



# Red Hat build of Apache Camel 4.4

## Kaoto

Create and edit integrations based on Apache Camel with Kaoto



## Red Hat build of Apache Camel 4.4 Kaoto

---

Create and edit integrations based on Apache Camel with Kaoto

## Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This guide provides a comprehensive overview of Kaoto, a visual design tool tailored for creating Apache Camel integrations. It covers both the installation process and the utilization of Kaoto to expedite the development lifecycle.

---

## Table of Contents

|  |          |
|--|----------|
| <b>PREFACE</b> .....                               | <b>3</b> |
| MAKING OPEN SOURCE MORE INCLUSIVE .....            | 3        |
| <b>CHAPTER 1. OVERVIEW OF KAOTO</b> .....          | <b>4</b> |
| 1.1. WHY KAOTO? .....                              | 4        |
| <b>CHAPTER 2. INSTALLING KAOTO</b> .....           | <b>6</b> |
| <b>CHAPTER 3. GETTING STARTED WITH KAOTO</b> ..... | <b>7</b> |
| 3.1. RUNNING THE CAMEL ROUTE .....                 | 12       |



## PREFACE

[Kaoto](#) stands out as an intuitive visual platform for Apache Camel integrations, specifically designed to streamline the development process. It offers an accessible entry point for junior integration engineers through its low-code/no-code capabilities, making the initial steps of integration development straightforward. Moreover, Kaoto supports a seamless transition to more sophisticated features, allowing expert Camel developers to develop and fine-tune complex integration routes effectively.

The audience for this guide is Apache Camel developers. This guide assumes familiarity with Apache Camel and the processing requirements for your organization.

### MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

# CHAPTER 1. OVERVIEW OF KAOTO



## IMPORTANT

The VS Code extensions for Apache Camel are listed as development support. For more information about the scope of development support, see [Development Support Scope of Coverage for Red Hat Build of Apache Camel](#).

[Kaoto](#) is an acronym for **K**amel **O**rchestration **T**ool. It is a low code and no code integration designer to create and edit integrations based on [Apache Camel](#). Kaoto is extendable, flexible, and adaptable to different use cases. For more information about the history of Kaoto, see [Statistics and History of Kaoto](#).

Kaoto offers a rich catalog of building blocks for use in graphical design. By default, Kaoto loads the official upstream [Camel Catalog](#) and [Kamelet Catalog](#).

Benefits of using Kaoto can be listed as follows:

- **Enhanced Visual Development Experience**  
By leveraging Kaoto's visual designing capabilities, users can intuitively create, view, and edit Camel integrations through the user interface. This low-code/no-code approach significantly reduces the learning curve for new users and accelerates the development process for seasoned developers.
- **Comprehensive Component Catalog Accessibility**  
Kaoto provides immediate access to a rich catalog of Camel components, enterprise integration patterns (EIPs), and Kamelets. This extensive Catalog enables developers to easily find and implement the necessary components for their integration solutions. By having these resources readily available, developers can focus more on solving business problems rather than spending time searching for and learning about different components.
- **Streamlined Integration Development Process**  
The platform is designed with an efficient user experience in mind, optimizing the steps required to create comprehensive integrations. This efficiency is achieved through features like auto-completion, configuration forms, and interactive feedback mechanisms. As a result, developers can quickly assemble and configure integrations, reducing the overall development time. This streamlined process encourages experimentation and innovation by making it easier to prototype and test different approaches.

## 1.1. WHY KAOTO?

- **Camel at Heart**  
Using the power of Apache Camel: Kaoto utilizes the Apache Camel models and schemas to always offer you all available Camel features.
- **Local Development**  
VS Code Extension: We provide Kaoto as an extension you can install from the [Microsoft Marketplace](#) and also from the [Open VSX Marketplace](#).
- **LCNC: Low Code and No Code**  
Care about developers: You can seamlessly switch between any IDE and Kaoto, allowing mixed teams and converting low-code integrators to developers.
- **FLOSS heart**



Free Libre and Open Source Forever: Truly open with no vendor lock-in. Use, reuse, share, modify, and resell to the users' needs.

## CHAPTER 2. INSTALLING KAOTO



### IMPORTANT

It is recommended to install the [Extension Pack for Apache Camel by Red Hat](#) . It will provide a set of tools to manage, edit, run, and debug Camel integrations in various contexts.

The following procedure explains how to install VS Code and other essential extensions required to get started with Kaoto.

1. If you do not have Visual Studio Code installed, install it from [here](#).
2. Launch VS Code.
3. Install the [Extension Pack for Apache Camel by Red Hat](#) into your Visual Studio Code instance.
4. To run and debug Camel integrations using VS Code UI, install [JBang CLI](#). Optionally, to run Camel integrations from the command line, install [Camel CLI](#).



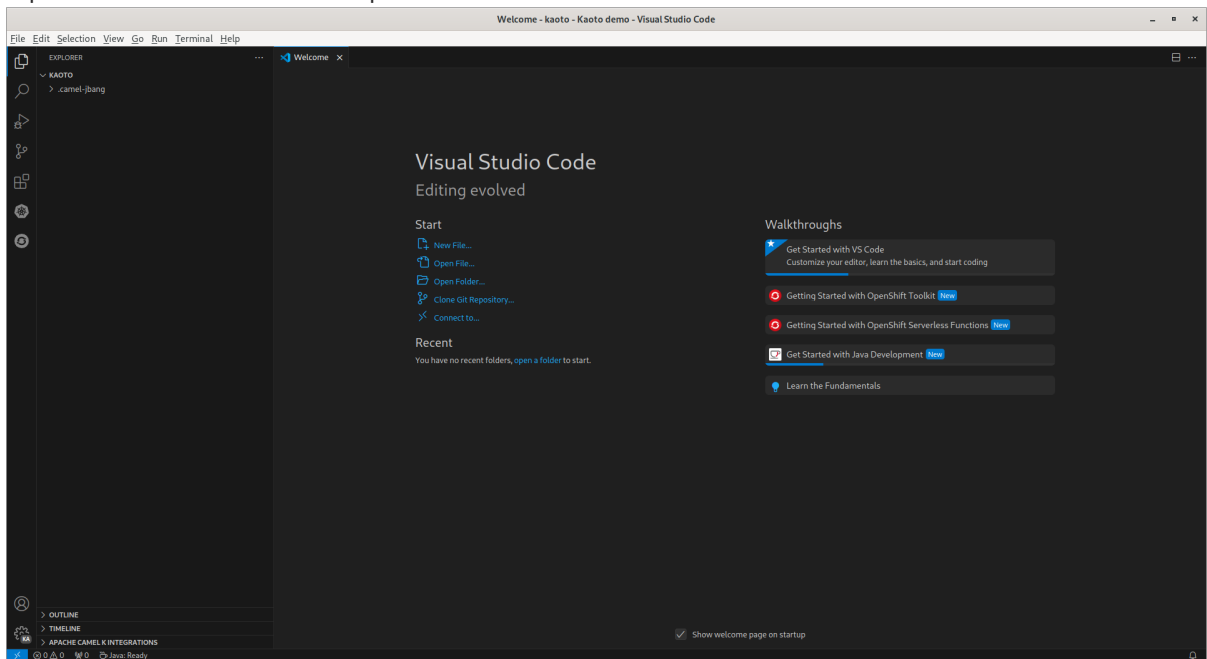
### IMPORTANT

Create and select a workspace folder where all the integration will be stored. Selecting a workspace is important to ensure all required commands will be available and behave correctly.

## CHAPTER 3. GETTING STARTED WITH KAOTO

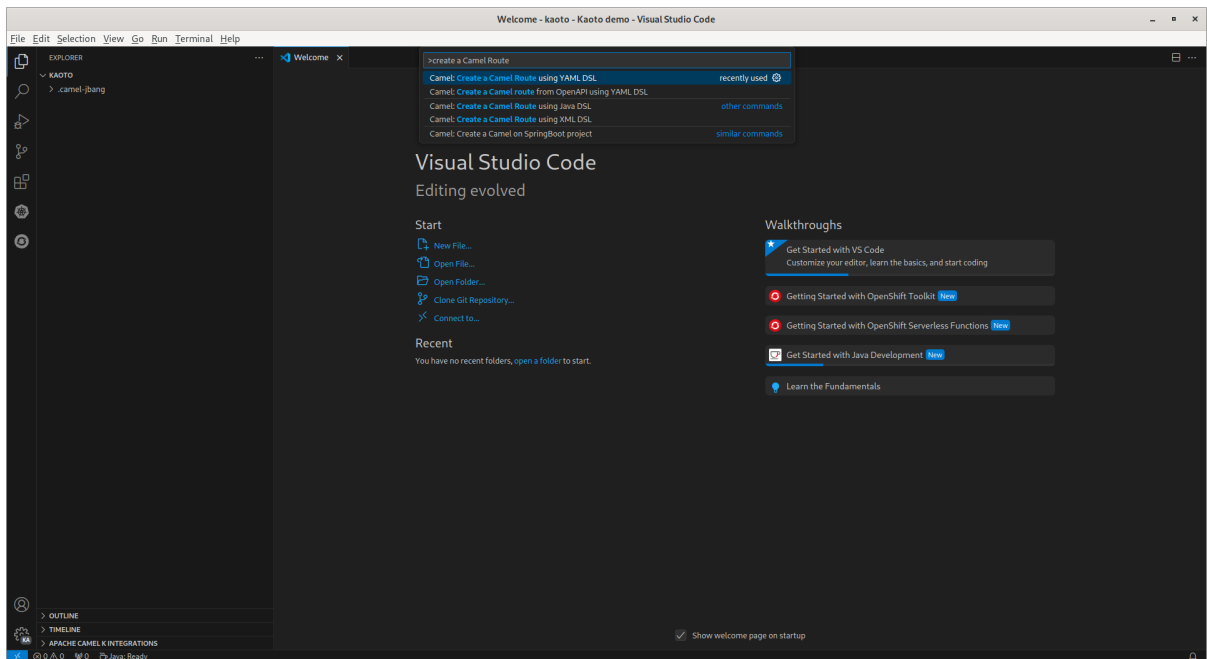
The following procedure explains how to create and store integrations with Kaoto.

1. Open VS Code and select "Open Folder".

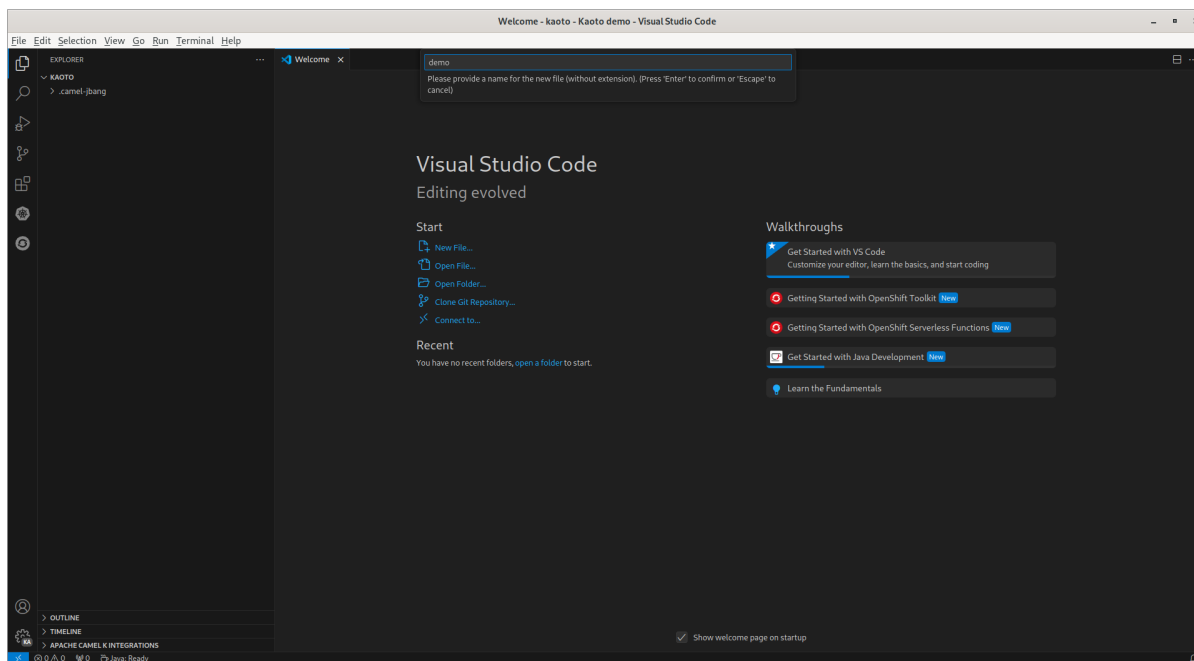


2. In the folder selection dialog select the folder to store your integrations or create a new folder and select it.
3. Open Command Palette (**Ctrl+Shift+P**), paste the following command, and press enter.

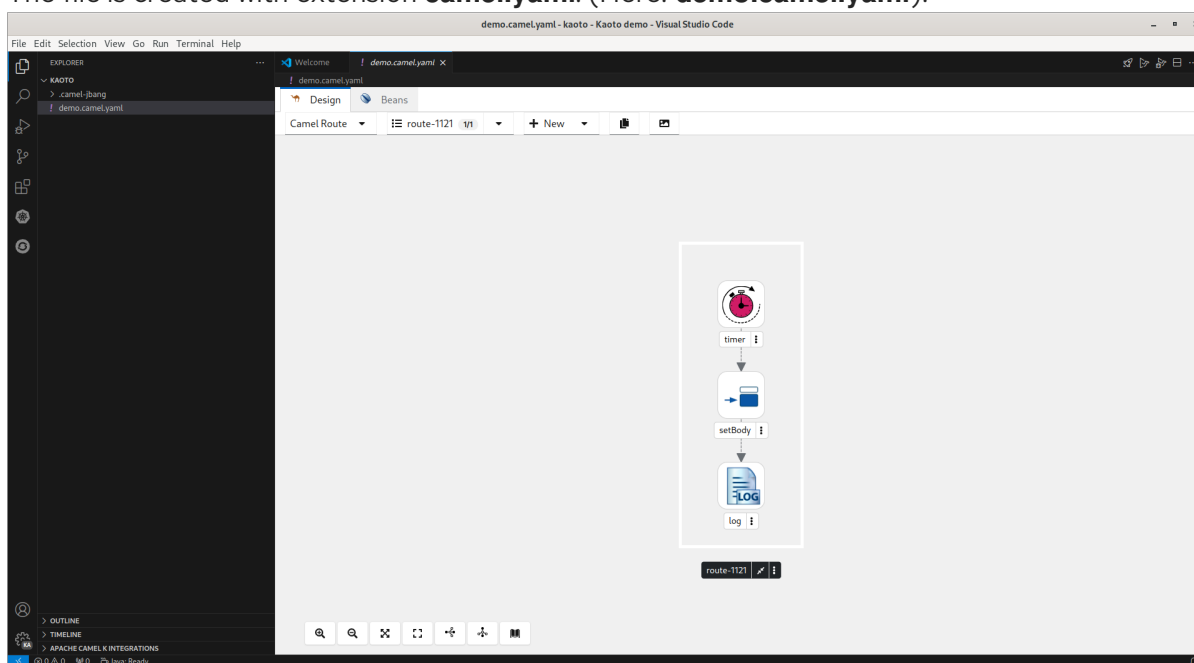
Create a Camel route using YAML DSL



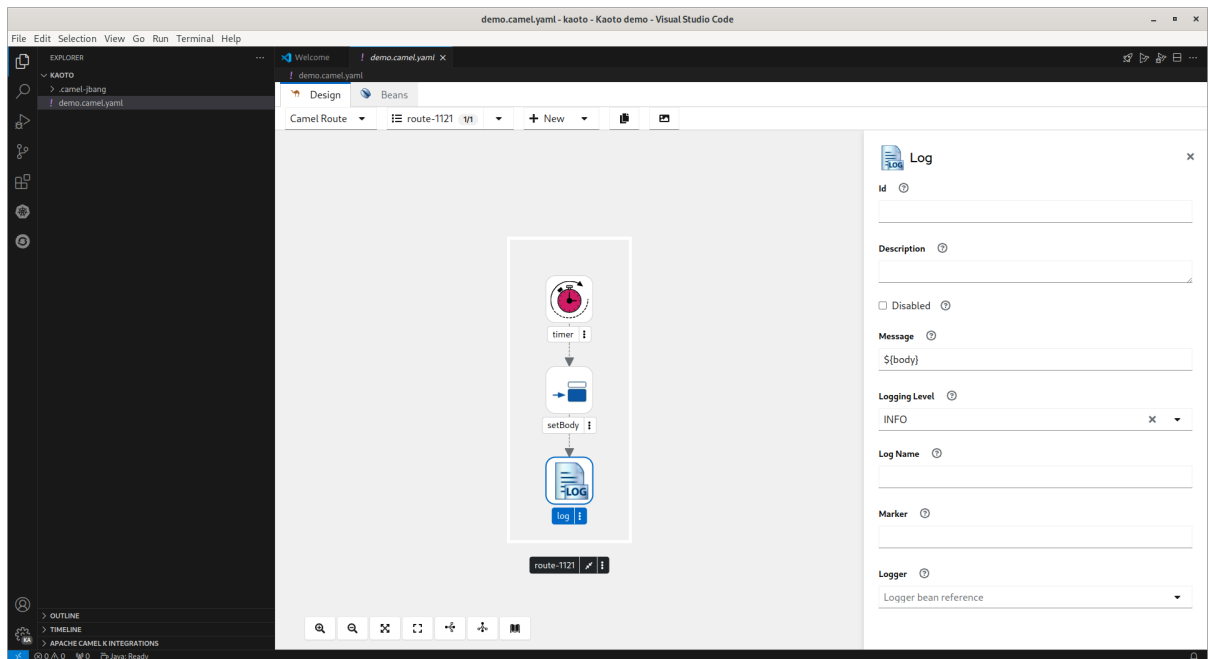
4. Provide a name for the new file without extension and press enter.



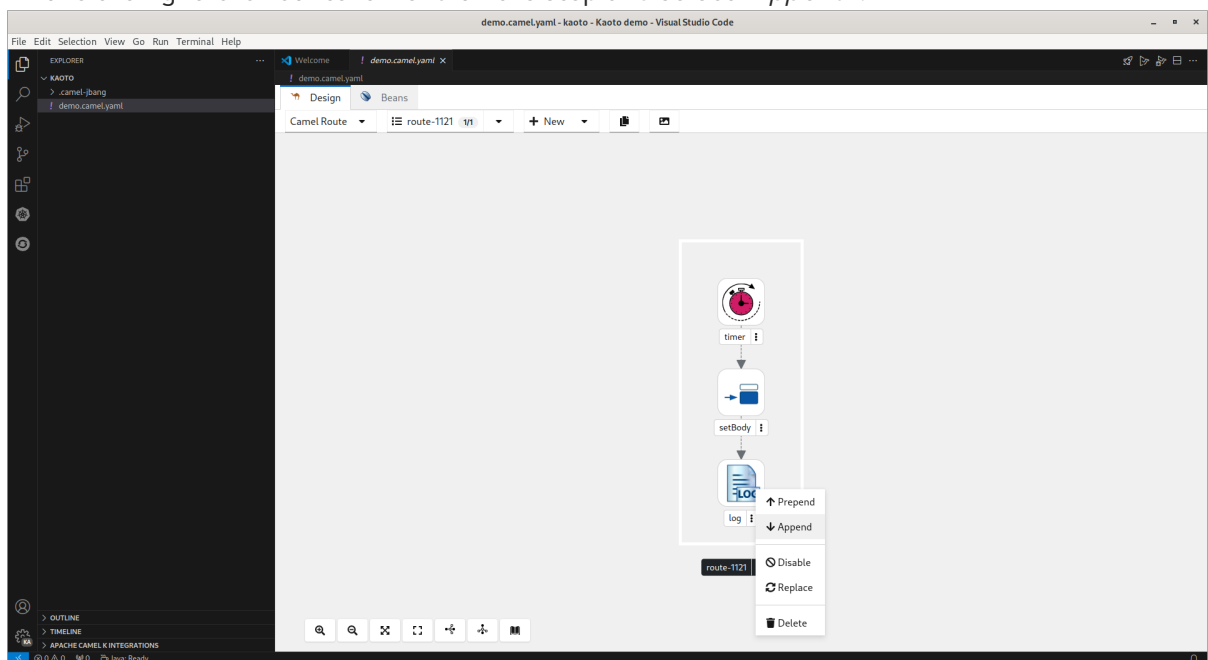
5. The file is created with extension **camel.yaml**. (Here: **demo.camel.yaml**).



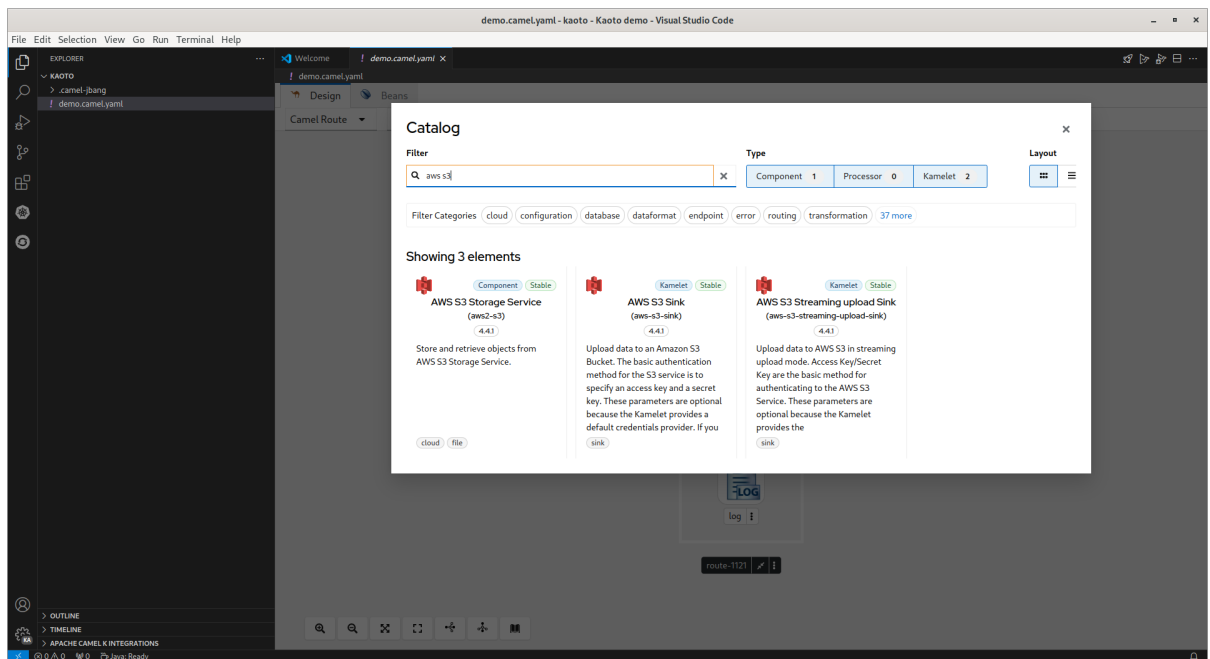
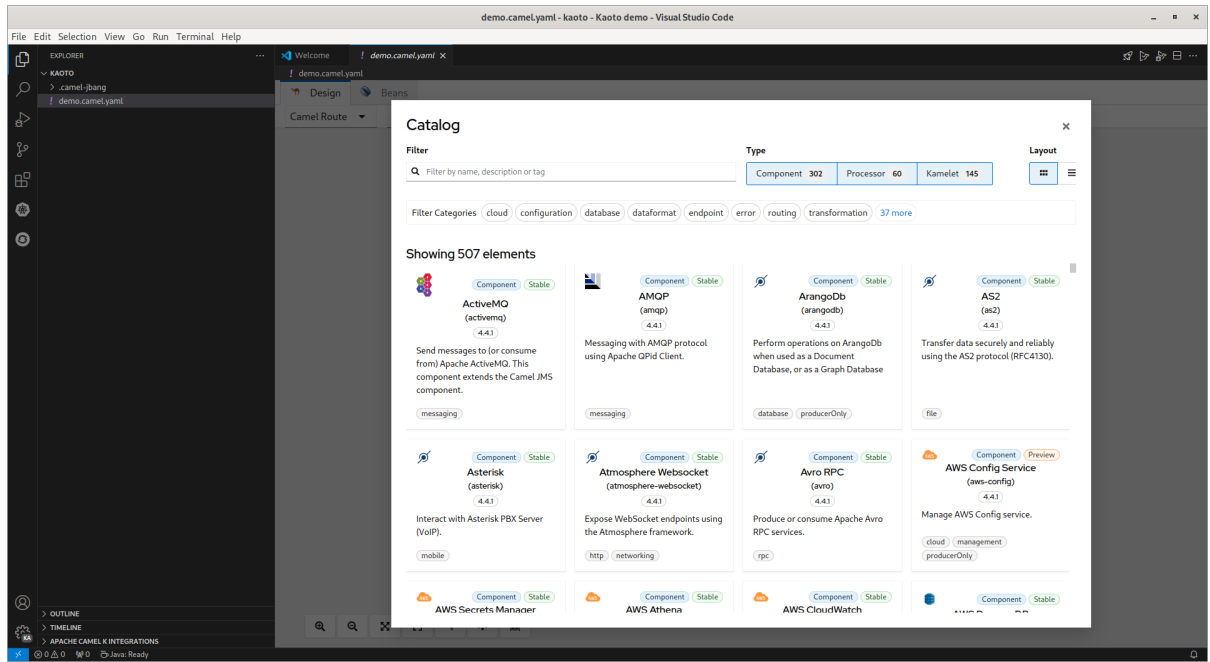
6. Select the log step to configure the **Message** in the configuration panel.

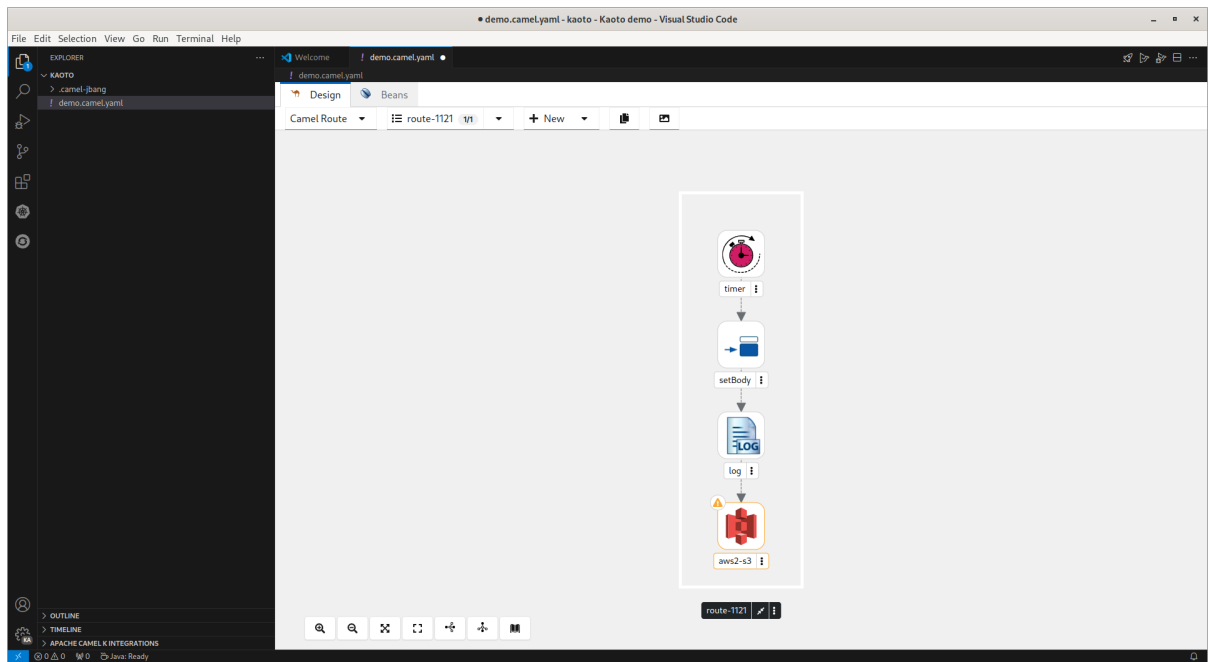


7. Using the icons below the route image, you can choose between a vertical/horizontal layout of the route, zoom in/out, etc. **Open Catalog** on the far right displays the Camel catalog. There are also several filtering options in the Catalog, which greatly simplifies finding what you need.
8. To add a component to the Camel route, click on the dot pattern of an existing component or invoke the right-click context menu on the step and select "Append".

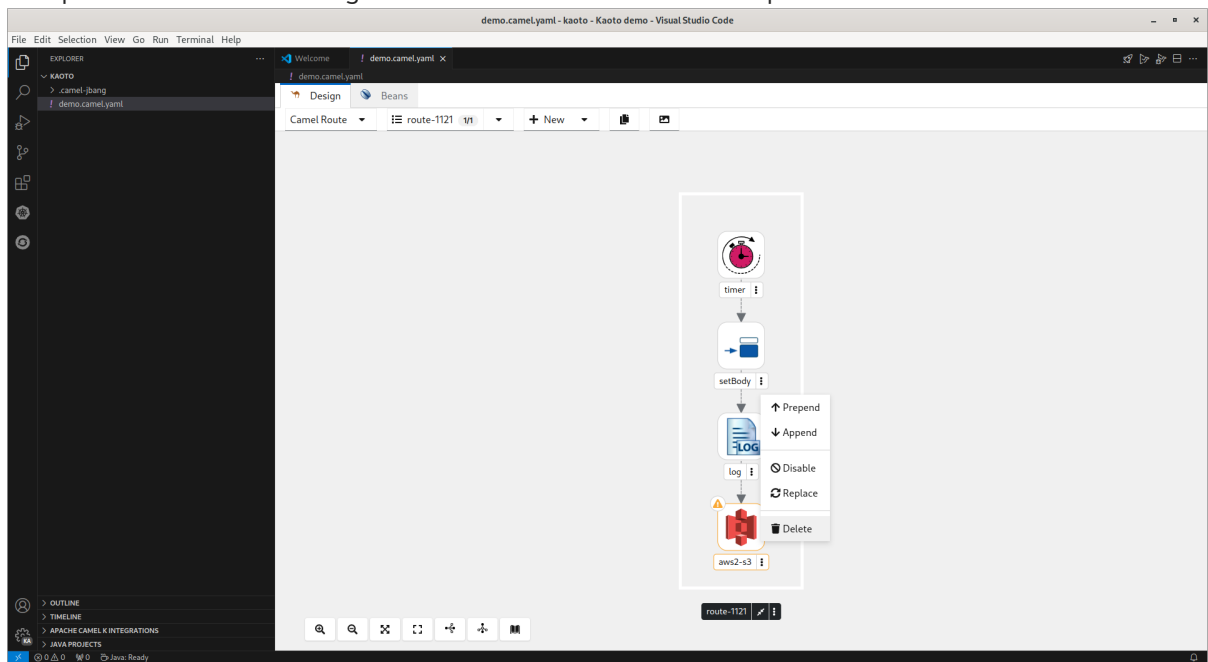


9. The camel component Catalog will be displayed, click on the additional component and it will be added to the route.

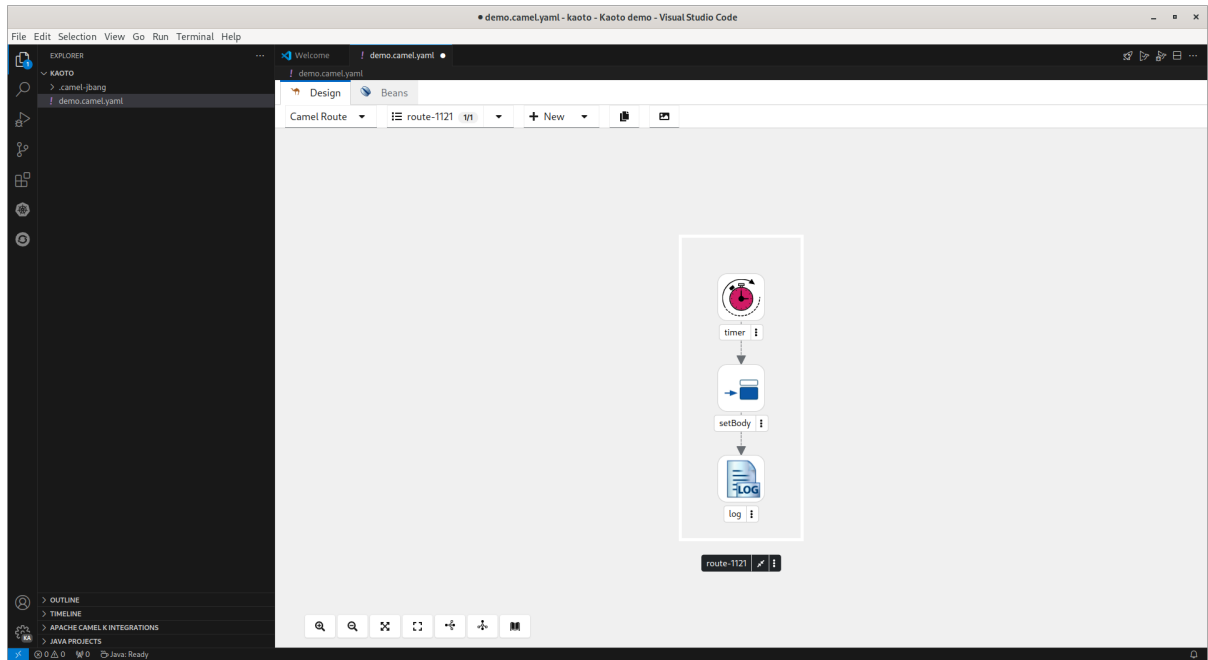




- To remove the added component from the Camel route, click on the dot pattern of the existing component or invoke the right-click context menu on the step and select "Delete".



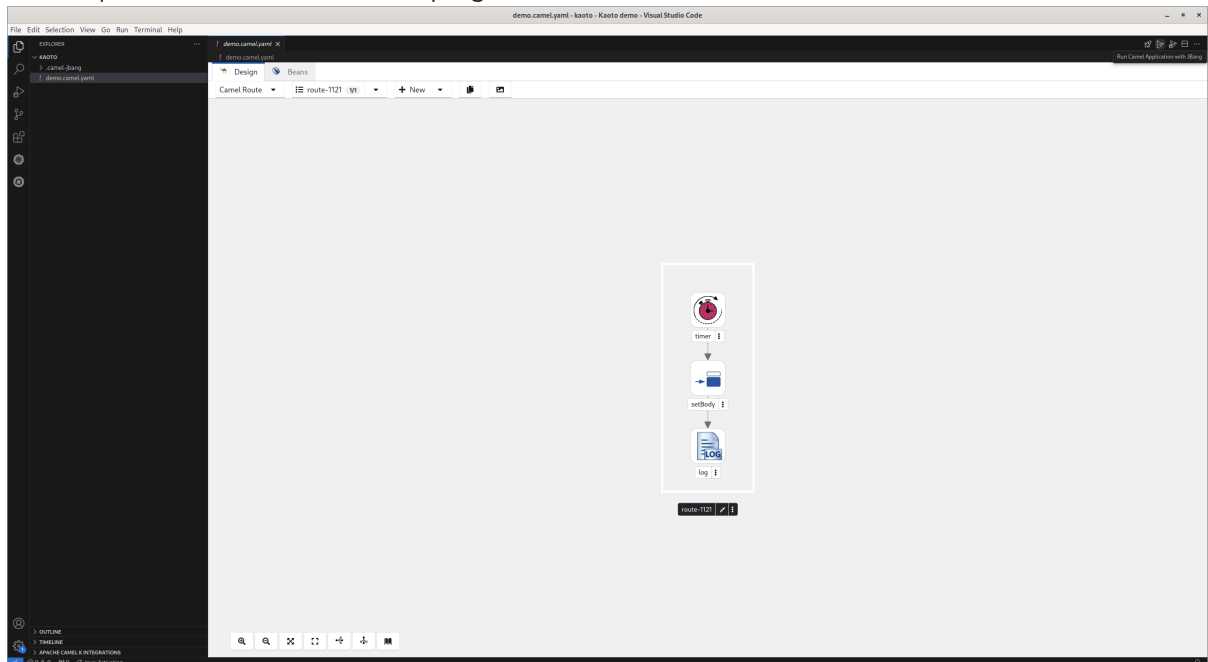
- The existing component will be removed from the route.



### 3.1. RUNNING THE CAMEL ROUTE

To get started with simple routes, they can be launched with Camel JBang.

1. With **demo.camel.yaml** open, click the button **Run Camel Application with JBang** in the editor quick action menu at the top right of the editor.



2. The terminal will open with the running Camel route. It can take several seconds the first time for Camel JBang to initialize.



The screenshot shows the Visual Studio Code interface with the Kaoto IDE extension. The main workspace displays a Camel route diagram for 'route-1121'. The diagram consists of a 'timer' component followed by a 'setBody' component. The terminal window at the bottom shows the following logs:

```

2024-05-09 12:02:31.713 INFO 35571 --- [
0
main] org.apache.camel.main.MainSupport : Using Java 21 with PID 35571. Started by rmartinez in /home/rmartinez/repos/temp/kaot
2024-05-09 12:02:31.844 INFO 35571 --- [
main] apache.camel.main.ProfileConfigurer : The application is starting with profile: dev
2024-05-09 12:02:32.048 INFO 35571 --- [
main] hel.cli.connector.LocalCliConnector : Management from Camel JBang enabled
2024-05-09 12:02:32.199 INFO 35571 --- [
main] el.impl.engine.AbstractCamelContext : Apache Camel 4.5.0 (demo) is starting
2024-05-09 12:02:32.230 INFO 35571 --- [
main] t.FileWatcherResourceLoadStrategy : Live route reloading enabled (directory: .)
2024-05-09 12:02:32.272 INFO 35571 --- [
main] el.impl.engine.AbstractCamelContext : Routes startup (total:1)
2024-05-09 12:02:32.272 INFO 35571 --- [
main] el.impl.engine.AbstractCamelContext : Started route-1121 (timer://yaml)
2024-05-09 12:02:32.272 INFO 35571 --- [
main] el.impl.engine.AbstractCamelContext : Apache Camel 4.5.0 (demo) started in 162ms (build:0ms init:0ms start:162ms)
2024-05-09 12:02:33.258 INFO 35571 --- [
-timer://yaml] demo.camel.yaml:13 : Hello Camel from route-1121
2024-05-09 12:02:34.247 INFO 35571 --- [
-timer://yaml] demo.camel.yaml:13 : Hello Camel from route-1121
2024-05-09 12:02:35.247 INFO 35571 --- [
-timer://yaml] demo.camel.yaml:13 : Hello Camel from route-1121
2024-05-09 12:02:36.247 INFO 35571 --- [
-timer://yaml] demo.camel.yaml:13 : Hello Camel from route-1121
2024-05-09 12:02:37.247 INFO 35571 --- [
-timer://yaml] demo.camel.yaml:13 : Hello Camel from route-1121
2024-05-09 12:02:38.247 INFO 35571 --- [
-timer://yaml] demo.camel.yaml:13 : Hello Camel from route-1121
2024-05-09 12:02:39.248 INFO 35571 --- [
-timer://yaml] demo.camel.yaml:13 : Hello Camel from route-1121

```