**Lab Exercise 4– Proxy Chaining in Apigee (Using LocalTargetConnection)**

**Objective**

* Create **two API proxies**:
  + proxy-A (main proxy that users call)
  + proxy-B (internal proxy providing response)
* Use **LocalTargetConnection** in proxy-A to forward request to proxy-B
* Return a response to the client via proxy-A

**Step 1: Create Proxy B (Backend Logic)**

1. Go to **Apigee Console** → **Develop > API Proxies**
2. Click **+ CREATE PROXY**
3. Choose **Reverse Proxy**, then click **Next**
4. Fill in:
   * **Proxy Name**: proxy-b
   * **Base Path**: /proxy-b
   * **Target URL**: https://mocktarget.apigee.net
5. Click **Next** through the wizard and click **Create**
6. In Develop tab, you can leave it as-is (acts as backend)

**Deploy** proxy-b to your environment (e.g., eval)

**Step 2: Create Proxy A (Main Proxy)**

1. Again, go to **+ CREATE PROXY**
2. Choose **Reverse Proxy** → click **Next**
3. Fill in:
   * **Proxy Name**: proxy-a
   * **Base Path**: /proxy-a
   * **Target URL**: Leave anything for now (we will override it)
4. Click **Next** and then **Create**

**Step 3: Configure Proxy A to Call Proxy B**

1. Open proxy-a → **Develop** tab
2. In left panel, open TargetEndpoints > default
3. Replace the <HTTPTargetConnection> block with:

<LocalTargetConnection>

<APIProxy>proxy-b</APIProxy>

<ProxyEndpoint>default</ProxyEndpoint>

</LocalTargetConnection>

This makes proxy-a internally forward requests to proxy-b.

**Step 4: Deploy Both Proxies**

1. Go to **Develop > Deployments**
2. Deploy both proxy-a and proxy-b to the **same environment** (e.g., eval)

**Step 5: Test Proxy A**

Use curl or Postman:

curl https://your-env-group.apigee.net/proxy-a

**Expected behavior:**

* proxy-a internally routes the request to proxy-b
* proxy-b calls mocktarget.apigee.net and returns the result to proxy-a
* You receive the final response via proxy-a

**Step 6 (Optional): Customize Proxy B to Return a Message**

If you want proxy-b to return a fixed message, do this:

1. In proxy-b, open TargetEndpoint > default
2. Replace <HTTPTargetConnection> with:

<LocalTargetConnection/>

1. Add an **AssignMessage** policy in PostFlow > Response:

<AssignMessage name="ReturnMessage">

<Set>

<Payload contentType="text/plain">Hello from Proxy B!</Payload>

</Set>

<AssignTo createNew="true" type="response"/>

</AssignMessage>

Then redeploy proxy-b and test again.