

# VS Code Port Forwarding Feature Documentation

---

## Prerequisites

- VS Code version 1.82 or later
- A GitHub account (required for authentication and port forwarding)
- An active internet connection
- A locally running service or application

## Accessing the Ports View

### 1. Method 1: Through the Panel

- Open the **Panel** region in VS Code (bottom panel)
- Click on the **Ports** tab

### 2. Method 2: Through the View Menu

- Go to **View > Ports** in the top menu

### 3. Method 3: Command Palette

- Press **Ctrl+Shift+P** (Windows/Linux) or **Cmd+Shift+P** (Mac)
- Type "Ports: Focus on Ports View" and select it

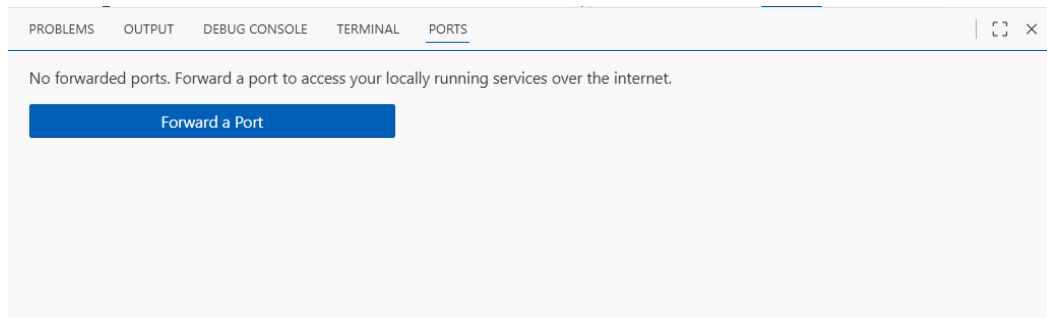
## How to Use Port Forwarding

---

### Basic Port Forwarding

#### 1. Open the Ports View

- Navigate to the Ports tab in the Panel region



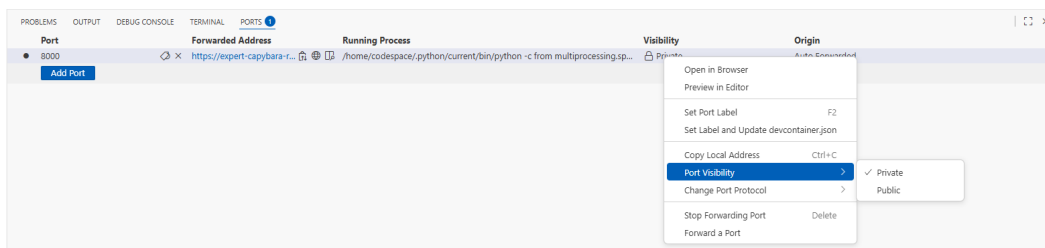
#### 2. Forward a Port

- Click the **Forward a Port** button (or the **+** icon)
- Enter the local port number (e.g., 3000, 8080, 5000)
- Press Enter or click the checkmark

Port	Forwarded Address	Running Process	Visibility	Origin
8000	https://expert-capibara-t4xg45jp5...	/home/codespace/python/current/bin/python -c from multiprocessing.sp...	Private	Auto Forwarded
Add Port				

### 3. Access Your Service

- VS Code will generate a public URL
- Copy and share this URL with others
- The URL format is typically: `https://[random-string].vscode.dev`



### Port Forwarding Options

When forwarding a port, you can configure:

- **Port Number:** The local port to forward
- **Label:** A custom name for easy identification
- **Privacy:** Public or private access
- **Protocol:** HTTP or HTTPS

### Managing Forwarded Ports

- **View Status:** See all forwarded ports in the Ports view
- **Stop Forwarding:** Click the stop button next to any port
- **Edit Settings:** Right-click on a port to modify its configuration
- **Copy URL:** Right-click to copy the public URL

### Practical Use Cases

#### 1. Web Development

**Local Server:** `http://localhost:3000`

**Forwarded URL:** `https://abc123.vscode.dev`

- Share your development server with clients for real-time feedback
- Test responsive designs on different devices
- Collaborate with team members on UI/UX changes

#### 2. API Development and Testing

**Local API:** `http://localhost:8080/api`

**Forwarded URL:** `https://def456.vscode.dev/api`

- Allow external services to consume your local APIs
- Test webhook integrations
- Share API endpoints with frontend developers

### **3. Database Management**

**Local Database Admin:** `http://localhost:8081`

**Forwarded URL:** `https://ghi789.vscode.dev`

- Access database management tools remotely
- Share database schemas with team members
- Perform remote database administration

### **4. Development Tools**

- **Storybook:** Share component libraries
- **Swagger/OpenAPI:** Share API documentation
- **Monitoring Dashboards:** Share development metrics

## **Security Considerations**

---

### **Access Control**

- **Public vs Private:** Choose appropriate privacy settings
- **Authentication:** Implement authentication for sensitive services
- **Time Limits:** Set expiration times for forwarded ports

### **Best Security Practices**

- Only forward necessary ports
- Avoid forwarding services with sensitive data
- Regularly monitor forwarded port usage
- Use HTTPS when possible
- Implement proper authentication mechanisms

### **Sensitive Services to Avoid**

- Database servers with production data
- Services containing API keys or secrets
- Administrative interfaces without authentication
- File systems with sensitive documents

## **Troubleshooting**

---

### **Common Issues**

#### **Port Already in Use**

- Check if the port is already forwarded
- Stop the existing forwarding session
- Choose a different port number

#### **Connection Refused**

- Verify the local service is running
- Check firewall settings
- Ensure the service is bound to the correct interface

## Slow Performance

- Check internet connection speed
- Consider the geographic location of users
- Optimize your local application

## Debug Steps

### 1. Verify Local Service

Test local access:

```
$ curl http://localhost:3000
```

### 2. Check Port Status

- Review the Ports view for error messages
- Check the VS Code output panel for logs

### 3. Network Diagnostics

- Test with different browsers
- Try from different network locations
- Check for proxy or firewall interference

## Best Practices

---

### Development Workflow

1. **Start Local Service:** Ensure your application is running locally
2. **Forward Port:** Use the Ports view to forward the appropriate port
3. **Share URL:** Distribute the public URL to stakeholders
4. **Monitor Usage:** Keep track of who accesses your forwarded service
5. **Clean Up:** Stop forwarding when no longer needed

### Team Collaboration

- Use descriptive labels for forwarded ports
- Document which ports are used for which services
- Establish team guidelines for port forwarding usage
- Regular cleanup of unused forwarded ports

### Performance Optimization

- Forward only necessary ports
- Use appropriate privacy settings
- Monitor bandwidth usage
- Consider using local development alternatives when possible

## Conclusion

---

VS Code's port forwarding feature is a powerful tool for modern development workflows. It enables seamless collaboration, facilitates testing across different environments, and supports various development scenarios. While other IDEs offer similar functionality, VS Code's implementation stands out for its ease of use, built-in GitHub integration, and intuitive interface.

By following security best practices and proper configuration, developers can leverage this feature effectively while maintaining security and performance.

For the most up-to-date information and additional features, refer to the official VS Code documentation and release notes.