Czech Hydrometeorological Institute

API Reference Documentation
MET-SWIM Czech Republic

Base URL: https://swim.chmi.cz/api.php

Provided by: Czech Hydrometeorological Institute (CHMI), Aeronautical Meteorological Section

Documentation version: 2026-05

Czech Hydrometeorological Institute

Document Control

Field	Value
Document title	API Reference Documentation – MET-SWIM Czech Republic
Document owner	Czech Hydrometeorological Institute (CHMI), Aeronautical Meteorological Section
Current version	2026-06
Status	~~Draft~~ / ~~Approved~~ / **Published**
Last updated	2026-06-10
Contact	Aeronautical Meteorological Section

Version History

Version	Date	Author / Editor	Change summary
2026-06	2026-06-10	CHMI / Aeronautical Meteorological Section	Initial API reference documentation for MET-SWIM Czech Republic.

Versioning approach: The document version follows a YYYY-MM format and represents the month in which the API reference was reviewed or published. Minor editorial updates may be recorded as YYYY-MM.x when needed. The Version History table records user-visible changes, such as endpoint behaviour updates, parameter changes, file availability changes, or significant documentation corrections.

Czech Hydrometeorological Institute

Obsah

Czech Hydrometeorological Institute

1. API – Reference Guide

1.1 Overview

The MET-SWIM API provides programmatic access to aeronautical meteorological data produced by CHMI for the Czech Republic. It is a simple HTTP REST-style API that returns JSON metadata (for file listings) or raw file content (for downloads).

Key characteristics:

- **No authentication required** – all endpoints are publicly accessible
- **No registration required**
- **No rate limiting** for standard use
- **HTTPS supported** (recommended)
- **Read-only** – the API only exposes GET operations
- **Real-time** – files reflect the latest issued observations and forecasts

1.2 Endpoints

The API has a single endpoint with behaviour controlled by query parameters:

```
GET https://swim.chmi.cz/api.php
```

All parameters are passed as URL query string arguments. The method is always HTTP GET.

1.3 Parameters – Complete Reference

Parameter: action

Controls what the API returns.

Value	Behaviour	Notes
`list`	Returns a JSON list of available files matching the other filters	Default when omitted
`get`	Returns the raw content of a single file	Requires the `file` parameter

Parameter: format

Filters by file format (for action=list) or specifies the format of the file to fetch (for action=get).

Value	Description	Notes
`iwxxm`	IWXXM/XML files	Default when omitted
`txt`	TAC/plain-text files	

Note: *When using action=get, the format parameter must match the actual format of the requested file (i.e. use format=txt when fetching .txt files).*

Note: *Besides the defined „IWXXM“ format, „XML“ text is acceptable as well.*

Parameter: icao

Filters results by aerodrome or issuing station identifier.

Value	Description
Standard 4-letter ICAO aerodrome code	e.g. LKPR, LKTB, LKMT
WMO bulletin originator code	e.g. FRCZ60, SPCZ41 (TXT format only, see §3.2)

If omitted, files for all stations are returned.

> ***Note:*** *For TXT files, the icao field in responses may contain WMO bulletin designators (e.g. FRCZ62, SACZ33) rather than standard ICAO aerodrome codes. These represent FIR-level or multi-station forecast products.*

Parameter: type

Filters results by meteorological message type.

Value	Description	Formats available
`METAR`	Meteorological Aerodrome Report (routine observation)	IWXXM, TAC/TXT
`SPECI`	Special METAR (significant weather change)	IWXXM, TAC/TXT
`TAF`	Terminal Aerodrome Forecast	IWXXM, TAC/TXT
`SIGMET`	Significant Meteorological Information	IWXXM, TAC/TXT

If omitted, all types are returned. For TXT files, additional product types appear (e.g. LKPW, LKCS as type values for forecast bulletins) – these are WMO routing indicators, not standard ICAO message types.

Parameter: file

Used only with action=get. Specifies the exact filename to download.

Value	Description
Filename string	e.g. LKPR_METAR.xml, LKTB_TAF.txt

Filenames are obtained from the filename field in a preceding action=list response.

1.4 Parameter Combinations

The table below shows which parameter combinations are valid and what they return:

action	format	icao	type	file	Result
list	omitted	omitted	omitted	–	All IWXXM files for all stations
list	txt	omitted	omitted	–	All TXT files for all stations
list	omitted	LKPR	omitted	–	All IWXXM files for Prague
list	txt	LKTB	omitted	–	All TXT files for Brno
list	omitted	omitted	METAR	–	All METAR IWXXM files
list	omitted	omitted	TAF	–	All TAF IWXXM files
list	omitted	omitted	SIGMET	–	All SIGMET IWXXM files
list	omitted	LKPR	TAF	–	TAF IWXXM file for Prague only
list	txt	omitted	METAR	–	All METAR TXT files
get	iwxxm	–	–	LKPR_METAR.xml	Raw IWXXM content of Prague METAR
get	txt	–	–	LKPR_TAF.txt	Raw TXT content of Prague TAF

1.5 Response Object – action=list

A successful action=list response is a JSON object. Example:


```
{
  "status": "success",
  "query": {
    "icao": "LKPR",
    "type": "METAR",
    "format": "iwxxm"
  },
  "count": 1,
  "files": [
    {
      "icao": "LKPR",
      "type": "METAR",
      "format": "iwxxm",
      "filename": "LKPR_METAR.xml",
      "updated": "2026-05-13T06:04:02+00:00",
      "size": 2940,
      "download_url": "api.php?action=get&file=LKPR_METAR.xml&format=xml"
    }
  ]
}
```


Top-level fields

Field	Type	Description
status	string	"success" on a valid response. Error states may return a different value.

`query`	object	Mirror of the filter parameters used in the request (useful for debugging).
`query.icao`	string	The icao filter value used; empty string if not specified.
`query.type`	string	The type filter value used; empty string if not specified.
`query.format`	string	The format filter value used; always present.
`count`	integer	Total number of file entries in the files array. 0 means no matching files.
`files`	array	Array of file descriptor objects (see below). May be empty.

File descriptor object (files[])

Field	Type	Description
`icao`	string	Station or bulletin identifier. For aerodrome products this is a 4-letter ICAO code (e.g. LKPR). For TXT forecast bulletins this may be a WMO designator (e.g. FRCZ62).
`type`	string	Product type: METAR, SPECI, TAF, SIGMET, or a WMO routing indicator for forecast bulletins.
`format`	string	File format: iwxxm (xml) or txt.
`filename`	string	Full filename including extension (e.g. LKPR_METAR.xml). Use this value in the file parameter when calling action=get.
`updated`	string	ISO 8601 timestamp of the last file update in UTC with timezone offset (e.g. 2026-05-13T06:04:02+00:00).
`size`	integer	File size in bytes at the time of the listing.
`download_url`	string	Relative URL path to download the file. Prepend https://swim.chmi.cz/ to obtain the full URL.

> ***Note:*** *Constructing the full download URL: https://swim.chmi.cz/ + download_url. Example: https://swim.chmi.cz/api.php?action=get&file=LKPR_METAR.xml&format=xml*

1.6 File Download – action=get

When action=get is specified, the API returns the raw file content directly (not wrapped in JSON). The response Content-Type depends on the format:

Format	Content-Type
`iwxxm`	application/xml or text/xml
`xml`	application/xml or text/xml
`txt`	text/plain

Czech Hydrometeorological Institute

Required parameters for action=get:

Parameter	Required	Example
`action`	yes	`get`
`file`	yes	`LKPR_METAR.xml`
`format`	yes	`iwxxm`

If the file parameter is omitted or the filename does not exist, the API will return an error.

1.7 Filename Conventions

Standard aerodrome files (IWXXM and TXT)

```
{ICAO}_{TYPE}.{ext}
```

Part	Description	Example
`{ICAO}`	4-letter ICAO aerodrome code	`LKPR`
`{TYPE}`	Product type	`METAR`, `TAF`, `SPECI`, `SIGMET`
`{ext}`	File extension	`xml` or `txt`

Examples: LKPR_METAR.xml, LKMT_TAF.txt, LKKB_SPECI.xml, LKAA_SIGMET.txt

FIR and forecast bulletin files (TXT only)

```
{BULLETIN}_{STATION}.txt
```

Part	Description	Example
`{BULLETIN}`	WMO bulletin designator	`FRCZ60`, `SPCZ41`, `SACZ33`
`{STATION}`	Issuing station or FIR identifier	`LKPW`, `LKPR`, `LKTB`

Examples: FRCZ60_LKPW.txt, SPCZ41_LKBC.txt, WOCZ60_LKPR.txt

2. Query Examples

2.1 Listing Files

All IWXXM files (default)

```
https://swim.chmi.cz/api.php
```

Czech Hydrometeorological Institute

```
https://swim.chmi.cz/api.php?action=list
https://swim.chmi.cz/api.php?action=list&format=iwxxm
```

All TXT files

```
https://swim.chmi.cz/api.php?action=list&format=txt
```

Files for a specific aerodrome

Prague (LKPR) – IWXXM:

```
https://swim.chmi.cz/api.php?action=list&icao=LKPR
```

Prague (LKPR) – TXT:

```
https://swim.chmi.cz/api.php?action=list&icao=LKPR&format=txt
```

Brno (LKTB) – IWXXM:

```
https://swim.chmi.cz/api.php?action=list&icao=LKTB
```

Brno (LKTB) – TXT:

```
https://swim.chmi.cz/api.php?action=list&icao=LKTB&format=txt
```

Files by type

All METAR – IWXXM:

```
https://swim.chmi.cz/api.php?action=list&type=METAR
```

All METAR – TXT:

```
https://swim.chmi.cz/api.php?action=list&type=METAR&format=txt
```

All TAF – IWXXM:

```
https://swim.chmi.cz/api.php?action=list&type=TAF
```

All TAF – TXT:

```
https://swim.chmi.cz/api.php?action=list&type=TAF&format=txt
```

All SIGMET – IWXXM:

```
https://swim.chmi.cz/api.php?action=list&type=SIGMET
```

All SIGMET – TXT:

```
https://swim.chmi.cz/api.php?action=list&type=SIGMET&format=txt
```

All SPECI – IWXXM:

```
https://swim.chmi.cz/api.php?action=list&type=SPECI
```

2.2 Downloading Files

Prague METAR (IWXXM):

```
https://swim.chmi.cz/api.php?action=get&file=LKPR_METAR.xml&format=iwxxm
```

Prague METAR (TXT):

```
https://swim.chmi.cz/api.php?action=get&file=LKPR_METAR.txt&format=txt
```

Brno TAF (IWXXM):

```
https://swim.chmi.cz/api.php?action=get&file=LKTB_TAF.xml&format=iwxxm
```

Ostrava TAF (TXT):

```
https://swim.chmi.cz/api.php?action=get&file=LKMT_TAF.txt&format=txt
```

Prague FIR SIGMET (TXT):

```
https://swim.chmi.cz/api.php?action=get&file=LKAA_SIGMET.txt&format=txt
```

GA Forecast (TXT):

```
https://swim.chmi.cz/api.php?action=get&file=FRCZ40_LKPW.txt&format=txt
```

2.3 Advanced Combinations

TAF for Prague only (IWXXM):

```
https://swim.chmi.cz/api.php?action=list&icao=LKPR&type=TAF
```

TAF for Prague only (TXT):

```
https://swim.chmi.cz/api.php?action=list&icao=LKPR&type=TAF&format=txt
```

METAR for Karlovy Vary (IWXXM):

```
https://swim.chmi.cz/api.php?action=list&icao=LKKV&type=METAR
```

SPECI for Pardubice (IWXXM):

```
https://swim.chmi.cz/api.php?action=list&icao=LKPD&type=SPECI
```

3. File Type Reference

3.1 Standard Aerodrome Products (LKXX codes)

These files follow the {ICAO}_{TYPE}.{ext} naming convention.

Type	Format	Description
METAR	IWXXM, TAC	Routine surface meteorological observation from an aerodrome, issued every 30 minutes. Includes wind, visibility, weather phenomena, cloud cover, temperature, dew point, and QNH.
SPECI	IWXXM, TAC	Special (non-routine) METAR issued immediately when a significant change in conditions occurs. Same structure as METAR.
TAF	IWXXM, TAC	Terminal Aerodrome Forecast. Issued every 6 hours (LKPR: every 3 hours). Covers wind, visibility, weather, and cloud for the next 9, 18, or 30 hours. May contain AMD (amendment) or COR (correction) suffixes.
SIGMET	IWXXM, TAC	Significant Meteorological Information issued by the Meteorological Watch Office for the Prague FIR (LKAA). Covers thunderstorms, severe icing, severe turbulence, and volcanic ash.

3.2 FIR-Level and Forecast Products (WMO bulletin codes)

These files appear in TXT format only and use WMO T1T2A1A2ii originator designators.

Bulletin prefix	Product type	Description
FRCZ40_LKPW	GA Forecast	Forecast for General Aviation flights
FRCZ60_LKPW	Area forecast	General area forecast for Czech airspace
FRCZ62_{ICAO}	Take-off Forecast	Take-off forecast for major airports (LKPR, LKTB, LKMT, LKKV)
FTCZ31_LKPW	Forecast	Upper wind and temperature forecast

FTCZ32_LKPW	Forecast	GAMET (low-level significant weather chart)
SACZ31_LKPW	Area	Special area forecast / meteorological watch
SACZ33_{ICAO}	Aerodrome	Aerodrome warnings for multiple stations
SACZ45_{ICAO}	Aerodrome	Aerodrome warnings (military and smaller aerodromes)
SPCZ41_{ICAO}	SPECI	Special observation bulletins for smaller aerodromes
SPCZ44_{ICAO}	SPECI	Special observation bulletins for major aerodromes
WOCZ60_{ICAO}	Aerodrome warning	Aerodrome wind shear or low-level wind warnings
WSCZ31_LKPW	SIGMET	SIGMET bulletin for Prague FIR

Note: *WMO T1T2 designators: FR = area forecast, FT = upper-air forecast, SA = surface advisory, SP = special report, WO = operational warning, WS = SIGMET.*

4. Full List of Available Files

4.1 XML Files (IWXXM format)

As of May 2026 the following XML files are available (live API response):

Filename	Station	Type
LKBC_METAR.xml	Bechyně	METAR
LKBC_SPECI.xml	Bechyně	SPECI
LKCR_METAR.xml	Chrudim	METAR
LKCS_METAR.xml	České Budějovice	METAR
LKCV_METAR.xml	Čáslav	METAR
LKCV_SPECI.xml	Čáslav	SPECI
LKCV_TAF.xml	Čáslav	TAF
LKKB_METAR.xml	Praha – Kbely	METAR
LKKB_SPECI.xml	Praha – Kbely	SPECI
LKKB_TAF.xml	Praha – Kbely	TAF
LKKU_METAR.xml	Kunovice	METAR
LKKV_METAR.xml	Karlovy Vary	METAR
LKKV_TAF.xml	Karlovy Vary	TAF
LKLN_METAR.xml	Plzeň – Líně	METAR
LKLN_SPECI.xml	Plzeň – Líně	SPECI
LKMT_METAR.xml	Ostrava – Mošnov	METAR

LKMT_TAF.xml	Ostrava – Mošnov	TAF
LKNA_METAR.xml	Náměšť nad Oslavou	METAR
LKNA_SPECI.xml	Náměšť nad Oslavou	SPECI
LKNA_TAF.xml	Náměšť nad Oslavou	TAF
LKPD_METAR.xml	Pardubice	METAR
LKPD_SPECI.xml	Pardubice	SPECI
LKPD_TAF.xml	Pardubice	TAF
LKPJ_METAR.xml	Prostějov	METAR
LKPJ_SPECI.xml	Prostějov	SPECI
LKPR_METAR.xml	Praha – Ruzyně	METAR
LKPR_TAF.xml	Praha – Ruzyně	TAF
LKTB_METAR.xml	Brno – Tuřany	METAR
LKTB_TAF.xml	Brno – Tuřany	TAF
LKTH_METAR.xml	Těchonín	METAR
LKVO_METAR.xml	Praha – Vodochody	METAR

Note: *SIGMET files in XML format are event-driven and may not always be present. Query action=list&type=SIGMET to check current availability.*

4.2 TXT Files (TAC format) – Key Files

Standard aerodrome METAR/TAF

Filename	Station	Type
LKPR_METAR.txt	Praha – Ruzyně	METAR
LKPR_TAF.txt	Praha – Ruzyně	TAF
LKTB_METAR.txt	Brno – Tuřany	METAR
LKTB_TAF.txt	Brno – Tuřany	TAF
LKMT_METAR.txt	Ostrava – Mošnov	METAR
LKMT_TAF.txt	Ostrava – Mošnov	TAF
LKKV_METAR.txt	Karlovy Vary	METAR
LKKV_TAF.txt	Karlovy Vary	TAF
LKBC_METAR.txt	Bechyně	METAR
LKCR_METAR.txt	Chrudim	METAR
LKCS_METAR.txt	České Budějovice	METAR
LKCV_METAR.txt	Čáslav	METAR
LKKB_METAR.txt	Praha – Kbely	METAR

Czech Hydrometeorological Institute

LKKU_METAR.txt	Kunovice	METAR
LKLN_METAR.txt	Plzeň – Líně	METAR
LKNA_METAR.txt	Náměšť nad Oslavou	METAR
LKPD_METAR.txt	Pardubice	METAR
LKPJ_METAR.txt	Prostějov	METAR
LKTH_METAR.txt	Těchonín	METAR

FIR forecasts, advisories and SIGMET

Filename	Description
FRCZ40_LKPW.txt	General Aviation (GA) forecast for Czech Republic
FRCZ60_LKPW.txt	Area forecast for Prague FIR
FRCZ62_LKPR.txt	Take-off forecast – Prague
FRCZ62_LKTB.txt	Take-off forecast – Brno
FRCZ62_LKMT.txt	Take-off forecast – Ostrava
FRCZ62_LKKV.txt	Take-off forecast – Karlovy Vary
FTCZ31_LKPW.txt	Upper wind / temperature forecast
FTCZ32_LKPW.txt	GAMET / low-level forecast
LKAA_SIGMET.txt	Active SIGMET for Prague FIR
WSCZ31_LKPW.txt	SIGMET bulletin

Aerodrome warnings and SPECI bulletins

Filename	Description
SACZ31_LKPW.txt	METAR bulletin
SACZ33_LKPR.txt	METAR bulletin – Prague
SACZ33_LKTB.txt	METAR bulletin – Brno
SACZ33_LKMT.txt	METAR bulletin – Ostrava
SACZ33_LKKV.txt	METAR bulletin – Karlovy Vary
SACZ33_LKCS.txt	METAR bulletin – České Budějovice
WOCZ60_LKPR.txt	Aerodrome warning – Prague
WOCZ60_LKTB.txt	Aerodrome warning – Brno
WOCZ60_LKKV.txt	Aerodrome warning – Karlovy Vary
WOCZ60_LKMT.txt	Aerodrome warning – Ostrava

SPCZ41_LKBC.txt	SPECI bulletin – Bechyně
SPCZ41_LKCR.txt	SPECI bulletin – Chrudim
SPCZ44_LKPR.txt	SPECI bulletin – Prague
SPCZ44_LKTB.txt	SPECI bulletin – Brno
SPCZ44_LKMT.txt	SPECI bulletin – Ostrava

5. Code Examples

5.1 Python

List all METAR XML files

```
import requests

BASE_URL = "https://swim.chmi.cz/api.php"

def list_files(icao=None, msg_type=None, fmt='iwxxm'):
    params = {"action": "list", "format": fmt}
    if icao: params['icao'] = icao
    if msg_type: params['type'] = msg_type
    response = requests.get(BASE_URL, params=params, timeout=10)
    response.raise_for_status()
    return response.json()

# List all METAR XML files
result = list_files(msg_type='METAR')
print(f"Found {result['count']} METAR files")
for f in result['files']:
    print(f"  {f['filename']}  updated: {f['updated']}")
```

Download a specific file

```
def download_file(filename, fmt='iwxxm'):
    params = {"action": "get", "file": filename, "format": fmt}
    response = requests.get(BASE_URL, params=params, timeout=10)
    response.raise_for_status()
    return response.text

# Download the Prague METAR in XML
xml_content = download_file("LKPR_METAR.xml", fmt="xml")
print(xml_content[:500])
```

5.2 JavaScript / Node.js

```
const BASE_URL = "https://swim.chmi.cz/api.php";
```

```
async function listFiles({ icao, type, format = 'iwxxm' } = {}) {
  const params = new URLSearchParams({ action: 'list', format });
  if (icao) params.set('icao', icao);
  if (type) params.set('type', type);
  const res = await fetch(`${BASE_URL}?${params}`);
  if (!res.ok) throw new Error(`HTTP error: ${res.status}`);
  return res.json();
}

async function downloadFile(filename, format = 'iwxxm') {
  const params = new URLSearchParams({ action: 'get', file: filename, format });
  const res = await fetch(`${BASE_URL}?${params}`);
  if (!res.ok) throw new Error(`HTTP error: ${res.status}`);
  return res.text();
}

// Example: list all TAF XML files then download the first one
(async () => {
  const result = await listFiles({ type: 'TAF' });
  console.log(`Found ${result.count} TAF files`);
  if (result.files.length > 0) {
    const first = result.files[0];
    const content = await downloadFile(first.filename, first.format);
    console.log(content.slice(0, 500));
  }
})();
```

5.3 curl (command line)

```
# List all XML files
curl "https://swim.chmi.cz/api.php"

# List all TXT files
curl "https://swim.chmi.cz/api.php?action=list&format=txt"

# Download Prague METAR (XML)
curl "https://swim.chmi.cz/api.php?action=get&file=LKPR_METAR.xml&format=iwxxm"

# Download Prague METAR (TXT) and save to file
curl -o LKPR_METAR.txt \
  "https://swim.chmi.cz/api.php?action=get&file=LKPR_METAR.txt&format=txt"

# Pretty-print the JSON list with jq
curl -s "https://swim.chmi.cz/api.php?action=list&type=METAR" | jq .

# Download all current METAR XML files using jq and a shell loop
curl -s "https://swim.chmi.cz/api.php?action=list&type=METAR" \
  | jq -r '.files[].filename' \
  | while read fname; do
      curl -s -o "$fname" \
        "https://swim.chmi.cz/api.php?action=get&file=${fname}&format=iwxxm"
      echo "Saved: $fname"
    done
```

5.4 PowerShell

```
$baseUrl = "https://swim.chmi.cz/api.php"

# List all METAR XML files
$response = Invoke-RestMethod "$baseUrl?action=list&type=METAR"
Write-Host "Found $($response.count) METAR files"
$response.files | Format-Table icao, type, filename, updated, size

# Download a specific file
$content = Invoke-WebRequest `
  "$baseUrl?action=get&file=LKPR_METAR.xml&format=iwxxm"
$content.Content | Out-File -FilePath "LKPR_METAR.xml" -Encoding UTF8

# Download all current METAR TXT files
$files = (Invoke-RestMethod "$baseUrl?action=list&type=METAR&format=txt").files
foreach ($f in $files) {
    $url = "https://swim.chmi.cz/" + $f.download_url
    Invoke-WebRequest $url -OutFile $f.filename
    Write-Host "Saved: $($f.filename)"
}
```

6. Error Handling

The API currently does not document a formal error schema. Observed behaviour:

Situation	Result
No matching files for given filters	status: "success", count: 0, files: []
action=list with valid but filtered-out parameters	Empty files array (not an error)
action=get with a valid filename	Raw file content returned
action=get with missing file parameter	Behaviour undefined; treat as error
Network or server error	Standard HTTP 4xx/5xx status code

Recommended practices:

- Always check response.status in your application for unexpected values.
- Check count > 0 before accessing files[0] to avoid index errors.
- For action=get, verify the HTTP response status code (expect 200 OK) before consuming the body.
- Wrap all API calls in try/catch or equivalent error-handling to handle transient network failures.
- An empty files array is not an error – it simply means no files match the requested filters (e.g. SIGMET files when no SIGMET is currently active).

7. Update Schedule

Product	Issue Times (UTC)	Validity period
METAR (routine)	:00 and :30 past every hour	–
SPECI	Immediately upon significant change	–
METREPORT	Every 1 minute	–
TAF – LKPR	00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, 21:00	30 h
TAF – other aerodromes	00:00, 06:00, 12:00, 18:00	9 / 24 h
SIGMET	Event-driven	Max. 4 hours
Take-off Forecast	Every 3 hours	8 h
GA Forecast (FRCZ40)	Every 12 hours	12 h
Area Forecast (FRCZ60)	Every 6 hours	9 h

> ***Note:*** *Files on the server always reflect the latest issued version of each product. There is no historical archive accessible via this API.*

8. Technical Standards

Standard	Description
ICAO Annex 3	Meteorological Service for International Air Navigation (Amendment 79+)
WMO No. 306	Manual on Codes – International Codes
IWXXM 3.0	ICAO Meteorological Information Exchange Model
W3C XML Schema (XSD)	Schema validation for all XML files
OGC GML 3.2.1	Geography Markup Language for spatial features
UTF-8	Character encoding for all files
UTC	Time standard for all timestamps
WGS-84	Coordinate reference system for geospatial data
ISO 8601	Timestamp format in API responses
HTTP/HTTPS	Transport protocol (HTTPS recommended)

Czech Hydrometeorological Institute

9. IWXXM Message Structure

Each IWXXM XML file encodes a single meteorological message following the ICAO IWXXM 3.0 schema:

```
IWXXM Message
├── Header
│   ├── Message metadata (UUID, issue sequence)
│   ├── Issue time (UTC, ISO 8601)
│   └── Issuing centre (CHMI / MWO Prague)
│
├── Observation or Forecast Record
│   ├── Aerodrome location
│   │   ├── ICAO designator
│   │   ├── Coordinates (WGS-84 latitude/longitude)
│   │   └── Elevation (metres above MSL)
│   │
│   ├── Observation time (UTC)
│   │
│   └── Meteorological parameters
│       ├── Surface wind (direction, speed, gusts, variability)
│       ├── Visibility (prevailing and minimum)
│       ├── Runway Visual Range (RVR, if applicable)
│       ├── Present weather phenomena (WMO code table)
│       ├── Recent weather phenomena (if applicable)
│       ├── Cloud cover (FEW/SCT/BKN/OVC, height, CB/TCU)
│       ├── Air temperature and dew point (°C)
│       ├── QNH pressure (hPa)
│       ├── Runway state (if applicable)
│       └── Trend forecast (NOSIG / TEMPO / BECMG, if included)
│
└── Footer
    ├── XML namespace declarations
    └── Schema reference (xsi:schemaLocation)
```