

Access Instructions for Local Testing

Overview

This document provides all the necessary URLs and access instructions for testing the Heart Disease Prediction MLOps application locally.

Quick Start

Prerequisites

Before accessing any service, ensure you have:

1. Completed the setup from [EXECUTION_GUIDE.md](#)
 2. Started the required services (see commands below)
-

Service URLs and Access Points

1. FastAPI Application (Main API)

Base URL: <http://localhost:8000>

API Documentation (Swagger UI): <http://localhost:8000/docs>

Interactive API Testing: <http://localhost:8000/redoc>

Endpoints:

- Health Check: <http://localhost:8000/health>
 - Single Prediction: <http://localhost:8000/predict>
 - Batch Prediction: <http://localhost:8000/predict/batch>
 - Metrics: <http://localhost:8000/metrics>
-

2. MLflow Tracking UI

URL: <http://localhost:5000>

Purpose: View experiment tracking, model metrics, parameters, and artifacts

No authentication required

3. Prometheus Monitoring

URL: <http://localhost:9090>

Purpose: Monitor API metrics, query performance data

No authentication required

4. Grafana Dashboard

URL: <http://localhost:3000>

Default Credentials:

- **Username:** admin
- **Password:** admin

Purpose: Visualize metrics and create custom dashboards

🧪 Testing the API

Method 1: Using Swagger UI

1. Open your browser and navigate to: <http://localhost:8000/docs>
 2. Click on any endpoint (e.g., [/predict](#))
 3. Click "Try it out"
 4. Enter the input data
 5. Click "Execute"
 6. View the response
-

Method 2: Using cURL Commands

Health Check

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

Expected Response:

```
{
  "status": "healthy",
  "model_loaded": true,
  "preprocessor_loaded": true
}
```

Single Prediction

```
curl -X POST "http://localhost:8000/predict" \
-H "Content-Type: application/json" \
-d '{'
```

```
"age": 63,  
"sex": 1,  
"cp": 3,  
"trestbps": 145,  
"chol": 233,  
"fbs": 1,  
"restecg": 0,  
"thalach": 150,  
"exang": 0,  
"oldpeak": 2.3,  
"slope": 0,  
"ca": 0,  
"thal": 1  
}'
```

Expected Response:

```
{  
    "prediction": 1,  
    "probability": 0.85,  
    "risk_level": "high",  
    "confidence": 0.85  
}
```

Batch Prediction

```
curl -X POST "http://localhost:8000/predict/batch" \  
-H "Content-Type: application/json" \  
-d '{  
    "instances": [  
        {  
            "age": 63,  
            "sex": 1,  
            "cp": 3,  
            "trestbps": 145,  
            "chol": 233,  
            "fbs": 1,  
            "restecg": 0,  
            "thalach": 150,  
            "exang": 0,  
            "oldpeak": 2.3,  
            "slope": 0,  
            "ca": 0,  
            "thal": 1  
        },  
        {  
            "age": 45,  
            "sex": 0,
```

```
        "cp": 1,
        "trestbps": 120,
        "chol": 200,
        "fbs": 0,
        "restecg": 0,
        "thalach": 170,
        "exang": 0,
        "oldpeak": 0.5,
        "slope": 1,
        "ca": 0,
        "thal": 2
    }
]
}'
```

Method 3: Using Python Requests

```
import requests
import json

# API endpoint
url = "http://localhost:8000/predict"

# Sample patient data
data = {
    "age": 63,
    "sex": 1,
    "cp": 3,
    "trestbps": 145,
    "chol": 233,
    "fbs": 1,
    "restecg": 0,
    "thalach": 150,
    "exang": 0,
    "oldpeak": 2.3,
    "slope": 0,
    "ca": 0,
    "thal": 1
}

# Make prediction request
response = requests.post(url, json=data)

# Print results
print("Status Code:", response.status_code)
print("Response:", json.dumps(response.json(), indent=2))
```

Method 4: Using the Sample Input File

```
# Use the provided sample_input.json file
curl -X POST "http://localhost:8000/predict" \
-H "Content-Type: application/json" \
-d @sample_input.json
```

🌐 Starting Services

Option 1: Direct Python API

```
# Navigate to project directory
cd heart-disease-mlops

# Activate virtual environment
source venv/bin/activate

# Start FastAPI server
uvicorn src.app:app --reload --host 0.0.0.0 --port 8000
```

Access: API will be available at <http://localhost:8000>

Option 2: Using Docker

```
# Build Docker image
docker build -t heart-disease-api .

# Run container
docker run -d -p 8000:8000 --name heart-api heart-disease-api

# View logs
docker logs -f heart-api
```

Access: API will be available at <http://localhost:8000>

Option 3: Full Stack with Docker Compose (Includes Monitoring)

```
# Start all services
docker-compose up -d

# View logs
docker-compose logs -f
```

```
# Stop all services
docker-compose down
```

Services Available:

- **API:** <http://localhost:8000>
- **Prometheus:** <http://localhost:9090>
- **Grafana:** <http://localhost:3000>

Starting MLflow UI

```
# In a new terminal
cd heart-disease-mlops
source venv/bin/activate

# Start MLflow UI
mlflow ui --port 5000
```

Access: <http://localhost:5000>

Verification Checklist

Use this checklist to verify all services are running correctly:

- **API Health Check:** `curl http://localhost:8000/health` returns status "healthy"
- **API Documentation:** <http://localhost:8000/docs> loads successfully
- **Prediction Works:** Test prediction endpoint returns valid JSON response
- **MLflow UI:** <http://localhost:5000> shows experiments (if started)
- **Prometheus:** <http://localhost:9090> loads dashboard (if using docker-compose)
- **Grafana:** <http://localhost:3000> accessible with admin/admin (if using docker-compose)

Feature Attributes (For Testing)

When testing predictions, use these attribute ranges:

Feature	Description	Range	Type
age	Age in years	29-77	Integer
sex	Gender	0=Female, 1=Male	Integer
cp	Chest pain type	0-3	Integer
trestbps	Resting blood pressure	94-200	Integer
chol	Serum cholesterol (mg/dl)	126-564	Integer

Feature	Description	Range	Type
fbs	Fasting blood sugar > 120 mg/dl	0=No, 1=Yes	Integer
restecg	Resting ECG results	0-2	Integer
thalach	Maximum heart rate	71-202	Integer
exang	Exercise induced angina	0=No, 1=Yes	Integer
oldpeak	ST depression	0.0-6.2	Float
slope	Slope of peak exercise ST	0-2	Integer
ca	Number of major vessels	0-3	Integer
thal	Thalassemia	0-3	Integer