Tips:

1. Das Repo klonen
2. Die Applikation unter „initial/src/main/java/gateway“ starten.

Das ist auch die Datei, welche editiert wird.

1. Die Curl Commands über ein Command Prompt, nicht die Powershell ausführen!
2. Aufpassen auf welchem Port der Server gestartet wird für die CURL Commands
3. Sicherstellen das Maven auf dem Rechner installiert ist und die Path/Maven\_HOME Variable Konfiguriert ist!
4. Mit mvn dependency:resolve die Maven dependencies installieren. Dafür muss ins /initial directory navigiert werden.
5. Wenn die Imports nicht richtig resolved werden können: > Java: Clean Java Language Server Workspace

Erklärungen:

Kunden stellen Anfragen an das Spring Cloud Gateway. Wenn das Gateway-Handler-Mapping feststellt, dass eine Anfrage zu einer Route passt, wird sie an den Gateway-Web-Handler gesendet. Dieser Handler führt die Anfrage durch eine Filterkette, die spezifisch für die Anfrage ist. Filter können sowohl vor als auch nach dem Senden der Proxy-Anfrage Logik ausführen. Alle "pre" Filter-Logiken werden zuerst ausgeführt. Dann wird die Proxy-Anfrage durchgeführt. Nachdem die Proxy-Anfrage durchgeführt wurde, wird die "post" Filter-Logik ausgeführt.

Auf gut Deutsch bedeutet dies, dass Kundenanfragen zunächst vom Spring Cloud Gateway empfangen werden. Wenn das Gateway feststellt, dass eine Anfrage zu einer der konfigurierten Routen passt, wird sie an den Gateway-Web-Handler weitergeleitet. Dort wird die Anfrage durch eine Filterkette geleitet, die spezifisch für diese Anfrage ist. Die Filter werden vor und nach dem Senden der Proxy-Anfrage ausgeführt, wobei alle "pre"-Filter zuerst und alle "post"-Filter danach ausgeführt werden.

Um die neue 'fallback'-Funktion zu testen, starten wir die Anwendung neu und geben den cURL-Befehl erneut ein. Leider hat der Befehl von der Webseite bei uns nicht funktioniert. Deshalb haben wir den folgenden Befehl verwendet:

Invoke-WebRequest -Uri 'http://localhost:9999/delay/3' -Headers @{'Host'='www.circuitbreaker.com'} -Method Get

Die Files Application.java und ApplicationTest.java mussten in verschiedene Ordner gezogen werden, weil es sonst Maven verwirrt hat. Diese Files sind jetzt in Main- und Testordner unterteilt.

Zum Abschluss führen wir die ApplicationTest-Klasse aus und stellen fest, dass die Tests erfolgreich waren:

