**✅ Jenkins Configuration for a Maven Project**

**🔧 Step 1: Install Jenkins (if not already)**

Download and install from:  
👉 https://www.jenkins.io/download/

Start Jenkins and open it in your browser (usually: http://localhost:8080)

**🔧 Step 2: Install Required Jenkins Plugins**

Go to:

* Manage Jenkins → Manage Plugins → **Available tab**

Search and install:

* ✅ **Maven Integration**
* ✅ **Git Plugin**
* ✅ (Optional) **Pipeline**

Restart Jenkins after installation.

**🔧 Step 3: Configure Global Tools**

Go to:

* Manage Jenkins → Global Tool Configuration

**➤ Configure Maven:**

* Click **Add Maven**
* Give a name: e.g., Maven 3.8.6
* Select: Install automatically (or manually specify Maven home if installed)

**➤ Configure JDK:**

* Add JDK (or use system default)
* Give it a name: e.g., JDK 17

**➤ Configure Git:**

* Make sure Git is installed and detected

**🔧 Step 4: Create a Maven Job**

1. Go to Jenkins dashboard
2. Click **"New Item"**
3. Name your job (e.g., My-Maven-Project)
4. Select **"Maven Project"** → Click **OK**

**🔧 Step 5: Configure the Maven Job**

**➤ Source Code Management:**

* Select **Git**
* Enter your Git repo URL (e.g., https://github.com/username/project.git)
* If private, provide credentials

**➤ Build Triggers (Optional):**

* You can enable:
  + Poll SCM (H/5 \* \* \* \*)
  + Build periodically
  + GitHub webhook triggers

**➤ Build Environment:**

* (Optional) Clean workspace before build

**➤ Build:**

* Choose the Maven version you added earlier
* Add your Maven goals:

bash

CopyEdit

clean install

or

bash

CopyEdit

clean verify

**🔧 Step 6: Save and Build**

* Click **Save**
* Click **"Build Now"**
* Monitor console output: **Console Output** → see if it builds successfully

**🎯 Optional: Post-Build Actions**

* Add **email notifications**
* Publish test reports or artifacts

**✅ Example: Maven Goal Breakdown**

| **Goal** | **What it does** |
| --- | --- |
| clean | Removes previous build files |
| compile | Compiles Java source files |
| test | Runs unit tests |
| package | Packages into JAR/WAR |
| install | Installs into local .m2 |
| verify | Runs integration tests |

Would you like a **Pipeline (Jenkinsfile)** setup instead of the Maven job UI? I can give you a sample Jenkinsfile too.

Ask ChatGPT

**You said:**

HTTP ERROR 403 No valid crumb was included in the request URI: /manage/descriptorByName/hudson.model.JDK/checkName STATUS: 403 MESSAGE: No valid crumb was included in the request SERVLET: Stapler

**ChatGPT said:**

The **HTTP ERROR 403: "No valid crumb was included in the request"** in Jenkins means that **Jenkins is rejecting your request due to missing or invalid CSRF protection (crumb token).**

This typically happens when:

* You're making a request (via browser or script) without a CSRF "crumb"
* Jenkins' security settings require a crumb token for POST or configuration requests

**✅ How to Fix “No valid crumb was included in the request”**

**🔧 Option 1: Enable CSRF Crumb Issuer (if not already)**

1. Go to Jenkins → Manage Jenkins
2. Click on Configure Global Security
3. Look for the **"CSRF Protection"** section
4. Ensure **"Enable Crumb Issuer"** is **checked**
5. Under **Crumb Algorithm**, you can select:
   * Default Crumb Issuer (safe for most use)
   * Or choose one based on your setup

✅ Then click **Save**

**🔧 Option 2: Use Crumb Header in API / Script**

If you're using a script (e.g., curl, Python, Postman), fetch the crumb first:

bash

CopyEdit

curl -u youruser:yourtoken http://localhost:8080/crumbIssuer/api/json

You’ll get something like:

json

CopyEdit

{

"crumb": "abc123...",

"crumbRequestField": ".crumb"

}

Then include it in your request header:

bash

CopyEdit

curl -X POST http://localhost:8080/job/your-job-name/build \

-H ".crumb: abc123..." \

-u youruser:yourtoken

**🔧 Option 3: Disable CSRF Protection (⚠️ Not Recommended)**

Only do this for **testing on a local/dev environment** — it's insecure for public or shared Jenkins servers.

1. Go to Jenkins → Manage Jenkins → Configure Global Security
2. Uncheck **"Prevent Cross Site Request Forgery exploits"**
3. Save

**🧠 Why This Happens**

Jenkins uses "crumbs" (CSRF tokens) to prevent unauthorized commands from malicious pages. When this protection is enabled, **every POST request must include a valid crumb**.

If you tell me exactly **what you were doing** when this error occurred (e.g., configuring JDK, running a script, using a plugin), I can tailor the fix for your case.

Top of Form



Attach

Search

Voice

Bottom of Form

ChatGPT can make mistakes. Check important info. See Cookie Preferences.