

---

## E EXAMPLES FROM DIFFERENT COMPONENTS OF BEAM

In this section, we provide illustrative examples of generating a chat in the *coding* domain. Specifically, we include a representative *chat seed* with its domain, title, theme, and subtopics, followed by the corresponding *narratives*, where only a truncated set is shown for brevity. We then present the *user profile* and the user's social *relationships*. Next, we provide excerpts from the *conversation plans*, showing only a subset of bullet points from each sub-plan while preserving their full descriptions to maintain clarity. Finally, we provide samples of the *generated chat*, highlighting exchanges where the user shares or requests code, and including follow-up turns to demonstrate the naturalistic back-and-forth flow. Together, these examples illustrate how different components of BEAM interact to form coherent, long-context dialogues.

### Chat Seed

**Domain:** Coding

**Title:** Automating Social Media Posts with Python

**Theme:** Scheduling and posting content across multiple platforms

**Subtopics:**

- Twitter API integration · Facebook Graph API usage · Instagram automation tools
- Scheduling with cron jobs / APScheduler
- Image and caption management; hashtag generation
- Error handling for failed posts; tracking engagement metrics

### Narratives (Truncated)

**Technical Problem-Solving:** Debugging Twitter OAuth/403/429; fixing hashtag validation; profiling scheduler bottlenecks.

**Learning & Knowledge:** API docs comprehension (Twitter v2, Facebook Graph v12–15); best practices for Instagram automation; mastering cron/APScheduler.

**Progress & Development:** Setting up Twitter/Facebook integrations; building Instagram tools; designing scheduling algorithms.

**Implementation:** Feature implementation and refactoring for efficiency; async migration; retry and backoff strategies.

**Framework & Technology:** Python libraries (Tweepy, facebook-sdk, requests); APScheduler/cron; Redis; asyncio.

**Testing & QA:** Unit/integration/E2E tests (pytest, Selenium); TDD for schedulers and hashtag rules.

**DevOps & Deployment:** CI/CD (GitHub Actions), containerization (Docker), EC2 deployment, blue–green releases.

**Data:** PostgreSQL schemas, indices, ETL for engagement metrics, Redis caching.

**Integration & APIs:** Webhooks, message queues (RabbitMQ), API Gateway, SNS/Lambda.

**Performance:** Caching, load balancing (HAProxy), CPU/memory targets, throughput goals.

**Security/Compliance:** OAuth, token rotation, TLS, GDPR.

**PM & Workflow:** Sprints, reviews, documentation standards.

### User Profile

**Name:** John Brooks   **Age:** 52   **Gender:** Male

**Location:** Port Charles, Luxembourg   **Profession:** Secretary/Administrator

**Personality:** He is a pillar of his community, always ready to lend a helping hand and offer guidance when needed. With a strong sense of tradition and order, he values honesty and dedication, often taking on a mentorship role to help others. His diligent and efficient approach to planning and organization makes him a reliable asset to those around him. He has a warm and welcoming demeanor, always willing to open his heart and home to friends, loved ones, and neighbors. Despite his strong convictions, he believes in the power of hospitality and good manners, often going out of his way to make others feel supported and cared for.