API Documentation

Real-World AR ChatGPT for Farmers

Version: 1.0.0

Base URL: `https://api.farmnavigator.app/v1`

Authentication: Bearer Token (JWT)

Table of Contents

---

1. Authentication

1.1 Obtaining a Token

POST `/auth/token`

Request:

```json
{
  "api_key": "your-api-key",
  "api_secret": "your-api-secret"
}
```

Response:

```json
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 86400,
  "refresh_token": "refresh-token-string"
}
```

1.2 Refreshing a Token

POST `/auth/refresh`

Request:

```json
{
  "refresh_token": "refresh-token-string"
}
```

Response: Same as obtaining token

1.3 Using the Token

Include in headers:

```http
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

---

2. Endpoints Overview

Category	Endpoint	Method	Description
Insights	`/insights`	GET	Get satellite data insights
Recommendations	`/recommendations`	POST	Get crop recommendations
Chat	`/chat`	POST	Send chat message
Voice	`/voice`	POST	Process voice input
Alerts	`/alerts`	GET/POST	Manage alerts
Companion	`/companion`	GET/PUT	Crop companion status

Category	Endpoint	Method	Description
History	`/history`	GET	User activity history
Locations	`/locations`	GET/POST/PUT/DELETE	Manage field locations

---

3. Core APIs

3.1 Insights API

Get Location Insights

GET `/insights`

Query Parameters:

Parameter	Type	Required	Description
lat	float	Yes	Latitude (-90 to 90)
lon	float	Yes	Longitude (-180 to 180)
area_m2	integer	No	Area in square meters
layers	string	No	Comma-separated layer names
date	string	No	ISO date (YYYY-MM-DD)

Example Request:

```http
GET /v1/insights?lat=40.7128&lon=-95.3698&area_m2=10000&layers=moisture,ndvi
```

Example Response:

```json
```

```
{
  "status": "success",
  "data": {
    "location": {
      "latitude": 40.7128,
      "longitude": -95.3698,
      "area_m2": 10000
    },
    "insights": {
      "soil_moisture": {
        "value": 0.35,
        "unit": "volumetric_fraction",
        "status": "moderate",
        "optimal_range": {
          "min": 0.30,
          "max": 0.60
        }
      },
      "surface_temperature": {
        "value": 28.5,
        "unit": "celsius",
        "daily_range": {
          "min": 22.0,
          "max": 31.0
        }
      },
      "ndvi": {
        "value": 0.72,
        "status": "healthy",
        "change_7d": 0.05
      },
      "evapotranspiration": {
        "value": 4.2,
        "unit": "mm_per_day"
      },
      "precipitation": {
        "last_24h": 0,
        "last_7d": 12.5,
        "unit": "mm"
      }
    },
    "metadata": {
      "timestamp": "2025-01-15T14:30:00Z",
```

```
    "data_sources": {
      "smap": {
        "granule_id": "SMAP_L3_SM_P_20250115",
        "acquisition_time": "2025-01-15T06:00:00Z"
      },
      "modis": {
        "product": "MOD13Q1",
        "tile": "h10v05"
      }
    },
    "cache_status": "hit",
    "cache_ttl": 1800
  }
 }
}
```


3.2 Recommendations API

Get Crop Recommendations

POST `/recommendations`

Request Body:


```
json

{
 "location": {
  "latitude": 40.7128,
  "longitude": -95.3698
 },
 "season": "spring",
 "constraints": {
  "water_availability": "moderate",
  "organic_only": false,
  "field_size_hectares": 10
 },
 "preferences": {
  "crop_types": ["grains", "vegetables"],
  "risk_tolerance": "medium"
 }
}
```


Response:

json

```
{
  "status": "success",
  "data": {
    "recommendations": [
      {
        "rank": 1,
        "crop": {
          "name": "Corn",
          "variety": "Pioneer P9234",
          "type": "grain"
        },
        "suitability_score": 92,
        "metrics": {
          "expected_yield": {
            "value": 180,
            "unit": "bushels_per_acre"
          },
          "water_requirement": {
            "value": 22,
            "unit": "inches_per_season"
          },
          "growing_period": {
            "value": 120,
            "unit": "days"
          }
        },
        "rationale": {
          "pros": [
            "Optimal soil moisture for germination",
            "Temperature range ideal for variety",
            "Strong local market demand"
          ],
          "cons": [
            "Requires consistent irrigation",
            "Susceptible to corn borer in region"
          ]
        },
        "planting_window": {
          "optimal_start": "2025-04-15",
          "optimal_end": "2025-05-15",
          "latest_viable": "2025-05-31"
        }
      }
```

```
  ],
  "alternative_options": [
   {
    "crop": "Soybeans",
    "score": 87,
    "reason": "Lower water requirement"
   },
   {
    "crop": "Sorghum",
    "score": 79,
    "reason": "Drought tolerant option"
   }
  ]
 }
}
```


3.3 Chat API

Send Chat Message

POST `/chat`

Request Body:


```
json

{
 "message": "What fertilizer should I use for corn?",
 "context": {
  "location": {
   "latitude": 40.7128,
   "longitude": -95.3698
  },
  "crop": "corn",
  "growth_stage": "V4",
  "session_id": "sess_abc123"
 },
 "preferences": {
  "response_length": "detailed",
  "include_citations": true
 }
}
```


Response:

```json
{
  "status": "success",
  "data": {
    "response": "For corn at V4 stage in your location, I recommend a sidedress application of nitrogen. Basec
    "citations": [
      {
        "type": "research",
        "source": "University of Nebraska Extension",
        "title": "Nitrogen Management for Corn",
        "year": 2024
      },
      {
        "type": "data",
        "source": "NASA SMAP",
        "metric": "soil_moisture",
        "value": "35%"
      }
    ],
    "follow_up_suggestions": [
      "How do I calculate exact nitrogen needs?",
      "What are signs of nitrogen deficiency?",
      "When should I do a soil test?"
    ],
    "session": {
      "id": "sess_abc123",
      "message_count": 5,
      "context_retained": true
    }
  }
}
```

3.4 Voice API

Process Voice Input

POST `/voice`

Request Headers:

```http
```

```
Content-Type: multipart/form-data
```

Request Body (Form Data):

- `audio`: Audio file (webm, wav, mp3)
- `language`: Language code (default: "en-US")
- `context`: JSON string with session context

Response:

```json
{
  "status": "success",
  "data": {
    "transcription": {
      "text": "What's the weather forecast for this week?",
      "confidence": 0.96,
      "language": "en-US"
    },
    "response": {
      "text": "This week's forecast shows mostly sunny conditions with temperatures ranging from 22-28°C. Th
      "audio_url": "https://cdn.farmnavigator.app/audio/response_abc123.mp3",
      "duration": 8.5
    }
  }
}
```

3.5 Alerts API

Get Active Alerts

GET `/alerts`

Query Parameters:

Parameter	Type	Required	Description
location_id	string	No	Filter by location
severity	string	No	Filter by severity (low, medium, high, critical)
active	boolean	No	Only active alerts (default: true)

Response:

json


```
{
  "status": "success",
  "data": {
    "alerts": [
      {
        "id": "alert_789xyz",
        "type": "weather",
        "severity": "high",
        "title": "Heat Stress Warning",
        "message": "Temperatures exceeding 35°C expected tomorrow. Consider irrigation.",
        "location": {
          "id": "loc_field1",
          "name": "North Field"
        },
        "triggers": {
          "temperature": 36,
          "humidity": 25
        },
        "recommendations": [
          "Irrigate in early morning",
          "Consider shade nets for sensitive crops",
          "Monitor soil moisture closely"
        ],
        "created_at": "2025-01-15T10:00:00Z",
        "expires_at": "2025-01-16T18:00:00Z",
        "status": "active",
        "acknowledged": false
      }
    ],
    "summary": {
      "total": 3,
      "critical": 0,
      "high": 1,
      "medium": 2,
      "low": 0
    }
  }
}
```


Create Custom Alert

POST `/alerts`

Request Body:

```json
{
  "name": "Irrigation Reminder",
  "location_id": "loc_field1",
  "conditions": {
    "soil_moisture_below": 30,
    "days_since_rain": 5
  },
  "notification_preferences": {
    "push": true,
    "email": false,
    "time_of_day": "08:00"
  }
}
```

3.6 Companion API

Get Companion Status

GET `/companion/{companion_id}`

Response:

```json
```

```
{
  "status": "success",
  "data": {
    "companion": {
      "id": "comp_abc123",
      "name": "Corny",
      "crop_type": "corn",
      "avatar_state": "happy",
      "growth_stage": "V4",
      "health": 85,
      "happiness": 90,
      "planted_date": "2025-04-20",
      "days_old": 35,
      "achievements": [
        {
          "id": "first_week",
          "name": "Week One Warrior",
          "earned_at": "2025-04-27"
        }
      ],
      "stats": {
        "water_efficiency": 92,
        "growth_rate": "normal",
        "stress_events_avoided": 3
      },
      "next_milestone": {
        "stage": "V6",
        "days_remaining": 7,
        "requirements": [
          "Maintain soil moisture above 35%",
          "Apply nitrogen fertilizer"
        ]
      }
    }
  }
}
```


Update Companion Action

PUT `/companion/{companion_id}/action`

Request Body:

```json
{
  "action": "water",
  "amount": {
    "value": 25,
    "unit": "mm"
  },
  "timestamp": "2025-01-15T14:00:00Z"
}
```

3.7 History API

Get User Activity History

GET `/history`

Query Parameters:

Parameter	Type	Required	Description
start_date	string	No	ISO date
end_date	string	No	ISO date
type	string	No	Filter by activity type
limit	integer	No	Max results (default: 50)

Response:

```json
```

```
{
  "status": "success",
  "data": {
    "activities": [
      {
        "id": "act_123",
        "type": "irrigation",
        "timestamp": "2025-01-14T08:30:00Z",
        "location": "North Field",
        "details": {
          "amount": "25mm",
          "duration": "2 hours"
        },
        "outcome": "successful",
        "impact": {
          "soil_moisture_change": "+15%"
        }
      }
    ],
    "pagination": {
      "total": 245,
      "page": 1,
      "per_page": 50,
      "next_cursor": "cursor_xyz"
    }
  }
}
```


3.8 Locations API

List User Locations

GET `/locations`

Response:

```
json
```

```
{
  "status": "success",
  "data": {
    "locations": [
      {
        "id": "loc_field1",
        "name": "North Field",
        "coordinates": {
          "latitude": 40.7128,
          "longitude": -95.3698
        },
        "area_hectares": 10,
        "current_crop": "corn",
        "soil_type": "loam",
        "created_at": "2025-01-01T00:00:00Z",
        "last_accessed": "2025-01-15T14:00:00Z"
      }
    ]
  }
}
```


Add New Location

POST `/locations`

Request Body:


```
json

{
  "name": "South Field",
  "coordinates": {
    "latitude": 40.7000,
    "longitude": -95.3700
  },
  "area_hectares": 15,
  "soil_type": "clay_loam",
  "notes": "Recently acquired, needs soil testing"
}
```


---

4. Request/Response Formats

4.1 Standard Request Headers

```http
Content-Type: application/json
Authorization: Bearer <token>
X-API-Version: 1.0
X-Request-ID: <unique-id>
Accept-Language: en-US
```

4.2 Standard Response Format

```json
{
  "status": "success|error",
  "data": {},
  "error": {
    "code": "ERROR_CODE",
    "message": "Human readable message",
    "details": {}
  },
  "metadata": {
    "timestamp": "2025-01-15T14:30:00Z",
    "request_id": "req_abc123",
    "version": "1.0.0"
  }
}
```

4.3 Pagination Format

```json
```

```
{
  "data": [],
  "pagination": {
    "page": 1,
    "per_page": 20,
    "total": 100,
    "total_pages": 5,
    "next_cursor": "cursor_string",
    "has_more": true
  }
}
```

---

5. Error Handling

5.1 Error Response Format

```
json

{
  "status": "error",
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid latitude value",
    "details": {
      "field": "latitude",
      "value": 200,
      "constraint": "Must be between -90 and 90"
    },
    "request_id": "req_abc123",
    "documentation_url": "https://docs.farmnavigator.app/errors/VALIDATION_ERROR"
  }
}
```

5.2 Error Codes

Code	HTTP Status	Description
AUTH_REQUIRED	401	Authentication required
AUTH_INVALID	401	Invalid or expired token
PERMISSION_DENIED	403	Insufficient permissions
NOT_FOUND	404	Resource not found

Code	HTTP Status	Description
VALIDATION_ERROR	400	Request validation failed
RATE_LIMIT_EXCEEDED	429	Too many requests
INTERNAL_ERROR	500	Server error
SERVICE_UNAVAILABLE	503	Service temporarily unavailable
NASA_API_ERROR	502	Upstream NASA API error
QUOTA_EXCEEDED	402	API quota exceeded

5.3 Error Recovery Examples

Rate Limit Response

http

```
HTTP/1.1 429 Too Many Requests
Retry-After: 60
X-RateLimit-Limit: 60
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 1642267200
```

json

```
{
  "status": "error",
  "error": {
    "code": "RATE_LIMIT_EXCEEDED",
    "message": "Rate limit of 60 requests per minute exceeded",
    "details": {
      "limit": 60,
      "remaining": 0,
      "reset_at": "2025-01-15T14:31:00Z"
    }
  }
}
```

---

6. Rate Limiting

6.1 Rate Limit Rules

Tier	Requests/Minute	Requests/Hour	Requests/Day
Free	10	100	1,000
Basic	60	1,000	10,000
Pro	300	5,000	50,000
Enterprise	Custom	Custom	Custom

6.2 Rate Limit Headers

```http
X-RateLimit-Limit: 60
X-RateLimit-Remaining: 45
X-RateLimit-Reset: 1642267200
X-RateLimit-Tier: basic
```

6.3 Burst Limits

- Short burst: 2x limit for 10 seconds
- Cooldown period: 30 seconds after burst

---

7. Webhooks

7.1 Webhook Events

Event	Description	Payload
alert.created	New alert generated	Alert object
alert.triggered	Alert condition met	Alert + trigger data
companion.milestone	Growth milestone reached	Companion + milestone
recommendation.ready	New recommendations available	Recommendation summary
data.updated	New satellite data available	Location + data type

7.2 Webhook Configuration

POST `/webhooks`

Request:

```json
{
  "url": "https://your-domain.com/webhook",
  "events": ["alert.created", "alert.triggered"],
  "secret": "webhook_secret_key",
  "active": true
}
```

7.3 Webhook Payload Example

```json
{
  "event": "alert.created",
  "timestamp": "2025-01-15T14:30:00Z",
  "data": {
    "alert": {
      "id": "alert_123",
      "type": "weather",
      "severity": "high"
    }
  },
  "signature": "sha256=abcd1234..."
}
```

---

8. SDKs and Libraries

8.1 Official SDKs

JavaScript/Node.js

```javascript
npm install @farmnavigator/sdk
```

```javascript
```

```javascript
const FarmNavigator = require('@farmnavigator/sdk');

const client = new FarmNavigator({
  apiKey: 'your-api-key',
  apiSecret: 'your-api-secret'
});

// Get insights
const insights = await client.insights.get({
  latitude: 40.7128,
  longitude: -95.3698
});

// Send chat message
const response = await client.chat.send({
  message: 'What should I plant?',
  context: { location: { latitude: 40.7128, longitude: -95.3698 } }
});
```

Python

```python
pip install farmnavigator
```

```python
```

```python
from farmnavigator import Client

client = Client(
    api_key='your-api-key',
    api_secret='your-api-secret'
)

# Get insights
insights = client.insights.get(
    latitude=40.7128,
    longitude=-95.3698
)

# Get recommendations
recommendations = client.recommendations.create(
    location={'latitude': 40.7128, 'longitude': -95.3698},
    season='spring'
)
```

8.2 Code Examples

Batch Location Processing

```javascript
const locations = [
  { lat: 40.7128, lon: -95.3698, name: 'Field 1' },
  { lat: 40.7130, lon: -95.3700, name: 'Field 2' }
];

const insights = await Promise.all(
  locations.map(loc =>
    client.insights.get({
      latitude: loc.lat,
      longitude: loc.lon
    })
  )
);
```

Streaming Alerts

```javascript
```

```javascript
const eventSource = new EventSource(
  'https://api.farmnavigator.app/v1/alerts/stream',
  {
    headers: {
      'Authorization': `Bearer ${token}`
    }
  }
);

eventSource.addEventListener('alert', (event) => {
  const alert = JSON.parse(event.data);
  console.log('New alert:', alert);
});
```

---

9. Testing

9.1 Test Environment

Base URL: `https://api-test.farmnavigator.app/v1`

Test Credentials:

```json
{
  "api_key": "test_key_abc123",
  "api_secret": "test_secret_xyz789"
}
```

9.2 Test Locations

Name	Latitude	Longitude	Description
Test Farm 1	40.7128	-95.3698	Normal conditions
Test Farm 2	35.6762	139.6503	High moisture
Test Farm 3	-33.8688	151.2093	Drought conditions

9.3 Postman Collection

Download: FarmNavigator API Collection

---

10. Changelog

Version 1.0.0 (2025-01-15)

- Initial API release
- Core endpoints: insights, recommendations, chat
- Basic authentication with JWT
- Rate limiting implementation

Upcoming (v1.1.0)

- GraphQL support
- Batch operations
- Advanced analytics endpoints
- WebSocket support for real-time updates

---

11. Support

Contact Information

- Email: api-support@farmnavigator.app
- Developer Forum: https://forum.farmnavigator.app
- Status Page: https://status.farmnavigator.app
- Documentation: https://docs.farmnavigator.app

SLA

- Uptime: 99.5%
- Response time: <2 seconds (p95)
- Support response: <24 hours (business days)