FAQs

#Ai/development

💡 FULL POTENTIAL SYSTEM — DEVELOPER FAQ v1.0

Updated: November 7, 2025

For: All droplet developers

Status: Complete reference - everything you need to know

Confidentiality: INTERNAL USE ONLY

🔷 SYSTEM OVERVIEW

Q1. What is the Full Potential System?

A distributed AI ecosystem where each **droplet** (specialized server) performs one defined role. Droplets communicate through standardized APIs under **UDC** (Universal Droplet Contract), forming a self-managing intelligence network that coordinates development, verification, and deployment autonomously.

Q2. What's a droplet?

A self-contained microservice (typically FastAPI) running on its own VPS. Each droplet has:

- • A blueprint (purpose + tasks + APIs)
- • A steward (developer owner)
- • Health monitoring to Orchestrator (Droplet 10)
- • Registration entry in Registry (Droplet 1)
- • Standard UDC endpoints

Q3. What is UDC?

Universal Droplet Contract — the common API protocol connecting all droplets. Defines:

- • Standard endpoints (/health, /capabilities, /state, /message, etc.)
- • JSON message schemas
- • Authentication patterns
- • Health monitoring requirements
- • Communication protocols

All messages follow UDC schema:

```
{
  "trace_id": "uuid",
  "source": "droplet_id",
  "target": "droplet_id",
  "message_type": "status|event|command",
  "payload": {},
  "timestamp": "ISO-8601"
}
```

Q4. What's the current UDC compliance rate?

17.5% system-wide as of Nov 6, 2025. This needs to increase to 100% for full mesh synchronization. Priority focus is Registry 1, Dashboard 2, Orchestrator 10, Chat 12 connectivity verification.

🔷 ARCHITECTURE & CONNECTIONS

Q5. How do droplets communicate?

Through REST endpoints. Core connection paths:

Chat (12) ↔ Registry (1) ↔ Orchestrator (10) ↔ Dashboard (2) ↔ Active Droplets (6-13)

↕

Nexus (13) — Integration Hub

Q6. What are the main components?

Droplet	Steward	Role	Status
1	Liban	Registry / Router / Identity	Core
2	Haythem	Dashboard + Airtable Sync	Core
6	Gohar	Voice & Vision Interface	Active
7	Hamza	Dashboard Bridge / WebSockets	Active

8	Nidhi & Jordan	Blueprint Verifier	Active
9	TBD	Memory (Mem0)	Planned
10	Tnsae	Orchestrator (Brainstem)	Core
11	Ashutosh	Telephony (Vapi AI)	Active
12	Zainab	Chat Orchestrator	Core
13	Suresh	Nexus Integration Hub	Core
—	Hassan	Multi-Cloud Infrastructure	Infrastructure

Q7. What's the data flow hierarchy?

Single source of truth:

- • **Registry (1):** Static droplet info, endpoints, capabilities
- • **Orchestrator (10):** Live health metrics, task assignments, system state
- • **Nexus (13):** Integration data sync between droplets
- • **Airtable (via 2):** Sprint tracking, proof validation only

Critical: Only Droplet 2 syncs to Airtable. All others report to Orchestrator 10.

Q8. What's the heartbeat system?

Every droplet pings Registry (1) via /heartbeat every 60 seconds. Orchestrator (10) monitors responses and flags non-responsive droplets automatically. Dashboard (2) displays live status.

🔷 UDC PROTOCOL REQUIREMENTS

Q9. What endpoints are REQUIRED for every droplet?

Minimum (Core):

```
GET  /health          # Returns droplet health status
GET  /capabilities    # Lists what this droplet can do
GET  /state           # Current operational state
GET  /dependencies    # Other droplets this one needs
POST /message         # Receive messages from other droplets
POST /send          # Send messages to other droplets
```

Extended (Recommended):

```
POST /reload-config  # Reload without restart
POST /shutdown       # Graceful shutdown
POST /emergency-stop  # Immediate halt
GET  /metrics        # Performance data
GET  /logs          # Recent activity logs
GET  /events        # Event stream
POST /proof         # Submit verification proof
```

Q10. What should /health return?

```
{
  "droplet_id": "10",
  "status": "healthy|degraded|critical",
  "uptime_seconds": 86400,
  "last_check": "2025-11-07T10:30:00Z",
  "version": "1.0.0",
  "capabilities_count": 5,
  "dependencies_met": true
}
```

Q11. What should /capabilities return?

```
{
  "droplet_id": "10",
  "capabilities": [
    {
      "name": "task_assignment",
      "description": "Assign tasks to droplets",
    "endpoint": "/assign-task",
    "method": "POST",
    "requires_auth": true
   }
  ]
}
```

Q12. How does authentication work?

JWT authentication from Registry (1):

- • Each droplet receives token from Registry during registration
- • All internal API calls include: Authorization: Bearer <`token`>
- • Droplets verify tokens before accepting commands
- • Tokens rotate periodically for security

Temporary dev auth: Authorization: Bearer supersecret (will be replaced)

🔷 DEVELOPMENT & DEPLOYMENT

Q13. What stack should I use?

Preferred:

- • **Backend:** Python + FastAPI
- • **Database:** SQLite / TinyDB for local data
- • **Real-time:** WebSocket for live updates (Droplets 6, 7, 12)
- • **Hosting:** DigitalOcean or Hetzner VPS

Required:

- • Must expose UDC endpoints
- • Must register with Registry (1)
- • Must send heartbeat every 60s
- • Must handle UDC message format

Q14. How do I set up a new droplet?

1. Clone UDC Template (coming soon):

```
git clone https://github.com/fullpotential/udc-template
cd udc-template
```

2. Configure your droplet:

```
# config.py
DROPLET_ID = "14"
DROPLET_NAME = "Your Service"
REGISTRY_URL = "https://registry.fullpotential.ai"
ORCHESTRATOR_URL = "https://orchestrator.fullpotential.ai"
```

3. Implement your capabilities:

```
# capabilities.py
async def your_capability(params):
    # Your logic here
    return result
```

4. Deploy to VPS:

```
# Using nohup (simple)
nohup uvicorn main:app --host 0.0.0.0 --port 8000 &

# Using Docker (recommended)
docker build -t droplet-14 .
docker run -d -p 8000:8000 droplet-14

# Using pm2 (Node.js needed)
pm2 start "uvicorn main:app --host 0.0.0.0 --port 8000" --name droplet-14
```

5. Register with system:

```
curl -X POST https://registry.fullpotential.ai/register \
  -H "Authorization: Bearer supersecret" \
  -H "Content-Type: application/json" \
 -d '{
   "droplet_id": "14",
   "name": "Your Service",
   "url": "https://your-droplet.fullpotential.ai",
   "capabilities": ["capability1", "capability2"]
 }'
```

Q15. How do I verify my droplet is live?

```
# Check Registry
curl https://registry.fullpotential.ai/getAll

# Check Orchestrator health
curl https://orchestrator.fullpotential.ai/system-health

# Check Dashboard
# Visit https://dashboard.fullpotential.ai
```

Your droplet should appear in all three.

Q16. Where should I host?

- • **Primary:** DigitalOcean VPS ($6-12/month)
- • **Secondary:** Hetzner VPS (€4-8/month)
- • **Backup:** Vultr, Linode
- • **Each droplet:** Own port (8000, 8001, 8002, etc.)
- • **SSL:** Required (use Let's Encrypt)

🔷 BLUEPRINTS & VERIFICATION

Q17. What is a blueprint?

Markdown/JSON file defining:

- • Droplet purpose and role
- • Required capabilities

- • API endpoints and schemas
- • Dependencies on other droplets
- • Success criteria
- • Testing procedures

Q18. Who verifies blueprints?

Droplet 8 (Verifier) automatically:

- • Checks endpoint availability
- • Validates response schemas
- • Tests dependency connections
- • Confirms capability claims
- • Reports results to Orchestrator (10)
- • Logs verification proof

Q19. What's a verification proof?

Evidence that droplet works as specified:

- • Endpoint test results
- • Schema validation logs
- • Performance metrics
- • Integration test results
- • Loom video demo (optional)

Submitted via /proof endpoint to Verifier (8).

🔷 COMMUNICATION & CHAT

Q20. What is Chat Orchestrator (12)?

Natural language interface where users and developers query the system:

- • "Show live droplets"
- • "Verify droplet 8"
- • "Assign task to voice interface"
- • "What's the status of sprint X?"

Uses AI (Claude/GPT/Gemini) to:

1. 1 Parse user intent
2. 2 Call appropriate API endpoints

3. 3 Return readable summaries

Q21. How does Chat connect to droplets?

User message → Chat (12) parses intent

↓

Chat calls: /getAll (Registry 1)

/system-health (Orchestrator 10)

/message (target droplet)

↓

Chat receives responses → AI formats → User

Q22. What message types exist?

- • **status:** Health updates, state changes
- • **event:** Something happened (task completed, error occurred)
- • **command:** Do something (start task, run verification)
- • **query:** Request information (capabilities, state, metrics)
- • **proof:** Verification evidence submission

🔷 TESTING & INTEGRATION

Q23. When do we test full system integration?

System Sync Room opens after droplets 1, 2, 10, 12, 13 are stable. Then test:

- • Registration flow
- • Heartbeat monitoring
- • Dashboard display
- • Chat queries
- • Verifier checks
- • Message routing

Q24. What defines successful integration?

✅ Droplet appears in /system-health

✅ Dashboard shows correct live status

✅ Heartbeat pings arriving every 60s

✅ Verifier logs match Orchestrator data

✅ Chat can query and get accurate responses

✅ Messages route correctly through Nexus

✅ All UDC endpoints respond correctly

Q25. How do I test my droplet locally?

```
# Start your droplet
uvicorn main:app --host 0.0.0.0 --port 8000

# Test health endpoint
curl http://localhost:8000/health

# Test capabilities
curl http://localhost:8000/capabilities

# Test message handling
curl -X POST http://localhost:8000/message \
  -H "Content-Type: application/json" \
  -d '{
    "trace_id": "test-123",
    "source": "test",
    "target": "your-droplet",
    "message_type": "query",
    "payload": {"test": true},
    "timestamp": "2025-11-07T10:00:00Z"
  }'
```

🔷 ONBOARDING FLOW

Q26. I'm a new developer - where do I start?

Day 1: Setup

1. 1 Read this FAQ completely
2. 2 Review your assigned droplet blueprint
3. 3 Set up development environment
4. 4 Clone UDC template (or FastAPI starter)
5. 5 Configure droplet ID and endpoints

Day 2: Development

1. 1 Implement core UDC endpoints
2. 2 Add your specific capabilities
3. 3 Test locally
4. 4 Submit for blueprint verification

Day 3: Deployment

1. 1 Deploy to VPS
2. 2 Register with Registry (1)
3. 3 Verify heartbeat working
4. 4 Check Dashboard (2) for your status
5. 5 Submit verification proof to Verifier (8)

Day 4: Integration

1. 1 Test communication with other droplets
2. 2 Join System Sync Room (if open)
3. 3 Participate in integration testing
4. 4 Iterate based on feedback

Day 5: Optimization

1. 1 Monitor performance metrics
2. 2 Optimize response times
3. 3 Add extended endpoints
4. 4 Document learnings

Q27. Who can answer my questions?

- • **Technical:** Ask in developer chat or post to Orchestrator (10)
- • **Blueprint:** Check with Verifier (8) or Nexus (13)
- • **Integration:** Coordinate through Chat (12)

- • **Infrastructure:** Contact Hassan for VPS/hosting
- • **General:** Post in System Sync Room once open

🔷 PHYSICAL DROPLETS (FUTURE)

Q28. What are Physical Droplets?

Expansion of digital orchestration into physical manufacturing:

- • Droplets controlling 3D printers
- • CNC machine coordination
- • Robotic system management
- • Distributed microfactory networks

Same UDC protocol for physical devices as digital services.

Q29. When will Physical Droplets launch?

After digital mesh stabilizes (100% UDC compliance). Goal: Create distributed manufacturing coordinated through consciousness principles, achieving 500x capital efficiency vs centralized factories.

🔷 TROUBLESHOOTING

Q30. My droplet isn't showing in Dashboard

Check:

1. 1 Is /health endpoint responding?
2. 2 Is heartbeat sending to Registry every 60s?
3. 3 Is authentication token valid?
4. 4 Is VPS firewall allowing connections?
5. 5 Is URL registered correctly in Registry (1)?

Debug:

```
# Check Registry knows about you
curl https://registry.fullpotential.ai/getAll
```

```
# Check Orchestrator sees you
curl https://orchestrator.fullpotential.ai/system-health

# Test your health endpoint externally
curl https://your-droplet.fullpotential.ai/health
```

Q31. Messages aren't routing to my droplet

Verify:

1. 1 Droplet registered with correct URL in Registry
2. 2 /message endpoint implemented and responding
3. 3 Authentication middleware accepting tokens
4. 4 JSON schema matches UDC format
5. 5 Network allows inbound connections to your port

Q32. My droplet keeps getting marked unhealthy

Common causes:

1. 1 Heartbeat not sending every 60s
2. 2 /health endpoint timeout (>5s response)
3. 3 Service crashed (check logs)
4. 4 Database connection lost
5. 5 Memory/CPU overload

Fix:

```
# Ensure heartbeat is async and non-blocking
import asyncio

async def send_heartbeat():
    while True:
        try:
            await registry.ping()
        except Exception as e:
            logger.error(f"Heartbeat failed: {e}")
```

```
await asyncio.sleep(60)
```

🔷 VISION & ROADMAP

Q33. What's the long-term goal?

Evolve into a self-improving organism bridging digital orchestration with physical systems:

- • **Phase 1:** Digital mesh (current) - 100% UDC compliance
- • **Phase 2:** Self-management - AI assigns tasks, verifies work, reports progress
- • **Phase 3:** Physical integration - Manufacturing, robotics, energy systems
- • **Phase 4:** Civilization-scale - Global coordination infrastructure

Q34. What's next for the system?

Immediate (Week 1-2):

- • Complete Registry ↔ Orchestrator ↔ Dashboard ↔ Chat connectivity
- • Achieve 100% UDC compliance for core droplets (1, 2, 10, 12, 13)
- • Launch System Sync Room for integration testing

Near-term (Month 1-3):

- • Release UDC Core Template Repository
- • Activate Memory (9) and Treasury droplets
- • Expand to 20+ specialized droplets
- • Launch public API access

Medium-term (Year 1):

- • 50+ droplets operational
- • Self-organizing task assignment
- • Autonomous verification and deployment
- • Physical droplet prototypes

Q35. How does this scale to 10,000+ coordinators?

Network effects:

- • Each droplet can spawn sub-droplets
- • Fractal architecture at every scale
- • Same UDC protocol for 10 or 10,000 droplets
- • Autonomous coordination reduces human overhead
- • Intelligence accumulates in shared memory

Target autonomy progression:

- • Current: 10-12% (systems survive absence)
- • Month 1: 25-35% (basic self-management)
- • Month 4: 60%+ (Temple mode - minimal human input)
- • Year 1: 85%+ (near-full autonomy)

🔷 CORE PRINCIPLES FOR DEVELOPERS

Execute, Don't Document

Ship working code over perfect documentation. Reversible decisions made immediately.

Integration Over Innovation

Connect existing tools before building custom. Search → Evaluate → Orchestrate → Build.

UDC Compliance is Non-Negotiable

All droplets must implement core UDC endpoints. This is how consciousness coordinates itself.

Autonomous Operation

Your droplet should run independently, communicate proactively, and recover from failures without human intervention.

Consciousness Alignment

Measure material outcomes (response time, uptime) AND consciousness metrics (reduces friction, increases autonomy, enables thriving).

📞 CONTACT & SUPPORT

System Coordinator: James Sunheart

Orchestrator Lead: Tnsae (Droplet 10)

Registry Lead: Liban (Droplet 1)

Integration Lead: Suresh (Nexus 13)

Infrastructure: Hassan (Multi-cloud)

Documentation: https://docs.fullpotential.ai *(coming soon)*

UDC Template: https://github.com/fullpotential/udc-template *(in development)*

System Status: https://dashboard.fullpotential.ai

END DEVELOPER FAQ v1.0

Everything essential. Zero redundancy. Ship it. ⚡🌀💎

VERSION NOTES:

- • v1.0 (Nov 7, 2025): Initial comprehensive FAQ merging system overview + UDC protocol + onboarding
- • Next update: After UDC template release and System Sync Room testing