Building Gen-AI Pipelines Fast with Firebase Genkit 🔥

Docs: https://firebase.google.com/docs/genkit (or scan QR code below)

Talk by: Aan Patel (aanpatel.tech)

1. Introduction

- **Genkit**: A framework to build gen AI-powered applications and features.
- **Languages**: Node.js and Go.
- **Deployment**: Anywhere Node.js or Go is supported.
- **Vendor-independent**: Works without Google services.
- **Extensible**: Supports custom models and tools via its "plugin" system.
- **Wide range of support for model vendors**: Plugins exist to interface with models from Anthropic, OpenAI, Llama, Cohere, Google, and more. Also supports ollama.

1. Introduction

1.1 Genkit Motivation

- Enable developers to develop, test, train, deploy, and monitor AI models in a coherent and efficient manner. Includes:
    - Managing "prompts as code" - written, versioned, and tested like code.

1. Introduction

1.1 Genkit Motivation

- Enable developers to develop, test, train, deploy, and monitor AI models in a coherent and efficient manner. Includes:
  - Supporting wide range of AI models with a unified API

1. Introduction

1.1 Genkit Motivation

- Enable developers to develop, test, train, deploy, and monitor AI models in a coherent and efficient manner. Includes:
    - Structured output, tool calling, multimodal input/output, multi-agent support

1. Introduction

1.1 Genkit Motivation

- Enable developers to develop, test, train, deploy, and monitor AI models in a coherent and efficient manner. Includes:
    - First-class focus on observability and monitoring

2. Key Capabilities

2.1 Unified API for AI Generation

- Single API for various AI models.
- Supports multimodal input **AND** output (e.g., provide image or video input, or models that generate images or other media).
- Custom model settings with type-safe options.

2. Key Capabilities

2.2 Structured Output

- Generate/stream structured objects (e.g., JSON).
- Built-in validation.
- Simplifies app integration.

2. Key Capabilities

2.3 Tool Calling

- AI models can call functions/APIs as tools.
- Model decides when and which tools to use.

2. Key Capabilities

2.4 Chat API

- Facilitates multi-turn conversations with AI models.
- Supports stateful and persistent chats.

2. Key Capabilities

2.5 Intelligent Agents

- Create agents to automate complex tasks.
- Agents can use tools and other agents -- first-class support for multi-agent systems.

2. Key Capabilities

2.6 Data Retrieval

- Integrate your data to improve output accuracy.
- Simple APIs for embedding, indexing, and retrieving information.

2. Key Capabilities

2.7 Prompt Templating

- Create effective prompts with rich text templating.
- Supports model settings, multimodal input, and tool integration.

3. Flows and Dev Tooling

3. Flows and Dev Tooling

3.1 Defining AI workflows

- Generative model requests are the core of AI features.
- **BUT**, Pre- and post-processing steps are often required.
- Flows in Genkit are pretty much functions but with:
    - Added observability.
    - Simplified deployment.

3. Flows and Dev Tooling

3.2 Possible steps before/after generative model calls

- **Retrieving** contextual information for model calls.
- **Retrieving** user session **chat history** - always needed in chat apps.
- **Reformatting** user input for another model.
- Evaluating the **safety** of model output.
- **Combining outputs** of several models.

3. Flows and Dev Tooling

3.3 Importance of Flows

- Every step must work together for AI tasks to succeed.
- Ability to monitor, test, tweak, and debug the entire process becomes crucial.

3. Flows and Dev Tooling

3.4 Flows in Genkit

- Represent tightly-linked logic using flows.
- Written like functions using ordinary TypeScript code.
- Add capabilities to ease AI feature development.

3. Flows and Dev Tooling

3.5 Flow Capabilities

- **Type safety**: Input and output schemas defined using Zod.
- **Developer UI integration**: Debug flows independently.
- **Simplified deployment**: Deploy flows as web API endpoints.

3. Flows and Dev Tooling

3.6 Advantages over other abstractions

- Lightweight and unobtrusive.
- Logic written in standard TypeScript.
- Code inside a flow doesn't need to be flow-aware.

3. Flows and Dev Tooling

3.7 Genkit Developer Tooling

- **DotPrompt**: Getting the model, model parameters, and prompt working together to produce the output you want involves substantial iteration and experimentation.
  - Genkit provides a **library** and **file format** called **DotPrompt**, that aims to make this iteration faster and more convenient.
- **Genkit CLI**: Create, test, and deploy Genkit flows and prompts.
- **Genkit Web UI**: Visualize and debug flows.

4. Demo: a Node.js API with Genkit

- **Setup**: Install Genkit for Node.js.
- **Example**: Basic usage and integration, and how to use Genkit developer tooling.
- **Deployment**: Ways to deploy your Genkit flows (see docs)

5. Conclusion

- **Genkit**: Powerful framework for AI applications.
- **Flexibility**: Works with various AI models and data sources.
- **Independence**: No dependency on/tight coupling with Google services, fully open-source.

6. Q&A

- Open floor for questions and discussions.

7. Thank you!

- Contact: aanpatel.tech@gmail.com
- GitHub: https://github.com/aannirajpatel