📘 Oracle APEX Vanity URL Reverse Proxy Setup

🔗 Goal

- Make your **Oracle APEX** app clean and professional with a short vanity URL:

```
https://supplier.unionsystechnologies.com/login
```

- **Hide** the backend ORDS path (`/ords/...`) and port (`8443`).
- Run **NGINX** as a reverse proxy on your **public server**.

🔗 Real Setup

Item	Value
Domain Name (DNS)	`supplier.unionsystechnologies.com`
Public IP	`106.201.234.138`
Backend Private IP	`192.168.0.177`
Backend Port	`8443`
Backend Path	`/ords/r/ust/unionsys-ats/login`
Proxy	`NGINX` on same server

🔗 Traffic Flow — How It Works

1️⃣ **User** visits:

```
https://supplier.unionsystechnologies.com/login
```

2️⃣ **Public DNS** resolves:

```
supplier.unionsystechnologies.com → 106.201.234.138
```

3️⃣ The request reaches your **NGINX reverse proxy** listening on port `443` (HTTPS).

4️⃣ **NGINX**:

- Accepts the public HTTPS request.
- Opens a **new HTTPS request** internally to:

```
https://192.168.0.177:8443/ords/r/ust/unionsys-ats/login
```

- Adds correct `Host` and `SNI` headers so Jetty doesn't reject with **Invalid SNI**.

5️⃣ **Jetty (ORDS)**:

- Receives the request.
- Validates the SNI.
- Responds with the APEX login page.

6️⃣ **NGINX**:

- Passes the response back to the user.
- **User never sees** the internal IP, port, or ORDS path.

🔗 Final `/etc/hosts` (Optional)

If you want to use the **hostname** instead of the IP for backend, add:

```
192.168.0.177 supplier.unionsystechnologies.com
```

This makes sure the backend Jetty sees the correct hostname and the **TLS handshake (SNI)** works naturally.

🔗 Final `NGINX` Config Example

Save as: `/etc/nginx/sites-available/apex.conf`

```
##########################################
# HTTP → HTTPS Redirect
##########################################

server {
    listen 80;
    server_name supplier.unionsystechnologies.com;

    return 301 https://$host$request_uri;
}
```

```
##########################################
# Main HTTPS Reverse Proxy
##########################################

server {
    listen 443 ssl;
    server_name supplier.unionsystechnologies.com;

    # Replace with your SSL cert paths!
    ssl_certificate /etc/ssl/certs/your_cert.crt;
    ssl_certificate_key /etc/ssl/private/your_cert.key;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    ##########################################
    # Vanity URL: /login → ORDS backend path
    ##########################################

    location /login {
        proxy_pass https://192.168.0.177:8443/ords/r/ust/unionsys-ats/login;

        # ✅ Fix: Send correct hostname in TLS handshake (SNI)
        proxy_ssl_server_name on;

        # ✅ Fix: Force Host header for Jetty to match cert
        proxy_set_header Host supplier.unionsystechnologies.com;

        # Good practice: Pass client IP info
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Real-IP $remote_addr;
    }

    ##########################################
    # Optional: Root → /login
    ##########################################
    location = / {
        return 301 /login;
    }
}
```

🔗Commands to Apply

1️⃣Save as:

```
/etc/nginx/sites-available/apex.conf
```

2️⃣Link:

```
sudo ln -s /etc/nginx/sites-available/apex.conf /etc/nginx/sites-enabled/
```

3️⃣Test:

```
sudo nginx -t
```

4️⃣Reload:

```
sudo systemctl reload nginx
```

🔗How to Test

Local test:

```
curl -I https://supplier.unionsystechnologies.com/login
```

Expected: `HTTP/1.1 200 OK` or a redirect — no `400 Invalid SNI`.

🔗Troubleshooting

Problem	Fix
400 Invalid SNI	Check `proxy_ssl_server_name on;` and `proxy_set_header Host supplier.unionsystechnologies.com;`
Backend not reachable	Confirm `192.168.0.177:8443` is reachable from proxy server (`curl https://192.168.0.177:8443/...`)
SSL cert mismatch	The backend cert must be valid for `supplier.unionsystechnologies.com`.

🔗 Security Tip

- Restrict port `8443` in your firewall so only `106.201.234.138` can access it internally.
- Use strong SSL ciphers and keep your certs renewed.

🗜️ **You now have a clean, production-ready Oracle APEX vanity URL with NGINX.**