Here is a **detailed guided practice ICN plugin project** with a **concise Eclipse-importable plugin template** to **practically master ICN plugin development**.

**🚀 Guided Practice ICN Plugin Project**

**🎯 Goal:**

**Build an ICN Plugin that:**  
✅ Adds a **context menu action** (“Stamp Document”) to selected documents.  
✅ Sends document metadata to a **custom Java service**.  
✅ Updates a custom property (StampedBy) with the logged-in user’s name.  
✅ Displays a confirmation dialog in ICN.

**🪜 Step-by-Step Plan**

**✅ 1️⃣ Setup Environment**

✅ Ensure:

* ICN is installed and running.
* Access to ICN Admin Desktop.
* Eclipse IDE with:
  + navigatorAPI.jar (from ICN\_INSTALL\_PATH/navigator.ear/navigator.war/WEB-INF/lib/).
  + Java 8.

**✅ 2️⃣ Create Plugin Project in Eclipse**

1️⃣ File → New → Java Project → ICNStampPlugin.  
2️⃣ Create folders:

/src/com/turnpike/icnplugin

/WebContent

3️⃣ Add navigatorAPI.jar to your **Build Path** (do not export in JAR).

**✅ 3️⃣ Create the Plugin Main Class**

src/com/turnpike/icnplugin/StampPlugin.java:

package com.turnpike.icnplugin;

import com.ibm.ecm.extension.Plugin;

public class StampPlugin extends Plugin {

@Override

public String getId() {

return "StampPlugin";

}

@Override

public String getName() {

return "Stamp Plugin";

}

@Override

public String getVersion() {

return "1.0";

}

@Override

public String getDojoModule() {

return "stampPlugin";

}

}

**✅ 4️⃣ Create the Service Class**

src/com/turnpike/icnplugin/StampService.java:

package com.turnpike.icnplugin;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.Iterator;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.ibm.ecm.extension.PluginService;

import com.ibm.ecm.extension.PluginServiceCallbacks;

import com.ibm.json.java.JSONArray;

import com.ibm.json.java.JSONObject;

public class StampService extends PluginService {

@Override

public String getId() {

return "StampService";

}

@Override

public void execute(PluginServiceCallbacks callbacks, HttpServletRequest request, HttpServletResponse response) throws IOException {

String userId = request.getRemoteUser();

String repositoryId = request.getParameter("repositoryId");

String docIdsJson = request.getParameter("docIds");

JSONArray docIds = JSONArray.parse(docIdsJson);

for (Object obj : docIds) {

String docId = (String) obj;

try {

com.filenet.api.core.ObjectStore os = callbacks.getObjectStore(repositoryId);

com.filenet.api.core.Document doc = com.filenet.api.core.Factory.Document.fetchInstance(os, docId, null);

doc.getProperties().putValue("StampedBy", userId);

doc.checkin(AutoClassify.DO\_NOT\_AUTO\_CLASSIFY, CheckinType.MAJOR\_VERSION);

doc.save(RefreshMode.REFRESH);

} catch (Exception e) {

e.printStackTrace();

}

}

response.setContentType("application/json");

PrintWriter out = response.getWriter();

JSONObject result = new JSONObject();

result.put("status", "success");

result.put("message", "Documents stamped by: " + userId);

out.print(result.toString());

}

}

**✅ 5️⃣ Create JavaScript Action**

WebContent/stampPluginAction.js:

define([

"dojo/\_base/declare",

"ecm/model/Request",

"ecm/model/Teamspace",

"ecm/widget/dialog/MessageDialog"

], function(declare, Request, Teamspace, MessageDialog) {

return declare("stampPlugin.StampAction", null, {

isEnabled: function(repository, items, callback) {

callback(items.length > 0);

},

execute: function(repository, items, callback) {

var docIds = [];

for (var i = 0; i < items.length; i++) {

docIds.push(items[i].id);

}

Request.invokePluginService("StampPlugin", "StampService", {

requestParams: {

repositoryId: repository.id,

docIds: JSON.stringify(docIds)

},

requestCompleteCallback: function(response) {

var dialog = new MessageDialog({

title: "Stamp Completed",

message: response.message

});

dialog.show();

if (callback) callback();

}

});

}

});

});

**✅ 6️⃣ Create plugin.xml**

WebContent/plugin.xml:

<plugin id="StampPlugin" name="Stamp Plugin" version="1.0">

<services>

<service id="StampService" className="com.turnpike.icnplugin.StampService"/>

</services>

<actions>

<action id="StampAction" className="stampPlugin.StampAction" name="Stamp Document" iconPath="images/stamp.png" context="Document"/>

</actions>

</plugin>

**✅ 7️⃣ Package and Deploy**

1️⃣ Export as a .jar:

* Right-click → Export → JAR File → Select /src and WebContent → Export ICNStampPlugin.jar.

2️⃣ Go to **ICN Admin Desktop**:

* Settings → Plug-ins → Add Plug-in → Upload ICNStampPlugin.jar.
* Assign to your Desktop and Repository.

**✅ 8️⃣ Configure Action in ICN**

* Add “Stamp Document” action to **Document Context Menu** in ICN admin.
* Save and refresh ICN.

**✅ 9️⃣ Test**

1. Login to ICN.
2. Right-click on a document → select “Stamp Document”.
3. Confirm:
   * Dialog shows “Documents stamped by: yourUserID”.
   * Property StampedBy updated in document properties.

**📦 Ready-to-Import Concise Plugin Template**

If you want a **ready-to-import Eclipse project with clean structure**, I can prepare:  
✅ ICNStampPlugin.zip containing:

* Source (src/)
* WebContent/ with plugin.xml and JS
* Preconfigured .classpath and .project
* README for deployment