Redis Quickstart Guide for Microsoft Login API

🚀 Quick Installation

macOS (using Homebrew)

```bash
# Install Redis
brew install redis

# Start Redis as a service (runs in background)
brew services start redis

# Or run Redis in foreground (for testing)
redis-server
```

Ubuntu/Debian

```bash
# Update package index
sudo apt update

# Install Redis
sudo apt install redis-server -y

# Start Redis
sudo systemctl start redis-server

# Enable Redis to start on boot
sudo systemctl enable redis-server

# Check status
sudo systemctl status redis-server
```

Windows

```bash
```

```
# Option 1: Use WSL2 (Recommended)
# Install WSL2 first, then follow Ubuntu instructions

# Option 2: Download Redis for Windows
# Download from: https://github.com/microsoftarchive/redis/releases
# Extract and run redis-server.exe
```

🔧 Basic Configuration

1. Test Redis Connection

bash

```bash
# Connect to Redis CLI
redis-cli

# Test connection
127.0.0.1:6379> ping
PONG

# Exit
127.0.0.1:6379> exit
```

2. Update Your .env File

bash

```bash
# For local development (no password)
REDIS_HOST=localhost
REDIS_PORT=6379
REDIS_DB=0
# REDIS_PASSWORD=  # Leave commented for local dev

# Or use connection URL
# REDIS_URL=redis://localhost:6379/0
```

🔒 Security (Optional for Development)

Set a Password

bash

```
# Edit Redis config
# macOS: /usr/local/etc/redis.conf
# Linux: /etc/redis/redis.conf

# Add or uncomment:
requirepass your_strong_password_here

# Restart Redis
# macOS: brew services restart redis
# Linux: sudo systemctl restart redis-server
```

Then update your `.env`:

```bash
REDIS_PASSWORD=your_strong_password_here
```

📊 Monitoring Redis

Using Redis CLI

```bash
# Monitor all commands in real-time
redis-cli monitor

# Get Redis info
redis-cli info

# Check memory usage
redis-cli info memory

# List all keys (careful in production!)
redis-cli keys "*"

# Check specific keys for your app
redis-cli keys "auth_state:*"
redis-cli keys "refresh_token:*"
```

GUI Tools (Optional)

- **RedisInsight** (Official, Free): https://redis.io/insight/

- **Medis** (macOS): Mac App Store
- **Redis Desktop Manager**: https://resp.app/

🧪 Testing Your Setup

1. Start Your FastAPI App

```bash
python main.py
```

You should see:

```
INFO:     Redis connected successfully
INFO:     Starting up...
INFO:     Application startup complete.
```

2. Check Health Endpoint

```bash
curl http://localhost:8000/health
```

Should return:

```json
{
  "status": "OK",
  "timestamp": "2024-01-20T10:30:00Z",
  "redis": "connected",
  "redis_version": "7.2.4",
  "redis_uptime_seconds": 3600,
  "redis_connected_clients": 2,
  "redis_used_memory_human": "1.2M"
}
```

🔍 Debugging Common Issues

Redis Connection Refused

```bash
```

```
# Check if Redis is running
ps aux | grep redis

# Check Redis logs
# macOS: tail -f /usr/local/var/log/redis.log
# Linux: sudo tail -f /var/log/redis/redis-server.log

# Try connecting manually
redis-cli -h localhost -p 6379 ping
```

Permission Denied

```bash
# Linux: Check Redis service permissions
sudo chown redis:redis /var/lib/redis
sudo chmod 770 /var/lib/redis
```

Memory Issues

```bash
# Check Redis memory config
redis-cli config get maxmemory

# Set memory limit (e.g., 256MB)
redis-cli config set maxmemory 256mb
redis-cli config set maxmemory-policy allkeys-lru
```

🚦 Production Considerations

1. Use Redis Cloud Services

- **Redis Cloud**: https://redis.com/try-free/
- **AWS ElastiCache**: https://aws.amazon.com/elasticache/
- **Azure Cache for Redis**: https://azure.microsoft.com/services/cache/
- **Upstash**: https://upstash.com/ (Serverless Redis)

2. Connection URL Examples

```bash
```

```
# Local
REDIS_URL=redis://localhost:6379/0

# With password
REDIS_URL=redis://:password@localhost:6379/0

# Redis Cloud
REDIS_URL=redis://default:password@redis-12345.c1.us-east-1.ec2.cloud.redislabs.com:12345

# With SSL/TLS
REDIS_URL=rediss://default:password@redis-12345.c1.us-east-1.ec2.cloud.redislabs.com:12345
REDIS_SSL=true
```

3. Enable Persistence

```bash
# Edit redis.conf
# Enable RDB snapshots
save 900 1
save 300 10
save 60 10000

# Enable AOF (Append Only File)
appendonly yes
```

📝 Quick Commands Cheat Sheet

```bash
```

```
# Start Redis
redis-server              # Foreground
redis-server --daemonize yes  # Background

# Connect to Redis
redis-cli               # Local
redis-cli -h host -p port -a password  # Remote

# Basic Commands
SET key value            # Set a key
GET key                # Get a key
EXISTS key              # Check if key exists
DEL key                # Delete a key
TTL key                # Check time to live
KEYS pattern             # Find keys (use SCAN in production)

# Monitor
MONITOR               # Watch all commands
INFO               # Server information
CLIENT LIST             # Connected clients

# Cleanup
FLUSHDB               # Clear current database
FLUSHALL              # Clear all databases (careful!)
```

✅ You're Ready!

Once Redis is running and your app connects successfully, you'll have:

- ✅ Persistent refresh tokens that survive server restarts
- ✅ Distributed rate limiting
- ✅ Centralized auth state management
- ✅ Better scalability for multiple app instances

Need help? Check the health endpoint at `http://localhost:8000/health` to see Redis connection status and details.