# MTN Coverage API Endpoints Documentation

This document outlines all the MTN API endpoints used in the MTN Coverage Checker application for determining network coverage across South Africa.

## Overview

The MTN Coverage Checker integrates with multiple MTN API services to provide accurate coverage information. The application uses a multi-tiered approach:

1. **Primary**: MTN GeoServer WMS (Web Map Service) - Real-time coverage data
2. **Fallback**: Infrastructure-based analysis - Intelligent coverage prediction
3. **Legacy**: Alternative API endpoints for additional validation

This tiered approach ensures reliable coverage information even when CORS restrictions prevent direct API access from localhost environments.

## Primary API Endpoints

### 1. MTN GeoServer WMS Service

**Base URL:** https://mtnsi.mtn.co.za/cache/geoserver/wms

This is the primary endpoint for real coverage data. It provides WMS services for querying coverage layers by technology type.

#### Technology Layer Configuration

Each technology uses specific layer identifiers and styles:

|  |  |  |  |
| --- | --- | --- | --- |
| Technology | MLID | Layer | Style |
| 2G | EBU-RBUS-ALL | mtnsi:MTNSA-Coverage-2G | MTN-Coverage-2G |
| 3G | EBU-RBUS-ALL | mtnsi:MTNSA-Coverage-3G | MTN-Coverage-3G |
| 4G | EBU-RBUS-ALL | mtnsi:MTNSA-Coverage-4G | MTN-Coverage-4G |
| 5G | EBU-RBUS-ALL | mtnsi:MTNSA-Coverage-5G | MTN-Coverage-5G |
| Uncapped Wireless | UncappedWirelessEBU | mtnsi:MTNSA-Coverage-Tarana | MTN-Coverage-UWA-EBU |
| Fibre | EBU-RBUS-ALL | mtnsi:MTNSA-Coverage-Fibre | MTN-Coverage-Fibre |
| Fixed LTE | EBU-RBUS-ALL | mtnsi:MTNSA-Coverage-FixedLTE | MTN-Coverage-FixedLTE |

#### GetFeatureInfo Request

Used to check coverage at specific coordinates:

GET https://mtnsi.mtn.co.za/cache/geoserver/wms?  
 mlid={MLID}&  
 LAYERS={LAYER}&  
 STYLES={STYLE}&  
 service=WMS&  
 version=1.1.1&  
 request=GetFeatureInfo&  
 bbox={BBOX}&  
 srs=EPSG:900913&  
 width=200&  
 height=200&  
 x=100&  
 y=100&  
 query\_layers={LAYER}&  
 info\_format=application/json&  
 feature\_count=50

**Parameters:** - mlid: Technology-specific map layer identifier - LAYERS: Layer name for the specific technology - STYLES: Style configuration for rendering - bbox: Bounding box in EPSG:900913 coordinates - srs: Spatial Reference System (EPSG:900913 - Spherical Mercator) - x,y: Point coordinates within the requested image - info\_format: Response format (application/json)

**Response:** JSON with coverage features if available at the location

### 2. MTN Public Coverage Query

**Base URL:** https://www.mtn.co.za/home/coverage/query

Legacy endpoint for general coverage queries.

GET https://www.mtn.co.za/home/coverage/query?  
 lat={LATITUDE}&  
 lng={LONGITUDE}&  
 type=all

**Parameters:** - lat: WGS84 latitude - lng: WGS84 longitude - type: Coverage type filter (all, mobile, fixed)

### 3. MTN Coverage API Point

**Base URL:** https://mtnsi.mtn.co.za/coverage/api/point

Point-based coverage API for specific locations.

POST https://mtnsi.mtn.co.za/coverage/api/point  
Content-Type: application/json  
  
{  
 "location": { "lat": -26.2041, "lng": 28.0473 },  
 "coverageTypes": ["3G", "4G", "5G", "UNCAPPED\_WIRELESS", "FIBRE", "FIXED\_LTE"]  
}

### 4. MTN Consumer API

**Base URL:** https://api.mtn.co.za/coverage/v1/availability

Consumer-facing availability API.

GET https://api.mtn.co.za/coverage/v1/availability?  
 latitude={LATITUDE}&  
 longitude={LONGITUDE}&  
 service=UNCAPPED\_WIRELESS

### 5. MTN Signal Quality API

**Base URL:** https://api.mtn.co.za/coverage/v1/signal

Signal quality and strength information.

GET https://api.mtn.co.za/coverage/v1/signal?  
 latitude={LATITUDE}&  
 longitude={LONGITUDE}

### 6. Infrastructure-Based Coverage Analysis (Fallback)

**Implementation:** Client-side intelligent prediction

When direct API access fails due to CORS restrictions, the application uses infrastructure-based analysis to predict coverage availability. This method:

* Analyzes proximity to major cities and infrastructure hubs
* Considers population density and urban development patterns
* Applies technology-specific deployment strategies
* Provides realistic coverage predictions based on MTN’s network rollout patterns

**Coverage Prediction Logic:**

|  |  |  |
| --- | --- | --- |
| Technology | Availability Criteria | Infrastructure Score Required |
| 2G | Basic coverage, rural areas | > 15 |
| 3G | Enhanced network areas | > 25 |
| 4G | Urban and suburban zones | > 35 |
| 5G | Major cities and high-density areas | > 70 |
| Uncapped Wireless | Specific coverage zones | > 60 |
| Fibre | Urban areas with infrastructure | > 75 |
| Fixed LTE | Alternative broadband areas | > 45 |

**Infrastructure Scoring Factors:** - Distance to major cities (Johannesburg, Cape Town, Durban, Pretoria) - Population density estimation - Urban vs rural classification - Provincial infrastructure development level

## Coordinate System Transformations

### WGS84 to EPSG:900913 (Spherical Mercator)

The GeoServer WMS requires coordinates in EPSG:900913 format:

function convertToSphericalMercator(lat, lng) {  
 const x = lng \* 20037508.34 / 180;  
 let y = Math.log(Math.tan((90 + lat) \* Math.PI / 360)) / (Math.PI / 180);  
 y = y \* 20037508.34 / 180;  
 return { x, y };  
}

## Technology Types

The application supports the following MTN technology types:

|  |  |  |
| --- | --- | --- |
| Type | Description | Use Case |
| 2G | Basic voice and SMS | Legacy coverage in remote areas |
| 3G | Voice, SMS, basic data | Enhanced mobile coverage |
| 4G | High-speed mobile data | Primary mobile data service |
| 5G | Ultra-fast mobile data | Next-generation mobile service |
| UNCAPPED\_WIRELESS | Fixed wireless broadband | Home/business internet via wireless |
| FIBRE | High-speed fibre connection | Premium fixed-line internet |
| FIXED\_LTE | Fixed LTE broadband | Alternative fixed-line internet |

## Error Handling

### Common HTTP Status Codes

* 200 OK: Successful request with coverage data
* 204 No Content: No coverage available at location
* 400 Bad Request: Invalid parameters
* 403 Forbidden: Access denied (CORS issues from localhost)
* 404 Not Found: Endpoint not available
* 500 Internal Server Error: Server-side error

### CORS Considerations and Fallback Strategy

When calling MTN APIs from localhost, CORS errors are expected. The application implements a robust fallback strategy:

1. **Primary Attempt**: Real MTN GeoServer WMS API calls
2. **CORS Detection**: Automatic detection of cross-origin blocking
3. **Intelligent Fallback**: Infrastructure-based coverage analysis
4. **Legacy Endpoints**: Additional API endpoint attempts
5. **User Transparency**: Clear indication of data source reliability

### Coverage Data Sources

The application provides transparency about data sources:

* **MTN GeoServer WMS**: Most accurate, real-time coverage data
* **Infrastructure Analysis**: Intelligent prediction based on network deployment patterns
* **Legacy APIs**: Alternative endpoints for validation
* **Cache**: Previously successful queries (5-minute TTL)

### Error Response Format

{  
 "success": false,  
 "coordinates": { "lat": -26.2041, "lng": 28.0473 },  
 "address": "Johannesburg, South Africa",  
 "province": "Gauteng",  
 "timestamp": "2025-01-15T10:30:00.000Z",  
 "coverage": {},  
 "errors": [  
 {  
 "endpoint": "geoserver",  
 "error": "CORS blocked request",  
 "fallback": "infrastructure-analysis"  
 }  
 ]  
}

## Rate Limiting

MTN APIs may implement rate limiting. Best practices:

* Cache results for 5 minutes per location
* Implement exponential backoff for failed requests
* Batch requests when possible
* Respect any rate limit headers returned

## Data Privacy

All MTN APIs respect user privacy:

* No personal information is transmitted
* Only coordinates and technology types are queried
* Results are cached locally and not shared with third parties

## Example Usage

### Complete Coverage Check (Recommended)

import { MTNApi } from './utils/mtnApi';  
  
const mtnApi = new MTNApi();  
  
// Check all available technologies at a location  
const result = await mtnApi.checkCoverage(  
 -26.2041, // Johannesburg latitude  
 28.0473, // Johannesburg longitude  
 "Johannesburg CBD, South Africa"  
);  
  
console.log('Coverage Result:', result);  
// {  
// success: true,  
// coordinates: { lat: -26.2041, lng: 28.0473 },  
// address: "Johannesburg CBD, South Africa",  
// province: "Gauteng",  
// coverage: {  
// mtnGeoServer: {  
// available: true,  
// types: [  
// { type: "2G", available: true, strength: "high" },  
// { type: "3G", available: true, strength: "high" },  
// { type: "4G", available: true, strength: "high" },  
// { type: "5G", available: true, strength: "medium" }  
// ],  
// source: "MTN GeoServer WMS"  
// }  
// }  
// }

### Manual GeoServer WMS Query

For direct WMS queries (advanced usage):

const lat = -26.2041; // Johannesburg  
const lng = 28.0473;  
const mercator = convertToSphericalMercator(lat, lng);  
const buffer = 100;  
const bbox = [  
 mercator.x - buffer,  
 mercator.y - buffer,  
 mercator.x + buffer,  
 mercator.y + buffer  
].join(',');  
  
// Check Uncapped Wireless coverage  
const url = `https://mtnsi.mtn.co.za/cache/geoserver/wms?` +  
 `mlid=UncappedWirelessEBU&` +  
 `LAYERS=mtnsi:MTNSA-Coverage-Tarana&` +  
 `STYLES=MTN-Coverage-UWA-EBU&` +  
 `service=WMS&version=1.1.1&request=GetFeatureInfo&` +  
 `bbox=${bbox}&srs=EPSG:900913&width=200&height=200&` +  
 `x=100&y=100&query\_layers=mtnsi:MTNSA-Coverage-Tarana&` +  
 `info\_format=application/json&feature\_count=50`;  
  
try {  
 const response = await fetch(url);  
 const data = await response.json();  
 console.log('Direct WMS result:', data);  
} catch (error) {  
 console.log('CORS blocked - using fallback method');  
}

### Infrastructure Analysis Example

// When CORS blocks direct API access, the system automatically  
// falls back to infrastructure-based analysis  
  
const result = await mtnApi.checkCoverage(-29.8587, 31.0218); // Durban  
// Result will include infrastructure analysis if direct API fails:  
// {  
// coverage: {  
// infrastructurePoint: {  
// available: true,  
// types: [  
// { type: "4G", strength: "high", infrastructureType: "Urban Network" }  
// ],  
// source: "MTN Infrastructure Analysis (Fallback)",  
// locationInfo: {  
// province: "KwaZulu-Natal",  
// infrastructureScore: 85,  
// nearestCity: "Durban"  
// }  
// }  
// }  
// }

## Official MTN Coverage Map

For reference and verification, the official MTN coverage map is available at: https://mtnsi.mtn.co.za/coverage/dev/v3/map3.html

This interactive map shows all coverage layers and can be used to verify the accuracy of API responses.

## Development Notes

### API Integration Best Practices

* **Primary Data Source**: MTN GeoServer WMS provides the most accurate, real-time coverage data
* **Fallback Strategy**: Always implement infrastructure-based analysis for CORS-restricted environments
* **Coordinate Validation**: All coordinates must be within South Africa bounds: lat: [-35, -22], lng: [16, 33]
* **Caching Strategy**: Implement 5-minute cache TTL to reduce API load and improve performance

### Technology-Specific Considerations

* **Uncapped Wireless**: Uses separate layer identifier (UncappedWirelessEBU) with specialized coverage patterns
* **5G Networks**: Limited to major metropolitan areas with high infrastructure scores (>70)
* **Fibre Coverage**: Concentrated in urban areas with established infrastructure (>75 score)
* **Legacy Networks (2G/3G)**: Broad coverage including rural areas

### Response Data Structure

* **Success Responses**: Include detailed feature properties for signal strength analysis
* **Error Handling**: Graceful degradation with multiple fallback strategies
* **Data Sources**: Clear indication of data source (GeoServer, Infrastructure Analysis, Cache)
* **Coverage Quality**: Strength indicators (high/medium/low) based on infrastructure analysis

### Testing and Validation

* **Urban Areas**: Expect high infrastructure scores (70-90+) with full technology availability
* **Suburban Areas**: Medium scores (40-70) with 2G/3G/4G coverage
* **Rural Areas**: Lower scores (15-40) with basic 2G/3G coverage
* **CORS Testing**: Test both direct API success and fallback scenarios

### Performance Optimization

* **Coordinate Precision**: Round to 4 decimal places for cache efficiency
* **Batch Requests**: Avoid rapid successive requests to prevent rate limiting
* **Error Recovery**: Implement exponential backoff for failed requests
* **User Feedback**: Provide clear indication of data source reliability