**1) Public FTP test server (use these exact values)**

* **Address:** ftp.dlptest.com
* **Username:** dlpuser
* **Password:** rNrKYTX9g7z3RgJRmxWuGHbeu  
  *(this is a public test FTP provided for integration testing)*

**2) Create CPI Security Material (User Credentials)**

1. In CPI: **Monitor → Security Material → Create → User Credentials**
2. **Name:** FTP\_Test
3. **Username:** dlpuser
4. **Password:** rNrKYTX9g7z3RgJRmxWuGHbeu
5. Save.

**3) FTP Receiver Adapter settings (in your iFlow)**

* **Adapter Type:** FTP (Receiver)
* **Address:** ftp.dlptest.com
* **Proxy Type:** Internet
* **Encryption:** Plain FTP - no encryption
* **Credential Name:** FTP\_Test
* **Directory:** / (or /incoming if you prefer — public server usually uses root)
* **File Name:** ${property.FileName}.csv (or static output.csv)
* **Timeout (ms):** 10000 (default)

**Important:** Don’t put ftp:// or port in Address. If you set a custom port, put it in the adapter advanced/port field.

**4) Set dynamic filename (Content Modifier)**

Before the FTP receiver, add a **Content Modifier** and set a **Property**:

* **Name:** FileName
* **Type:** Expression
* **Value:** ${date:yyyyMMddHHmmss}

This makes filenames like 20251025xxxxxx.csv. (Or use a static name for quick tests.)

**5) (Optional) Groovy script to convert JSON → CSV**

Add an **Script** step (if your incoming payload is JSON and you want CSV):

import com.sap.gateway.ip.core.customdev.util.Message

import groovy.json.JsonSlurper

Message processData(Message message) {

def body = message.getBody(String)

def json = new JsonSlurper().parseText(body)

def csv = "id,name\n"

csv += "${json.id},${json.name}\n"

message.setBody(csv)

message.setHeader("Content-Type", "text/plain")

return message

}

If your payload is plain text already, skip the script.

**6) Deploy iFlow & Test connection (optional)**

* Deploy the iFlow.
* In the FTP receiver adapter configuration, use the **Test Connection** button. If CPI allows it, it should succeed with these public creds. (If Test Connection isn't available, just trigger the iFlow.)

**7) Test with curl / Postman**

Example curl:

curl -X POST "https://<your-cpi-endpoint>/http/uploadtoftp" \

-H "Content-Type: application/json" \

-d '{"id":"5001","name":"Merlin"}'

Replace <your-cpi-endpoint> with the HTTP endpoint shown after deploying the iFlow.

Expected result: CPI receives the JSON, Groovy converts it to CSV, and the FTP receiver posts a file named like 20251025xxxxxx.csv to ftp.dlptest.com root.

**8) How to verify the file on the FTP server**

* You can use an FTP client (FileZilla) to connect to ftp.dlptest.com using the dlpuser creds and check the root for the uploaded file.
* Or run ftp ftp.dlptest.com from terminal and ls to list files.

**9) Troubleshooting (500s / common errors)**

* **500 after deploy / runtime:** Check iFlow logs (Monitor → Message Processing → Search by Message ID) — look for “Authentication failed”, “Unknown host”, or “Connection refused”.
* **Authentication failed:** Ensure FTP\_Test matches Security Material name exactly and creds are correct.
* **Unknown host:** Recheck ftp.dlptest.com spelling (no http://).
* **Timeouts / Connection refused:** Network or server issue — try connecting with FileZilla from your PC to see if public FTP is reachable.

**10) Next steps after success**

* Once this works, you know iFlow → FTP is fine. For your local PC FTP:
  + Install Cloud Connector and map your local FTP host → then change Address to the Cloud Connector virtual host and Proxy Type = On-Premise and set Location ID.
  + Or keep using a hosted FTP/SFTP for production (recommended).