Proper JWT Setup for watsonx Orchestrate Embed

Based on IBM's official documentation, the recommended approach for embedded chat is to use **JWT tokens generated from your backend server**, not anonymous access.

Why JWT Instead of Anonymous Access?

- ☑ More secure and production-ready
- ☑ Allows passing context variables to agents
- ☑ Better control over user identification
- ☑ No agent-level permission configuration needed
- ☑ Supports user authentication and session management

Current Issue

Your chat is stuck on "Connecting..." because:

1. Instance security is disabled (anonymous mode)
2. But agents still require individual configuration for anonymous access
3. This configuration may not be available in your watsonx Orchestrate instance

Solution: Switch to JWT Authentication

Option 1: Use IBM's Test Token (if available)

Note: There is no UI to generate JWT tokens in watsonx Orchestrate. You need to:

- Request a test token from your IBM account representative, OR
- Set up the backend server (see Option 2 below)

If you have a JWT token:

1. **Update** `.env.local`:

```
# Comment out security disabled mode
# VITE_WXO_SECURITY_DISABLED=true

# Add your JWT token
VITE_WXO_JWT=your_generated_jwt_token_here
```

2. **Re-enable security** on your instance:

```
cd path/to/wxO-embed-chat-security-tool
./wxO-embed-chat-security-tool.sh
# When prompted, do NOT choose "disable security"
# Choose to enable/configure security instead
```

3. **Restart dev server** and test

Option 2: Production Backend (Recommended)

Create a Node.js backend to generate JWTs dynamically:

```
// backend/routes/createJWT.js
const fs = require('fs');
const jwtLib = require('jsonwebtoken');
const crypto = require('crypto');
const express = require('express');
const router = express.Router();

// Your private key (keep secure on server)
const PRIVATE_KEY = fs.readFileSync('./keys/private.key');
const PUBLIC_KEY = fs.readFileSync('./keys/public.pub');

function createJWTString(userId, context) {
  const jwtContent = {
    sub: userId,  // User ID
    user_payload: {
      name: 'Chain AI User',
      custom_message: 'Encrypted payload',
    },
    context: context || {}  // Pass context to agents
  };

  // Encrypt user_payload
  const dataString = JSON.stringify(jwtContent.user_payload);
  const encryptedBuffer = crypto.publicEncrypt(
    {
      key: PUBLIC_KEY,
      padding: crypto.constants.RSA_PKCS1_OAEP_PADDING,
      oaepHash: 'sha256'
    },
    Buffer.from(dataString, 'utf-8')
  );
  jwtContent.user_payload = encryptedBuffer.toString('base64');

  // Sign JWT
  return jwtLib.sign(jwtContent, PRIVATE_KEY, {
    algorithm: 'RS256',
    expiresIn: '24h',
  });
}

router.get('/', (req, res) => {
  const userId = req.query.user_id || `anon-${Date.now()}`;

  // Add context variables for your agents
  const context = {
    crisis_type: req.query.crisis_type,
    region: req.query.region,
    severity: req.query.severity
  };

  const jwt = createJWTString(userId, context);
  res.send(jwt);
});

module.exports = router;
```

Then update your frontend to fetch JWT from backend:

```
// src/services/watsonx-auth.ts
export async function generateAuthToken(): Promise<string> {
  const ENV_JWT = import.meta.env.VITE_WXO_JWT;

  if (ENV_JWT) {
    return ENV_JWT;
  }

  // Fetch JWT from your backend
  const response = await fetch('http://localhost:3003/createJWT?user_id=chainai-user');
  return await response.text();
}
```

Option 3: Try Different Agent

Some agents may work with anonymous access while others don't. Try the Supervisor Agent instead:

1. Click on "Supervisor Agent" instead of "Communicator Agent"
2. See if it connects

Verify Agent Configuration

In watsonx Orchestrate UI:

1. Go to **Agents** section
2. Find **Communicator Agent** (ID: `b9b35094-185e-4aed-a537-b4dc8435cae8`)
3. Check:
    - ☑ Agent is Published (not Draft)
    - ☑ Environment `208dc747-4863-4d1e-810d-44b66a605c52` exists and is deployed
    - ☑ Agent has "Embedded Chat" enabled (if option exists)

Next Steps

1. **Try Option 1 first** (hardcoded JWT) - quickest to test
2. If that works, implement **Option 2** for production
3. If none work, contact IBM Support to verify agent configuration

Resources

- IBM watsonx Orchestrate Embed Documentation
- Your instance: https://us-south.watson-orchestrate.cloud.ibm.com
- Instance ID: `6e4a398d-0f34-42ad-9706-1f16af156856`