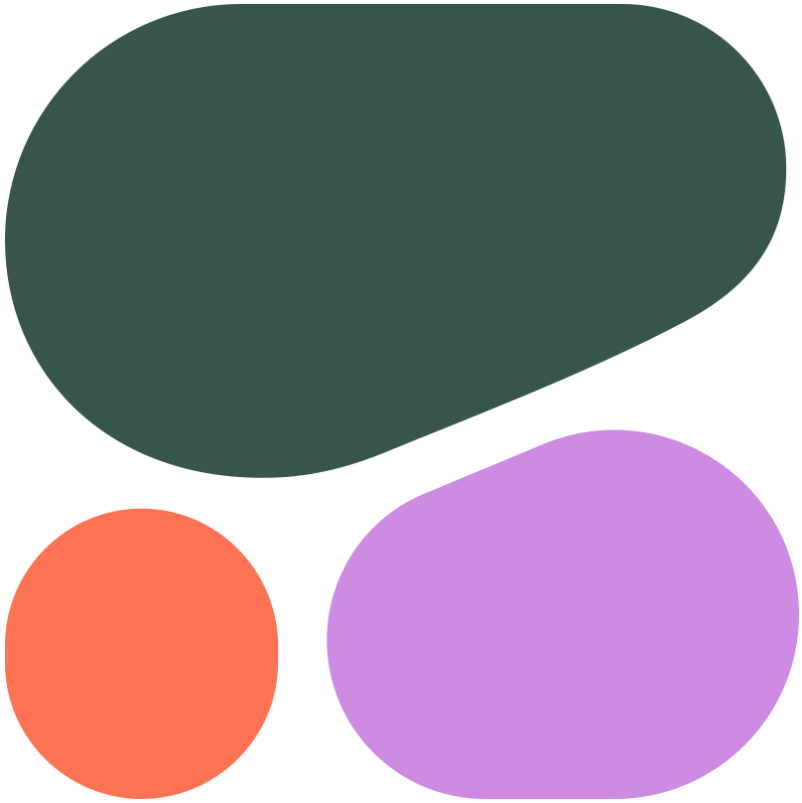


Cohere Connected System for Appian

V1.0.0

Appian Corporation

Version 1.0.0

Table of Contents

Overview

Cohere's language models can generate intelligent text completions using state-of-the-art AI. The Cohere Appian plugin allows users to leverage these capabilities from within the Appian platform.

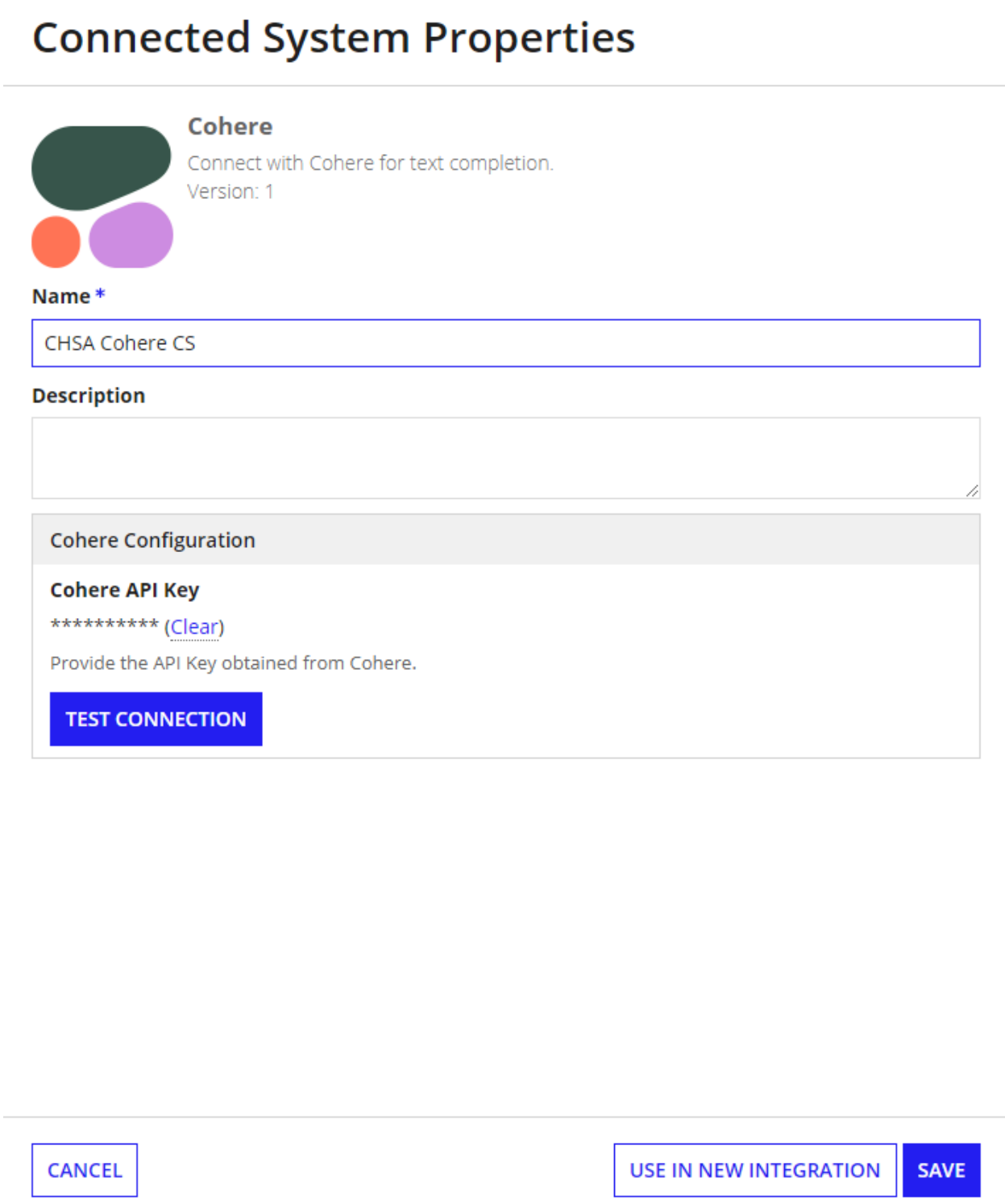
This plugin handles authentication with Cohere's API using API keys, enabling secure access to advanced text generation features.

By integrating Appian with Cohere's powerful language models, this plugin makes AI-assisted writing easily accessible for automating and enhancing text-based tasks. Users simply provide a text prompt and Cohere generates relevant continuations of coherent text.

Features

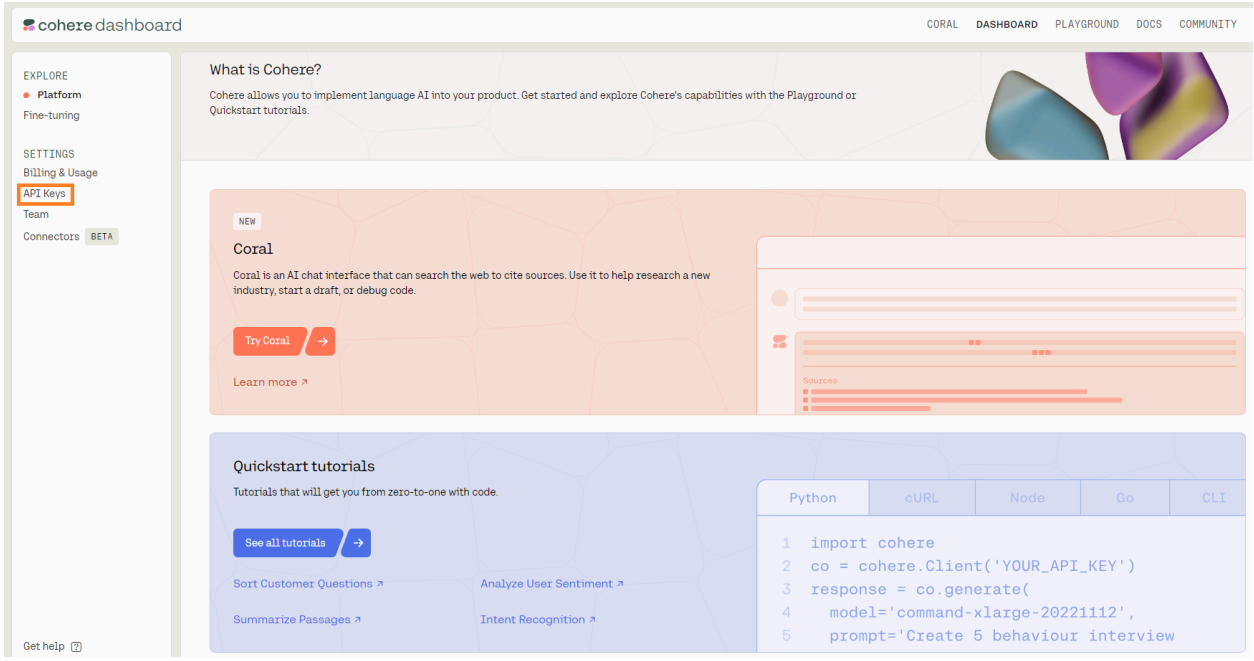
- Make API calls to Cohere Completions endpoint to generate text completions using Cohere’s API authentication
- Configure inputs such as Cohere model, number of generations, max tokens, temperature, topK, topP to customize completions
- Seamlessly integrate completions generation into Appian workflows with easy setup of connected system

Connected System Configuration



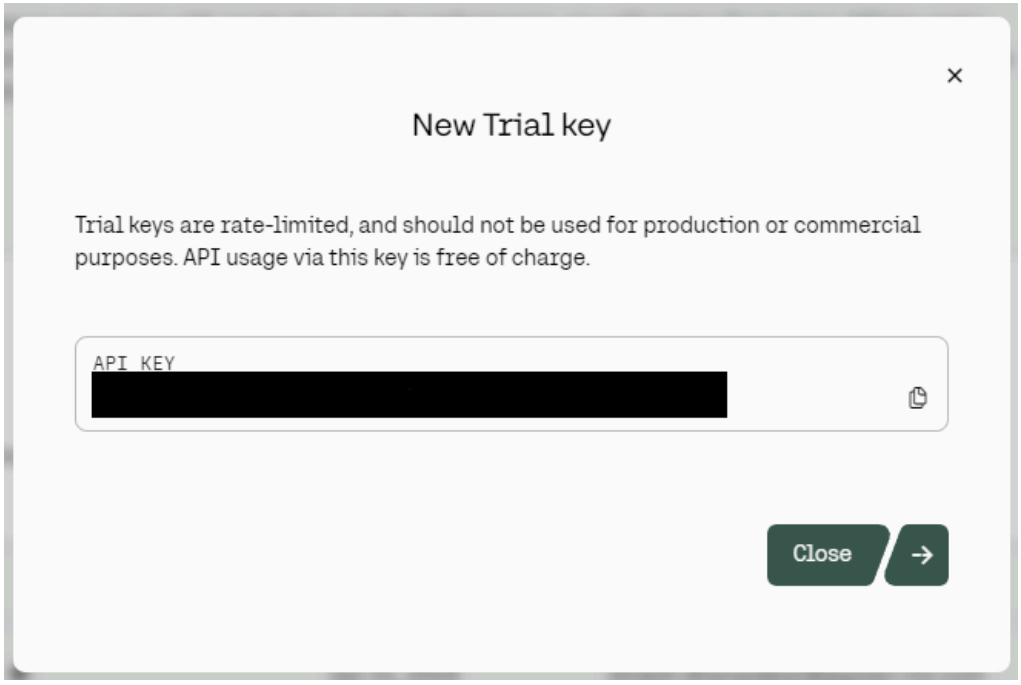
Steps to obtain Cohere API Key

1. **View API Keys:** In the [Cohere Dashboard], click on Click on **API Keys** under Settings to view all the API keys associated with your account.



2. **Create new API Key:** Click on the **New Trial Key** button to create a new API key for free use. To obtain API key for Production use, click on **Get your Production Key**. Provide a name for the key in the popup and Click on **Create Key**.

3. **Save the API Key:** Once the key is created, make sure to save it for later use. This Copied API Key will be used in the Connected System configuration.



Integration Configuration

This Integration generates realistic text conditioned on a given input.

Inputs:

Prompt (Text) - Required - Text input as the starting point for generating the response.

Model (Text Dropdown) - Required - Cohere model that will complete your prompt. You can select between "command", "command-light" and "Other" options.

Model Name (Text) - Visible and Required when Other is selected in Model - Name of the model you would like to use.

Number of Generations (Number(Integer)) - Optional - Maximum number of generations that will be returned. Accepts values between 1 and 5. Defaults to 1.

Max Tokens (Number(Integer)) - Optional - Maximum number of tokens the model will generate as part of the response. No limit set by default.

Truncate (Text Dropdown) - Optional - Specify how the API will handle inputs longer than the maximum token length. Valid values: NONE, START, END. Defaults to END.

Temperature (Number(Decimal)) - Optional - Tunes the degree of randomness in generation. Defaults to 0.75.

End Sequences (List of Text) - Optional - Sequences as List of Text that will cause the model to stop generating completion text. The sequence will be excluded from the text.

Stop Sequences (List of Text) - Optional - Sequences as List of Text that will cause the model to stop generating completion text. The sequence will be included from the text.

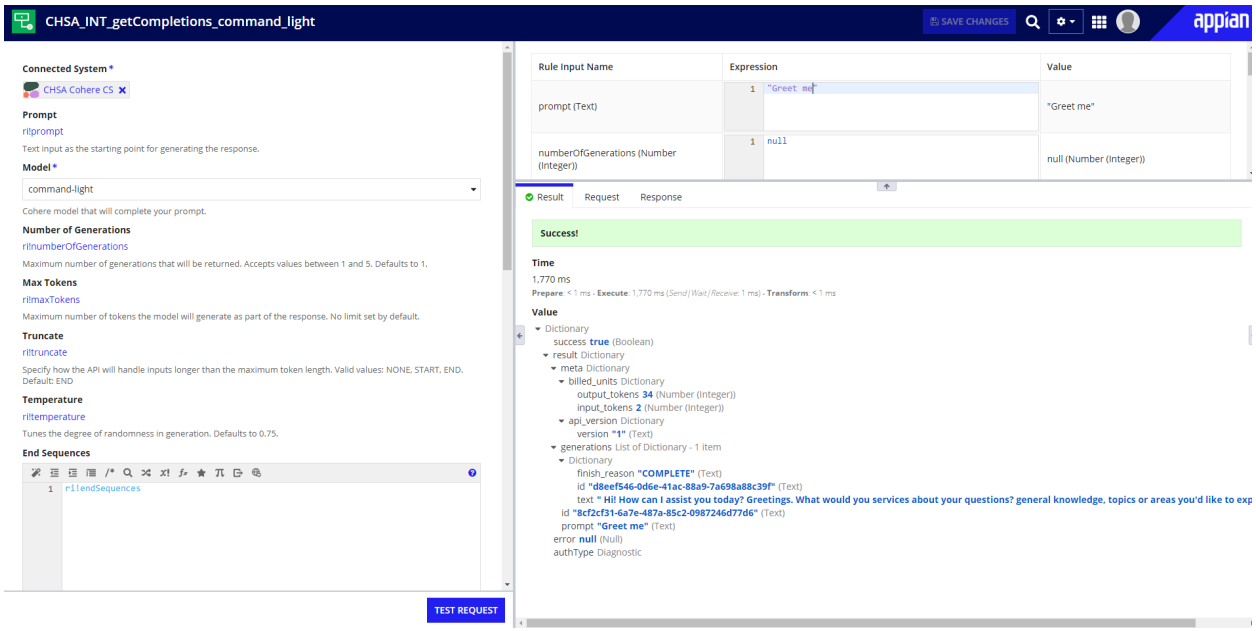
K (Number(Integer)) - Optional - Ensures only the top k most likely tokens are considered for generation at each step. Accepts values between 0 and 500. Defaults to 0.

P (Number(Decimal)) - Optional - Ensures that only the most likely tokens, with total probability mass of p, are considered for generation at each step. If both k and p are enabled, p acts after k. Accepts values between 0.01 and 0.99. Defaults to 0.75.

Frequency Penalty (Number(Decimal)) - Optional - Used to reduce repetitiveness of generated tokens. Accepts values between 0 and 1.0. Defaults to 0.0.

Presence Penalty (Number(Decimal)) - Optional - Used to reduce repetitiveness of generated tokens. Accepts values between 0 and 1.0. Defaults to 0.0.

Return Likelihoods (Text Dropdown) - Optional - Specify how and if the token likelihoods are returned with the response. Valid values: GENERATION, ALL, NONE. Default: NONE



Output:

```
JavaScript
{
  success: true,
  result: {
    meta: {
      billed_units: { output_tokens: 11, input_tokens: 3 },
      api_version: { version: 1 }
```

```
    },
    generations: {
      finish_reason: "COMPLETE",
      id: "01bbc878-4577-4254-acd8-5ede1ff16117",
      text: "Hi! I am happy to greet you!"
    },
    id: "a2e2e02f-0e16-4cab-81e0-32a0139c90bc",
    prompt: "greet me"
  }
}
```