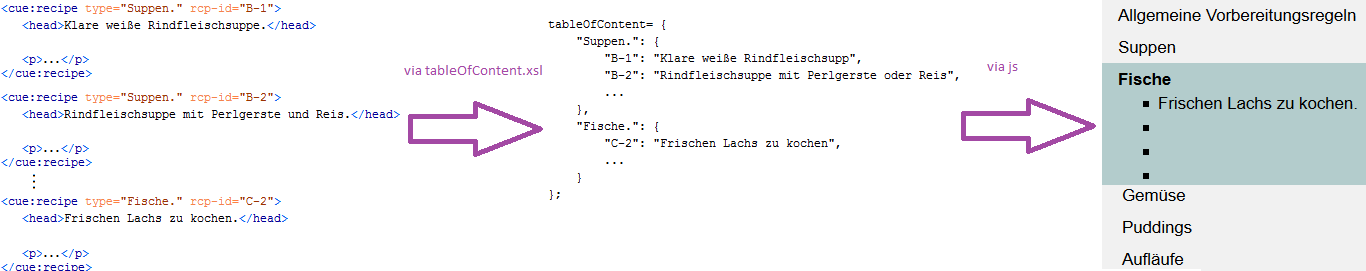
Diesen Blockeintrag werde ich in zwei Teile aufteilen. Hier erläutre ich kurz den statischen Datenflow von dem *recipes extracted cueML.xml* zu der Webseite, welcher sich in drei Teile aufspaltet:

1. Ein Inhaltsverzeichnis aufstellen
2. Die Rezepte extrahieren
3. Die Verweise einbauen

Im zweiten Teil werde ich meine Erkenntnisse zum Tag-Schemata zusammentragen. Ich denke, ich habe nun anhand der Beispiel-Webseite viel griffiger verstanden, was bis jetzt an meiner Zufriedenheit gekratzt hat. Aber zunächst kommt hier der erste Teil:

## 1. Ein Inhaltsverzeichnis aufstellen

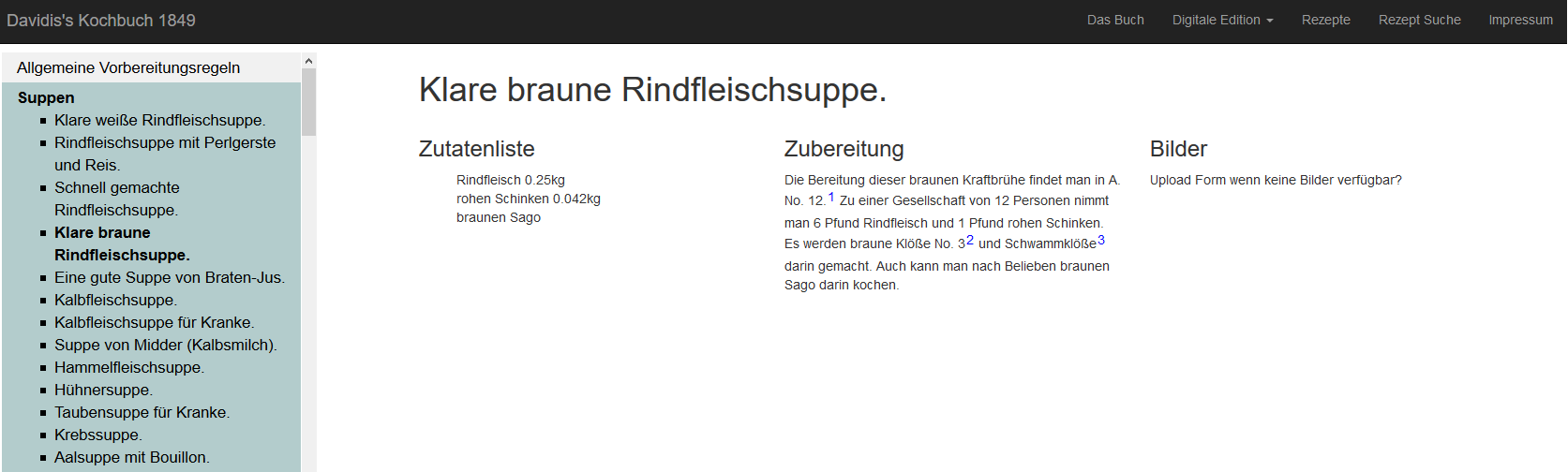
Das Inhaltsverzeichnis wird mittels XSLT in einem JSON-Objekt zur Verfügung gestellt. Das JSON-Objekt ist ein Dictionary, welches pro Rezept-Kategorie ein Dictionary anlegt. Dieses besteht aus rcp-id und Rezept-Name Paaren. Wenn die vorherige Rezept-Kategorie eine andere ist, als die aktuelle, legt das XSLT eine neue Kategorie an. Daher ist es wichtig, dass die Rezepte sortiert nach ihrer Kategorie vorliegen. Da die Rezepte ggf. danach vorsortiert werden können (was sie z.Z. bereits sind), stellt dies keine Einschränkung dar. Aus diesem JSON-Objekt kann dann eine Navigationsleiste erstellt werden.



## 2. Die Rezepte extrahieren

Für jedes Rezept wird mittels XSLT ein html-Dokument Namens rcp-id.html erstellt, welches via JS in die Webseite eingebunden werden kann.

Das html-Dokument besteht aus 3 Spalten. In der ersten Spalte ist eine Zutatenliste, in der 2. die Anleitung und in der 3. ein Bild. Für jedes link-Tag wird ein beginnend bei eins nummerierter Verweis angezeigt.



## 3. Die Verweise Einbauen

Die Verweise erscheinen bei Klick in einem transparenten Popup. Bei den Verweisen werden zwei Kategorien unterschieden:

1. Wenn auf ein ganzes Rezept verwiesen wird, wird rcp-id.html geladen.
2. Ansonsten wird der Text zwischen dem Start-tag bis zum End-tag geladen.

Technisch umgesetzt wird das, in dem die Rezepte mittels *extract recipes for website.xsl* in ein JSON-Objekt wie untenstehend umgewandelt wird.

