DialogFlow CX Flows

Allen Sanders

Visit Now www.netcomlearning.com

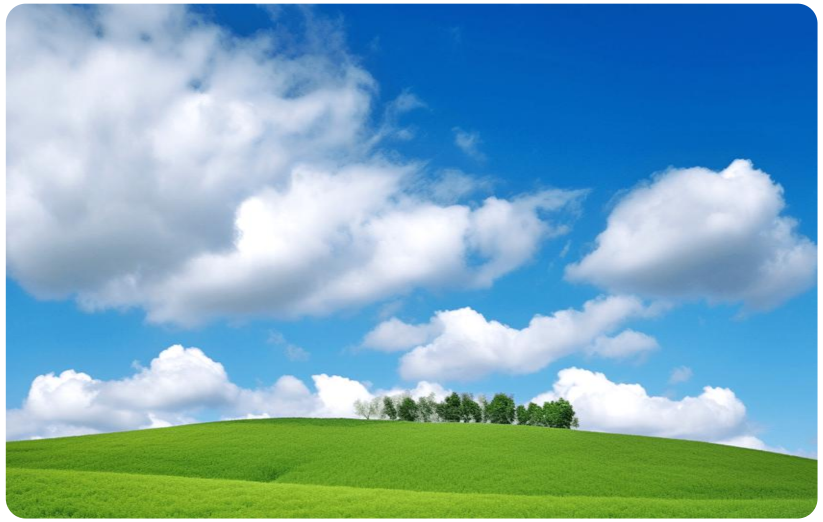
AGENDA

01

Purpose

Definition & Importance

Role in Conversational Design



- Serve as the backbone of conversational design
- Help guide the interaction between users and bots
- Promote structuring of dialogues logically to enhance understanding and user satisfaction

Enhancing User Experience

- Well-designed flows improve user experience
- Help to ensure smooth transitions between topics
- Allow for natural, intuitive interactions that make users feel understood and valued

Definition of Flows

01

Flow Structure

The flow structure refers to the overall layout and organization of a flow, detailing how different components interact and connect. Understanding this structure is critical for effective implementation.

02

Types of Flows

Types of flows can vary by purpose, such as user flows, data flows, or interaction flows. Each type serves a unique function in guiding users or processes through a system effectively.

Core Elements of a Flow

Start State (Entry Point)
The initial user interaction point

Intents & Pages
Define conversation goals and manage transitions

Transition Rules
Control flow progression based on user input

Parameters & Contexts
Store and manage session data for continuity

End States
Conclude the conversation or transition to another flow

Types of Flows – User Flows (Interaction-Based)

01

Core Functions

- Guides the user through structured dialogue
- Handles user intent recognition, response generation, and transitions

02

Example

A customer support chatbot helping users navigate troubleshooting steps

Types of Flows – Data Flows (Backend-Oriented)

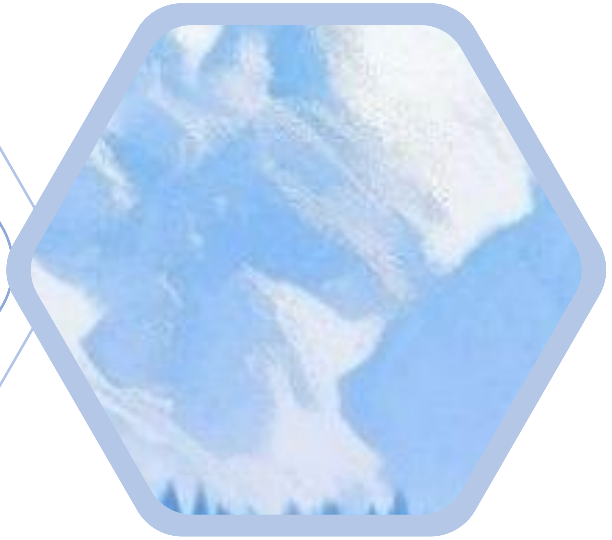
Core Functions

- Processes backend operations, such as API calls and database interactions
- Handles data retrieval, storage, and computation

Example

Order status retrieval using a REST API to fetch data from an external system

Types of Flows – Interaction Flows (Multi-Step User Engagement)



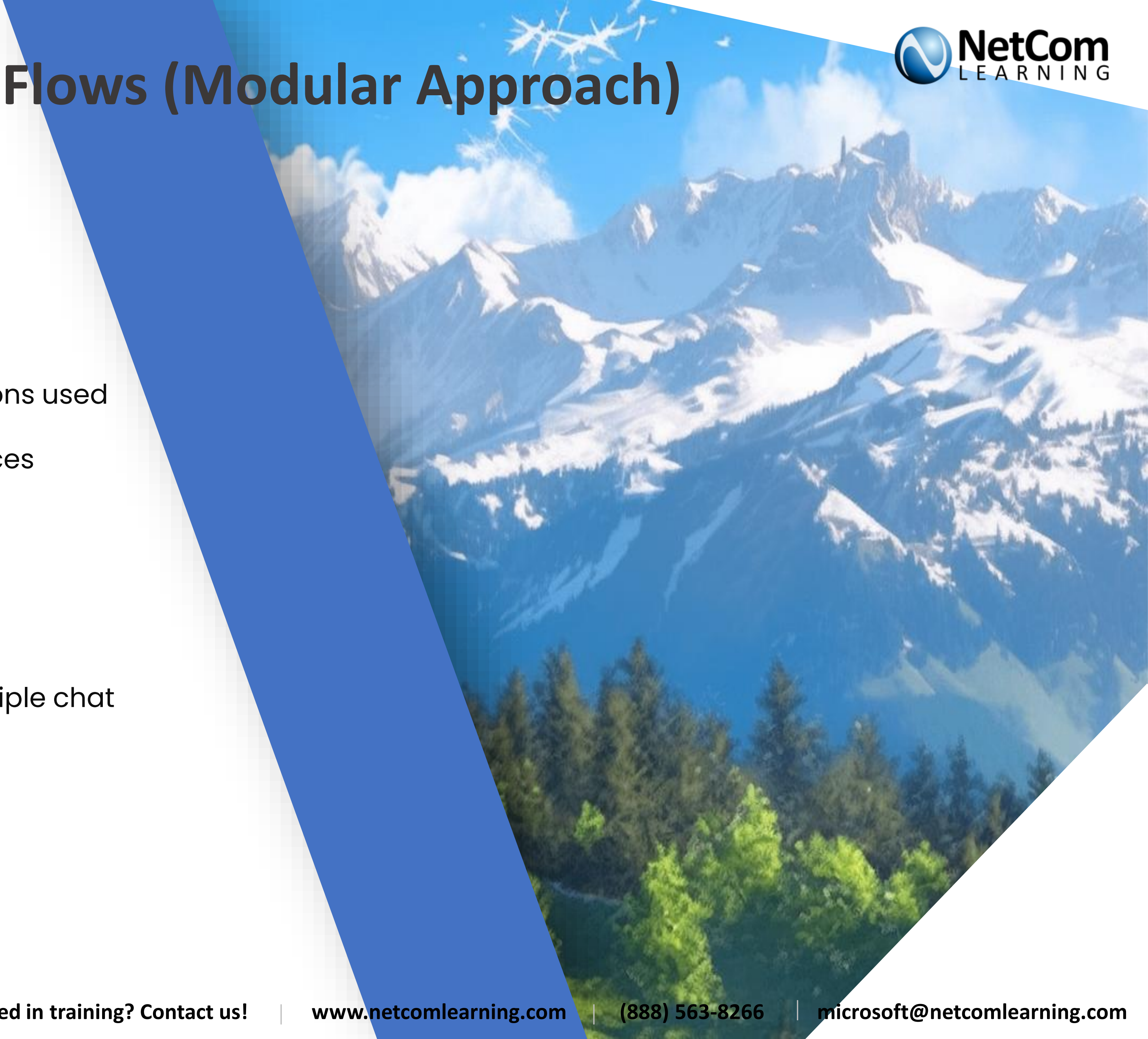
Core Functions

- Manages complex, multi-step conversations across different topics
- Can branch into sub-flows based on user input

Example

A banking assistant helping with both balance inquiries and fund transfers

Types of Flows – Reusable Flows (Modular Approach)



Core Functions

- Created for common interactions used across multiple flows
- Increases scalability and reduces redundancy

Example

User authentication flow that multiple chat functions can call

Types of Flows – System Flows (Predefined Logic)

01 Core Functions

- Manages system- level actions like handling errors, fallbacks, or session expiration
- Ensures robustness and better error handling

02 Example

Fallback flow to guide users when their input isn't understood

Choosing the Right Flow Type

Criteria

- Simple Conversations? → Use User Flows
- Need Backend Data? → Implement Data Flows
- Multi- Step Interaction? → Use Interaction Flows
- Frequent Use Cases? → Create Reusable Flows
- Handling Errors? → Define System Flows

02

Creating New Flows

Flow Creation Process

01

Accessing the Flows Section

To begin creating a flow, navigate to the Flows section in the DialogFlow CX console. This area is dedicated to organizing and managing all flow elements within your project.

02

Naming your Flow

Choose a clear and descriptive name for your flow to ensure easy identification and navigation. A well-named flow contributes to better project management and understanding of its purpose.

Defining Flow States

Entry Points

- Specific locations within your flow from which users can start interacting with your bot
- Define them carefully to guide users effectively through the conversation

Transitions

- Dictate how conversation moves from one state to another
- Clearly outline these transitions to create a seamless user experience
- Drive for cohesive dialogue flow throughout the interaction

Planning – Understanding Requirements

Determine the goals of the conversational agent

Identify who the end- users will be

Establish the scope of the project and desired outcomes

Identify Objectives

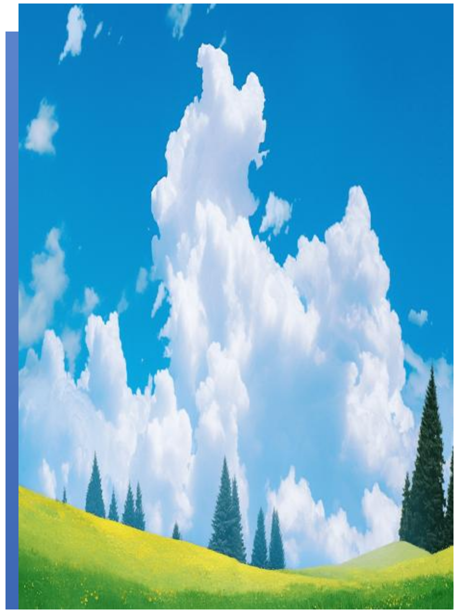
Planning – User Journey Mapping

User Journey Mapping

Outline different paths users might take within the conversation

Identify various intents users might express

Map out potential user responses and agent replies



Development – Creating Intents

Defining User Intents

Create intents that represent user goals and questions

Add training phrases to each intent to help identify user input accurately

Use example sentences that cover a wide range of possible user expressions

Development – Building Entities

Creating and Managing Entities

Define entities that categorize important pieces of information within user input

Use system entities for common data types like dates and numbers or create custom entities

Setup entity synonyms to improve user input recognition

Implementation – Developing Flows

Structuring the Dialogs

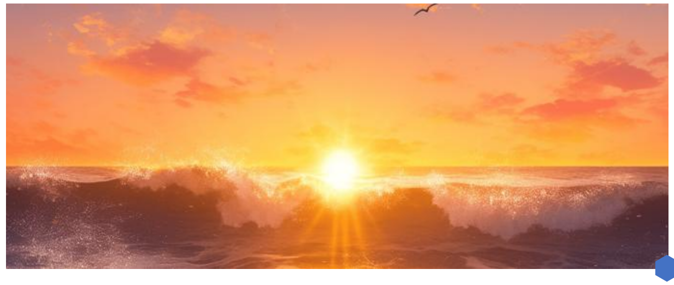
- Divide the conversation into manageable flows and pages
- Create entry points, transitions, and pathways within the flows
- Ensure logical progression and easy navigation for users

Implementation – Adding Fulfillment

Integrating Backend Systems

- Connect to external APIs or databases to provide users with dynamic content
- Set up fulfillment to execute business logic based on user queries
- Ensure security and privacy considerations are met when handling user data

Testing – Conducting Tests



Simulation and User Testing

Test the conversational flow thoroughly with simulations

Collect feedback from real users to identify areas of improvement

Track down and fix any bugs or issues within the flow

Testing – Refining the Agent

Iterative (Continuous) Improvements

- Make incremental changes based on test results
- Update training phrases and entity definitions for better accuracy
- Continuously monitor and evaluate the system performance

Deployment – Launching the Agent

Preparing for Live Environment

01 Ensure all configurations are correctly set for a production environment

02 Set up monitoring tools to track performance and user interactions

03 Plan for ongoing maintenance and future enhancements

Post-Deployment Activities

Monitoring and Support

Monitor system performance and user satisfaction

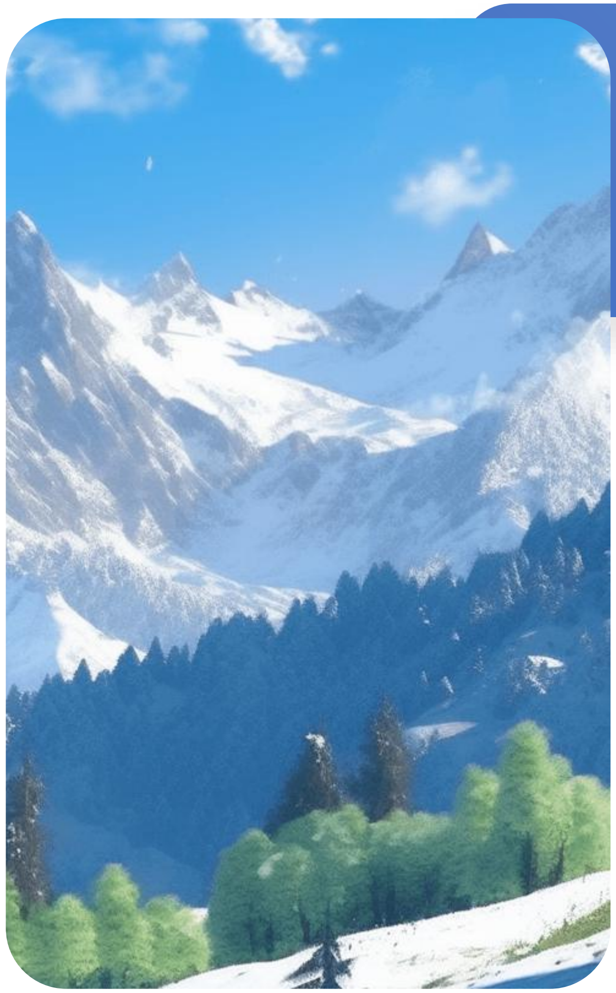
Regularly update and refine the agent based on new requirements and feedback

Provide support channels for user queries and issues

03

Entry Points & Transitions in Flows

Understanding Entry Points



Types of Entry Points

Default start flow entry point

Entry points from other flows

Intents that trigger entry points

External events that initiate entry points

Configuring Entry Points

Setting Default Start Flow

- Navigating to the flow’s entry point settings
- Selecting the default start flow
- Verifying the initial trigger conditions

Creating Entry Points from Other Flows

- Linking entry points to specific transitions
- Utilizing intents and events to control flow entry
- Managing multiple points of entry within complex flows

Handling Intents as Entry Points

- Mapping intents to specific entry points
- Prioritizing intents based on conversational context
- Adjusting settings to handle conflicting intents

Understanding Transitions

Types of Transitions

- State transitions within a flow
- Cross- flow transitions
- Event- triggered transitions
- Conditional transitions based on user input

Configuring Transitions

Creating State Transitions

- Identifying states within a flow
- Mapping paths between states
- Utilizing conditions to control state progression

Configuring Cross-Flow Transitions

- Setting up transitions between different flows
- Managing inter-flow connectivity
- Ensuring context preservation across flows

Event-Triggered Transitions

- Defining external events that trigger transitions
- Configuring system events and custom events
- Implementing real-time event handling in flows

Implementing Conditional Transitions

- Setting conditions based on user input
- Utilizing parameters and context for transitions
- Ensuring seamless conversational continuity

04

Best Practices

Defining User-Friendly Flows

01

Ensuring Clarity

Clear flows are essential for user engagement. Use straightforward language, familiar icons, and a logical sequence to make navigation intuitive for all users.

02

Reducing Complexity

Simplifying tasks enhances user experience. Break down complex processes into manageable steps, minimizing cognitive load and helping users to stay focused on their objectives.

Testing and Iteration

User Testing Techniques

Conduct direct user testing to gather feedback on flow effectiveness. Tools like user interviews, surveys, and observation help identify pain points and areas for improvement.

Iterative Development Approach

Embrace an iterative design process where flows are continuously refined based on user feedback. This approach encourages responsiveness to user needs and promotes ongoing enhancements.

Identifying Flow Errors

01

Debugging Techniques

Effective debugging strategies, including the use of breakpoints, logging messages, and step-by-step execution can assist with identifying and fixing flow errors.

02

Common Pitfalls

Include common mistakes encountered during flow design, such as infinite loops, incorrect conditions, and mismanaged variables that can disrupt the intended flow.

Debugging Techniques – Using the Simulator

Steps to Use

01 Open the DialogFlow CX console (https://dialogflow.cloud.google.com/cx/projects)

02 Navigate to your project & agent and click on "Test Agent"

03 Enter a user query in the simulator panel

04 Observe flow/intent triggered, parameters extracted, and fulfillment responses

05 Note active flow and intent if incorrect behavior occurs

Debugging Techniques - Logging

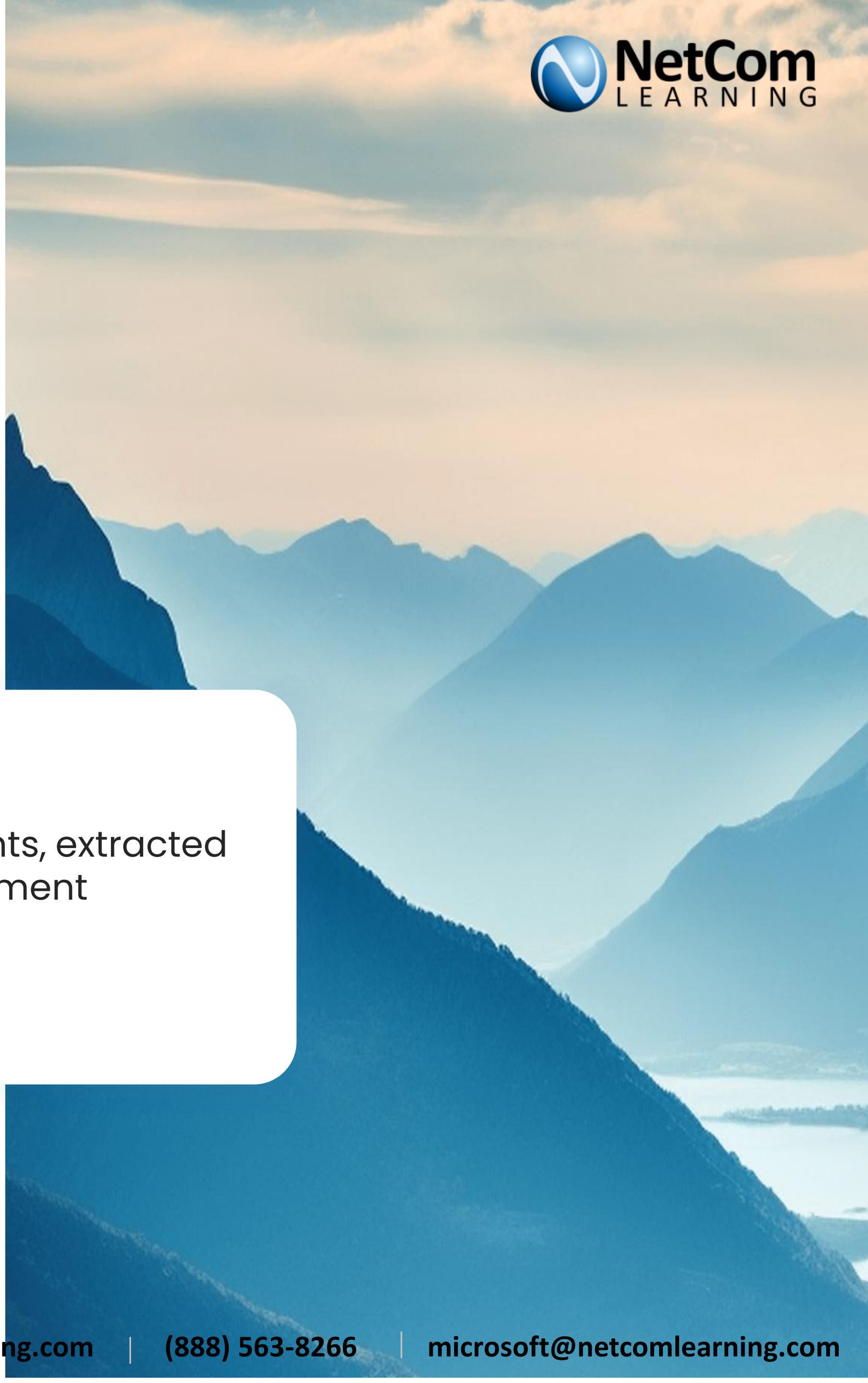
Benefits of Logging

Track execution of flows

Gain visibility into matched intents, extracted parameters, and fulfillment responses

Identify errors and API call failures

Debugging Techniques - Breakpoints



Benefits of Breakpoints

Pause execution at specific flow points

Inspect triggered intents, extracted parameters, and fulfillment execution

How to Use Breakpoints

Steps to Use Breakpoints

Identify problematic transitions in the flow

Add a conditional breakpoint before a transition to inspect inputs

Use the DialogFlow CX simulator to trigger the breakpoint

Review execution for unexpected intents, parameters, and transitions

Common Transition Issues

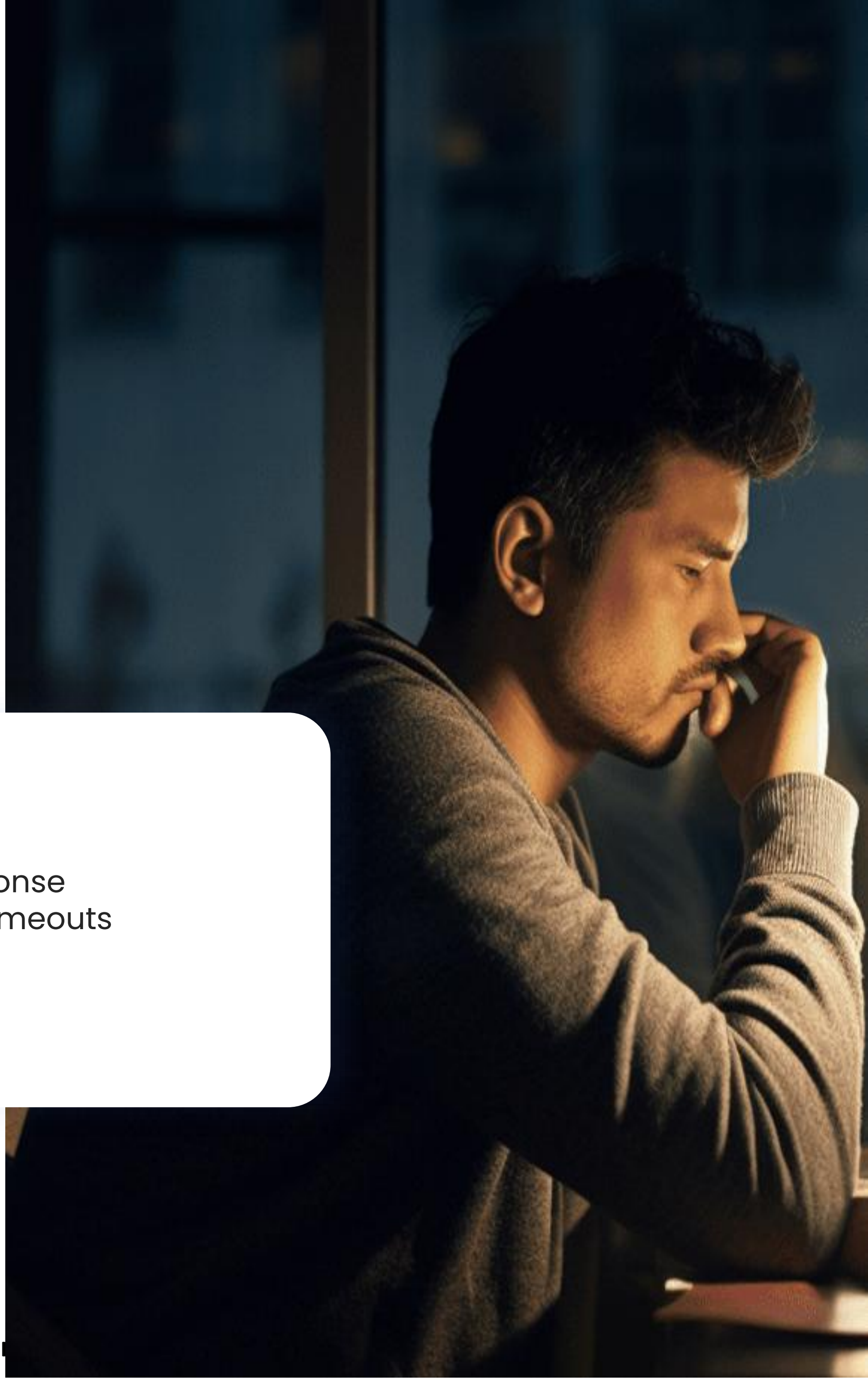
Types of Issues

- Unexpected route triggering
- Parameter extraction failure
- Looping behavior

Fulfillment – Common Issues

Types of Issues

- Webhook not responding
- Incorrect JSON response format
- Slow response causing timeouts



Steps to Debug Webhooks

Steps to Inspect and Correct

- Check Webhook Logs in Google Cloud Logging
- Identify error messages or failed requests
- Verify the response JSON structure and parameter formatting
- Test API calls separately using Postman or cURL
- Enable Webhook Debug Mode for request payload logging

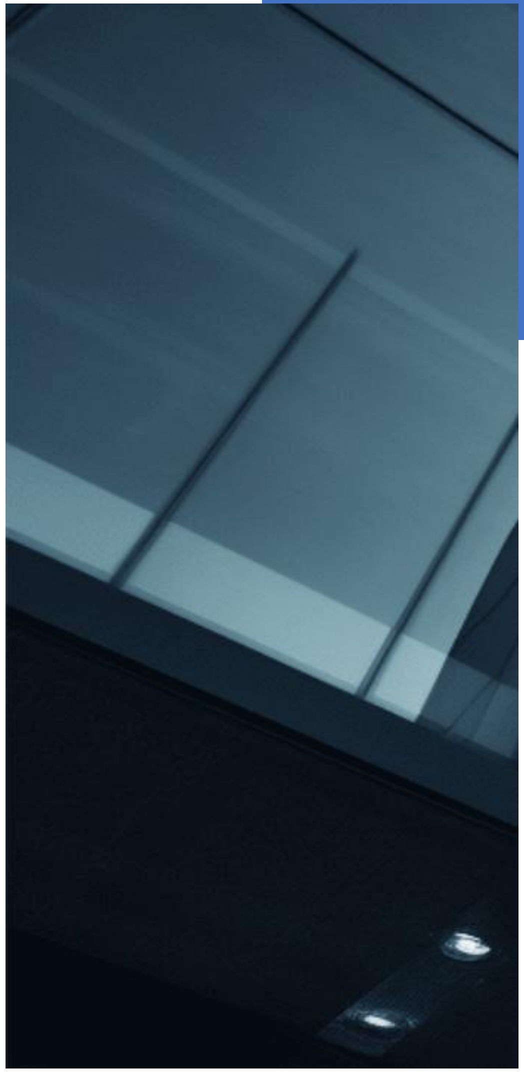
Common Issues with Intents

Types of Issues

01 Incorrect intent triggering

02 No intent match

03 Intent conflicts due to overlap



Debugging Intent Issues

Steps to Inspect and Correct

01

Add diverse training phrases for better accuracy

02

Remove ambiguous phrases with intent overlap

03

Review confidence scores in logs and improve training data

04

Use @sys.any entity for capturing varied user inputs

Enabling Verbose Logging

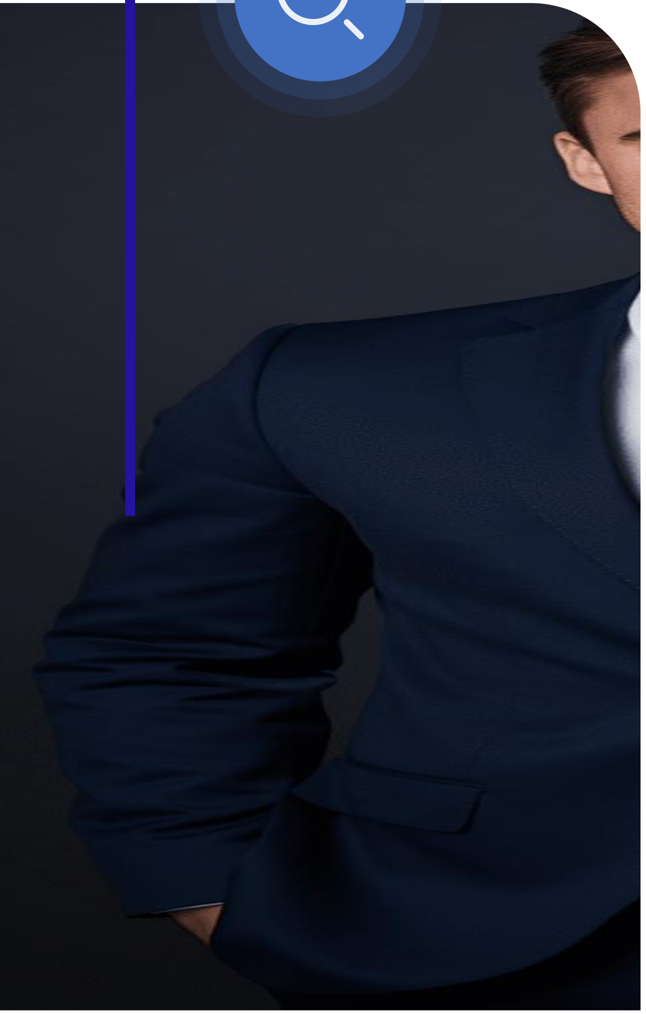
Steps to Enable

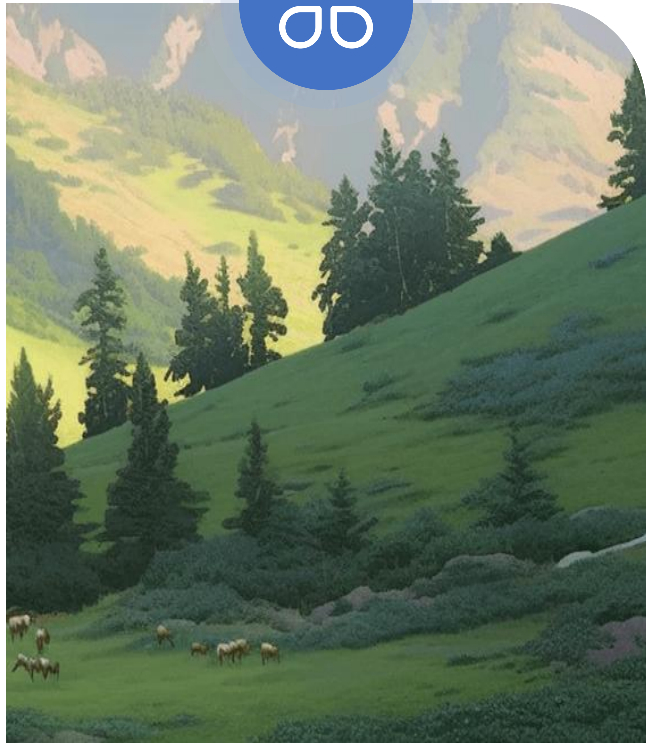
Go to Google Cloud Console → Operations → Logging

Use structured queries like resource.type="dialogflow_cx" and severity>=WARNING

Analyze logs for detailed error messages and execution traces

Flow Reusability





01 **Flow Reusability**

- Try to create reusable flows that can simplify maintenance and reduce redundancy
- This can ultimately enhance project scalability, execution, efficiency, and cost optimization

02 **Modular Design Strategies**

- Leverage modular design to help promote reuse
- Also benefits to breaking down complex systems into smaller, more manageable components
- Helps with organization, understanding, and ongoing maintenance

Thank you

NetCom LEARNING