

MCP Integration Learning Plan

Learning Objectives

By Week 6, you'll master:

- **MCP Server Architecture** - How MCP servers work under the hood
- **Using Existing MCP Servers** - Leverage community-built tools
- **Building Custom MCP Servers** - Create domain-specific integrations
- **Claude + MCP Integration** - Connect everything seamlessly
- **Production Deployment** - Deploy MCP servers alongside your app

MCP Fundamentals (Quick Primer)

What is MCP?

Model Context Protocol - A standardized way for AI models to interact with external tools and data sources.

Architecture Pattern:


Claude \longleftrightarrow MCP Client \longleftrightarrow MCP Server \longleftrightarrow External APIs/Databases

Key Concepts:

- **Tools**: Functions Claude can call (e.g., `get_btc_price()`)
- **Resources**: Data Claude can read (e.g., files, database records)
- **Prompts**: Pre-defined prompt templates
- **Sampling**: AI-generated content based on context

Integrated Timeline (Updated Schedule)

Week 1: Foundation

- Day 1:  Project structure (complete)
- Day 2-5: Git, React, API basics (as planned)

Week 2: Multi-Agent + MCP Foundations

- **Day 1**: LangGraph setup + **MCP ecosystem overview**

- **Day 2:** Market Data Agent + **File System MCP integration**
- **Day 3:** News Agent + **explore existing MCP servers**
- **Day 4:** Risk Agent + **plan custom Crypto Data MCP server**
- **Day 5:** Agent orchestration + **MCP server development start**

Week 3: Custom MCP Server Development

- **Day 1: Build Crypto Data MCP Server** + Claude integration
- **Day 2:** Analyst Agent + **MCP server testing and refinement**
- **Day 3:** News Intelligence + **Firebase MCP server integration**
- **Day 4:** Presenter Agent + **MCP server deployment**
- **Day 5:** Firebase Integration + **full MCP ecosystem testing**

Week 4-6: Advanced Integration (as planned, but MCP-enhanced)

MCP Servers We'll Build & Use

1. Existing MCP Servers (Week 2)

File System MCP Server

```
typescript
// Available in MCP ecosystem
{
  "name": "filesystem",
  "tools": [
    "read_file",
    "write_file",
    "list_directory",
    "create_directory"
  ]
}
```

Learning Value: Understanding MCP client integration **Time Investment:** 30 minutes setup **Benefit:** I can directly manage your project files

SQLite MCP Server

```
typescript
```

```
// For local development data
```

```
{  
  "name": "sqlite",  
  "tools": [  
    "query",  
    "execute",  
    "list_tables"  
  ]  
}
```

Learning Value: Database MCP patterns **Time Investment:** 20 minutes setup **Benefit:** Local progress tracking and experimentation

2. Custom Crypto Data MCP Server (Week 3)

Server Architecture:

```
python
```

```

# crypto_mcp_server/server.py
from mcp import Server, Tool
import asyncio
import requests

class CryptoMCPServer:
    def __init__(self):
        self.server = Server("crypto-data")
        self.setup_tools()

    def setup_tools(self):
        @self.server.tool()
        async def get_portfolio_snapshot(tokens: list[str]) -> dict:
            """Get current prices for portfolio tokens"""
            # CoinGecko API integration
            return await self.fetch_token_prices(tokens)

        @self.server.tool()
        async def get_fear_greed_index() -> dict:
            """Current crypto Fear & Greed Index"""
            # Alternative.me API integration
            return await self.fetch_fear_greed()

        @self.server.tool()
        async def get_trending_narratives() -> dict:
            """Top DeFi narratives by TVL growth"""
            # DeFiLlama API integration
            return await self.fetch_defillama_trends()

```

Learning Value: Custom MCP server development from scratch **Time Investment:** 2 hours over 2 days

Benefit: Real-time market data in our conversations

3. Firebase MCP Server (Week 3)

python

```
# firebase_mcp_server/server.py
class FirebaseMCPServer:
    @self.server.tool()
    async def save_briefing(content: str, date: str) -> dict:
        """Save daily briefing to Firestore"""

    @self.server.tool()
    async def get_briefing_history(days: int = 7) -> dict:
        """Retrieve past briefings for context"""

    @self.server.tool()
    async def update_user_preferences(preferences: dict) -> dict:
        """Update user investment profile"""
```

Learning Value: Cloud database MCP integration **Time Investment:** 1 hour **Benefit:** Persistent memory and context

Learning Progression

Week 2: MCP Consumer (Using Existing Servers)

Skills Gained:

- MCP client configuration
- Tool calling patterns
- Resource management
- Error handling

Hands-on Experience:

- Connect File System MCP server to our project
- Use Claude with MCP tools during development
- Debug MCP connection issues

Week 3: MCP Producer (Building Custom Servers)

Skills Gained:

- MCP server architecture
- Async tool development
- API integration patterns

- Server deployment

Hands-on Experience:

- Build complete Crypto Data MCP server
- Test server locally and in production
- Handle rate limiting and error cases

Week 4-6: MCP Integration Mastery

Skills Gained:

- Multi-server orchestration
- Performance optimization
- Production deployment
- Monitoring and logging

Unique Learning Outcomes

Technical Skills

- **MCP Protocol Mastery** - Understanding of cutting-edge AI integration patterns
- **Server Development** - Building production-ready MCP servers
- **API Integration** - Professional patterns for external data sources
- **Async Python** - Modern async/await patterns

Portfolio Value

- **Rare Expertise** - Very few developers have MCP experience yet
- **Full Stack AI** - From model to data integration
- **Production Ready** - Deployed, working MCP servers
- **Open Source Contribution** - Your crypto MCP server could help others

Updated Investment

Time Investment

- **Original Plan:** ~30 hours over 6 weeks
- **With MCP Integration:** ~36 hours over 6 weeks
- **Additional Learning:** 6 hours for cutting-edge MCP skills

Value Multiplier

- **Technical Skills:** 3x more valuable (MCP + Multi-agent + React)
- **Portfolio Strength:** 2x more impressive (fewer people have this combo)
- **Future Opportunities:** MCP skills will be highly sought after



Week 2 MCP Preview

When we start Week 2, here's what we'll build:

Day 1: MCP Setup

```
bash

# Install MCP tools
pip install mcp-server-filesystem
npm install @modelcontextprotocol/client

# Configure Claude to use MCP servers
# I'll guide you through the exact setup
```

Day 2: First MCP Integration

```
python

# I'll be able to do this during our conversation:
await write_file("backend/agents/market_data_agent.py", agent_code)
await read_file("progress_tracker.csv")
# See your actual progress in real-time!
```

Day 3-5: Custom Server Development

- Build crypto data MCP server together
- Test with real API calls
- Deploy alongside your main application



Ready to Level Up?

This plan gives you:

- **Core multi-agent skills** (original plan)
- **Advanced MCP integration** (cutting-edge addition)

- **Production deployment experience** (both systems together)
- **Rare, valuable expertise** (very few developers have this yet)

The additional 6 hours will make you one of the first developers with production MCP + multi-agent experience. That's consulting-level expertise!

Sound like a worthwhile investment? We'll start with MCP foundations in Week 2, Day 1.