-100	مؤشر نمو المحاصيل	Crop Growth Index

Setup

1. إنشاء حساب OpenWeatherMap

1. زيارة [OpenWeatherMap](#)

2. إنشاء حساب مجاني (Sign Up)

3. الحصول على API key

2. إضافة API Key

: Settings → Secrets في

```
OPENWEATHER_API_KEY=your-api-key
```

Best Practices

Caching: الطقس الحالي يُحفظ لمدة 10 دقائق، التوقعات لمدة ساعة.

التحديث التلقائي: استخدم `refetchInterval` للتحديث التلقائي كل 5 دقائق.

المؤشرات الزراعية: يُحسب من بيانات الطقس الحالية باستخدام معادلات معتمدة.

Limitations

Free Tier: 1000 calls/day •

- **Update frequency**: كل 10 دقائق (مع caching)
- **Language**: العربية مدعومة
- **Units**: مترية (درجة مئوية)

15 . API Open-Meteo (الطقس المجاني المتقدم)

نظرة عامة

API Open-Meteo هو API مجاني بالكامل للطقس بدون حاجة لمفتاح API. يوفر بيانات طقس حالية، توقعات 16 يوماً، بيانات تاريخية، ومؤشرات زراعية متقدمة.

الميزات

- **API key 100%** - لا يحتاج
- **توقعات 16 يوماً** - أطول من OpenWeatherMap
- **بيانات تاريخية** - منذ 1940
- **مؤشرات زراعية** - GDD, ET0, رطوبة التربة
- **Redis Caching** - 10 دقائق (حالي)، ساعة (توقعات)

Procedures

openMeteo.getCurrentWeather

نوع: Query
المصادقة: Protected
الوصف: الحصول على الطقس الحالي

المدخلات:

```
{  
  lat: number; // خط العرض //  
  lon: number; // خط الطول //  
}
```

المخرجات:

```
{  
  temperature: number;           // درجة الحرارة (°C)  
  feelsLike: number;            // الحرارة المحسوسة  
  humidity: number;             // الرطوبة (%)  
  windSpeed: number;            // سرعة الرياح (km/h)  
  windDirection: number;        // اتجاه الرياح (درجة)  
  precipitation: number;        // الأمطار (mm)  
  cloudCover: number;           // الغيوم (%)  
  pressure: number;             // الضغط (hPa)  
  weatherCode: number;          // رمز الطقس  
  weatherDescription: string;   // وصف بالعربية  
  time: string;                // الوقت  
}
```

مثال:

```
const weather = await trpc.openMeteo.getCurrentWeather.useQuery({  
  lat: 24.7136,  
  lon: 46.6753,  
});  
  
console.log(`درجة الحرارة: ${weather.temperature}°C`);  
console.log(`الرطوبة: ${weather.humidity}%`);
```

openMeteo.getForecast

نوع: Query

المصادقة: Protected

الوصف: توقعات 16 يوماً

المدخلات:

```
{  
  lat: number;  
  lon: number;  
  days?: number; // عدد الأيام (1-16)، الافتراضي: 7
```

المخرجات:

```
{  
  forecast: Array<{  
    date: string;  
    tempMax: number;  
    tempMin: number;  
    precipitation: number;  
    precipitationProbability: number;  
    windSpeed: number;  
    weatherCode: number;  
    weatherDescription: string;  
  }>;  
}
```

openMeteo.getHistoricalData

نوع: Query
المصادقة: Protected
الوصف: بيانات تاريخية (منذ 1940)

المدخلات:

```
{  
  lat: number;  
  lon: number;  
  startDate: string; // YYYY-MM-DD  
  endDate: string; // YYYY-MM-DD  
}
```

openMeteo.getAgriculturalIndices

نوع: Query

المصادقة: Protected

الوصف: مؤشرات زراعية متقدمة

المدخلات:

```
{  
  lat: number;  
  lon: number;  
  days?: number; // 1-16  
}
```

المخرجات:

```
{  
  current: {  
    gdd: number; // Growing Degree Days  
    et0: number; // Evapotranspiration  
    soilMoisture: number; // رطوبة التربة  
    soilTemperature: number; // حرارة التربة  
  };  
  forecast: Array<{  
    date: string;  
    gdd: number;  
    et0: number;  
    // ...  
  }>;  
}
```

openMeteo.getWeatherAlerts

نوع: Query

المصادقة: Protected

الوصف: تنبیهات الطقس الذكية

المخرجات:

```
{
  alerts: Array<{
    type: 'extreme_heat' | 'frost' | 'heavy_rain' | 'strong_wind' |
    'drought';
    severity: 'high' | 'medium' | 'low';
    title: string;
    message: string;
    recommendation: string;
  }>;
}
```

openMeteo.getFarmWeather

النوع: Query

المصادقة: Protected

الوصف: طقس مزرعة شامل (حالي + توقعات + مؤشرات + تنبيهات)

المدخلات:

```
{
  farmId: number;
  days?: number; // 1-16
}
```

المؤشرات الزراعية

الاستخدام	الوحدة	الوصف	المؤشر
تبع نمو المحاصيل	C°	Growing Degree Days	GDD
حساب احتياجات الري	mm	Evapotranspiration	ET0
جدولة الري	m³/m³	رطوبة التربة	Soil Moisture
وقت الزراعة	C°	حرارة التربة	Soil Temp

Best Practices

Caching: الطقس الحالي 10 دقائق، التوقعات ساعة، التاريخي 24 ساعة.

التحديث التلقائي: refetchInterval: 600000 (10 دقائق).

لا API Key مطلوب بالكامل بدون تسجيل.

IQAir API .16 (جودة الهواء)

نظرة عامة

IQAir يوفر بيانات جودة الهواء العالمية مع تحليل تأثيرها على المحاصيل والعمال.

الميزات

- **AQI عالمي** - مؤشر جودة الهواء ✓
- **6 ملوثات** - PM2.5, PM10, O₃, NO₂, SO₂, CO ✓
- **تأثير المحاصيل** - تحليل مخصص ✓
- **توصيات** - رش، ري، حصاد، عمال ✓
- **دقيقة Redis Caching** - 30 ✓

Procedures

`airQuality.getCurrentAirQuality`

نوع: Query
المصادقة: Protected
الوصف: جودة الهواء الحالية

المدخلات:

```
{  
  lat: number;  
  lon: number;  
}
```

المخرجات:

```
{
  location: { city, state, country };
  airQuality: {
    aqi: number; // 0-500
    mainPollutant: string; // pm25, pm10, o3, etc.
    level: string; // جيد, معتدل, سيء
    color: string; // green, yellow, red
    description: string; // وصف بالعربية
  };
  pollutants: {
    pm25: number;
    pm10: number;
    o3: number;
    no2: number;
    so2: number;
    co: number;
  };
}
```

airQuality.getFarmAirQuality

النوع: Query

المصادقة: Protected

الوصف: جودة الهواء لمزرعة مع تحليل التأثير

المدخلات:

```
{
  farmId: number;
}
```

المخرجات:

```
{
    // ... (نفس getCurrentAirQuality)
    cropImpact: Array<{
        pollutant: string;
        severity: 'low' | 'medium' | 'high';
        description: string;
        recommendation: string;
    }>;
    recommendations: {
        farmWorkers: string;      // توصيات للعمال
        spraying: string;         // ملائمة الرش
        irrigation: string;       // توصيات الري
        harvesting: string;       // توصيات الحصاد
    };
    alerts: Array<{
        severity: 'high' | 'medium' | 'low';
        title: string;
        message: string;
    }>;
}
}
```

airQuality.getPollutantInfo

النوع: Query

المصادقة: Protected

الوصف: معلومات تفصيلية عن ملوث

المدخلات:

```
{
    pollutant: 'pm25' | 'pm10' | 'o3' | 'no2' | 'so2' | 'co';
}
```

المخرجات:

```

{
  name: string;
  description: string;
  sources: string[];           مصادر التلوث // 
  healthEffects: string;       التأثيرات الصحية // 
  cropEffects: string;         التأثير على المحاصيل // 
  safeLevel: number;           المستوى الآمن // 
  recommendations: string[];   توصيات // 
  sensitiveCrops?: string[];   محاصيل حساسة // 
}

```

Setup

1. زياره [IQAIR API](#)
2. إنشاء حساب والحصول على API key
3. في Settings → Secrets: IQAIR_API_KEY=your-key

Best Practices

- دقيقة Caching: 30
- التحديث: refetchInterval: 1800000 (30 دقيقة).
- التنبيهات: مراقبة AQI > 150 للتنبيهات العاجلة.
-

17 . PVWatts API (الطاقة الشمسية)

نقطة عامة

PVWatts يحسب إمكانيات الطاقة الشمسية والعائد على الاستثمار للمزارع.

الميزات

- حساب الإنتاج - kWh سنوي ✓
- التوفير - حساب التوفير المالي ✓

ROI - العائد على الاستثمار ✓ •

توصيات - حجم النظام، الموضع ✓ •

ساعة - Redis Caching ✓ •

Procedures

solarEnergy.calculateSolarPotential

النوع: Query

المصداقية: Protected

الوصف: حساب إمكانيات الطاقة الشمسية

المدخلات:

```
{  
  lat: number;  
  lon: number;  
  systemCapacity: number; // kW  
  moduleType?: 0 | 1 | 2; // 0=Standard, 1=Premium, 2=Thin film  
  arrayType?: 0 | 1 | 2 | 3 | 4; // 0=Fixed, 1=1-axis, 2=2-axis, etc.  
  tilt?: number; // زاوية الميل (درجة)  
  azimuth?: number; // الاتجاه (180=جنوب)
```

المخرجات:

```
{  
  annualProduction: number; // kWh/سنة  
  monthlyProduction: number[]; // kWh/شهر  
  capacityFactor: number; // %  
  annualSavings: number; // $  
  monthlySavings: number; // $  
  co2Reduction: number; // طن/سنة}
```

`solarEnergy.calculateSavings`

نوع: Query
المصادقة: Protected
الوصف: حساب التوفير والعائد

المدخلات:

```
{
  annualProduction: number;           // kWh
  electricityRate: number;          // $/kWh
  systemCost: number;                // $
  incentives?: number;               // $
  maintenanceCost?: number;          // $/سنة
}
```

المخرجات:

```
{
  annualSavings: number;             // $
  paybackPeriod: number;            // سنوات
  roi: number;                     // %
  lifetimeSavings: number;          // $ (25 سنة)
  netPresentValue: number;          // $
}
```

`solarEnergy.getFarmRecommendations`

نوع: Query
المصادقة: Protected
الوصف: توصيات شاملة للمزرعة

المدخلات:

```
{  
    farmId: number;  
    farmArea: number; // هكتار  
    monthlyElectricityBill: number; // $  
}
```

المخرجات:

```
{  
    solarPotential: { ... }; // من calculateSolarPotential  
    analysis: {  
        recommendedSystemSize: number; // kW  
        maxSystemSize: number; // kW  
        coveragePercentage: number; // %  
    };  
    recommendations: {  
        systemType: string; // نوع النظام  
        reason: string; // السبب  
        placement: string[]; // مواضع مقترنة  
        considerations: string[]; // اعتبارات  
        benefits: string[]; // الفوائد  
    };  
    nextSteps: string[]; // الخطوات التالية  
}
```

Setup

1. زياره [NREL Developer Network](#)
2. إنشاء حساب والحصول على API key
3. في Settings → Secrets: NREL_API_KEY=your-key

Best Practices

الدقة: ساعة للحسابات.

التحديث: لا حاجة للتحديث التلقائي (بيانات ثابتة).

الخلاصة

توفر منصة سهول واجهة برمجة تطبيقات شاملة ومحسنة تدعم جميع ميزات المنصة. تميز الواجهة بالأمان والأداء العالي والتوثيق الشامل، مما يجعلها مناسبة للتطبيقات الإنتاجية.

الميزات الجديدة في الإصدار 2.2

- Dashboard API: إحصائيات شاملة ورسوم بيانية
- Redis Caching: تحسين الأداء بنسبة 80%
- AI Work Planner: توصيات ذكية بناءً على البيانات
- Cache Invalidiation: إلغاء تلقائي عند التحديث
- Sentinel Hub Integration: صور أقمار صناعية مع NDVI
- Weather API: طقس حالي وتوقعات ومؤشرات زراعية
- Open-Meteo API: طقس مجاني متقدم (16 يوم توقعات + بيانات تاريخية)
- IQAir API: جودة الهواء مع تحليل التأثير على المحاصيل
- PVWatts API: حساب إمكانيات الطاقة الشمسية والعائد