Bluemix OpenWhisk Hands-on Session

Serverless- oder eventbasierte Programmierung steht innerhalb der IBM Bluemix Plattform unter dem Angebot “OpenWhisk” zur Verfügung. Mit diesem Programmiermodell ist es möglich, gezielt auf bestimmte Events, z.B. Einträge in einer Datenbank, zu reagieren und spezifisch eine bestimmte Funktion für dieses Event auszuführen. Hierbei wird nur die Laufzeit der ausgeführten Funktion abgerechnet, ohne dass dafür permanent eine Anwendung laufen muss.

Die folgende Kurzanleitung zeigt die wichtigsten Schritte auf, um im Browser-Editor mit OpenWhisk erste Actions, Trigger und Rules zu definieren.

Kurzanleitung

1. Im Browser die Bluemix Console in den USA öffnen

https://console.ng.bluemix.net/

und mit den eigenen Zugangsdaten anmelden.

2. Für den späteren Verlauf der Übungen benötigen wir zwei Services, die wir an dieser Stelle schon einmal anlegen wollen. Dazu bitte den “Katalog” anwählen.
   a) Erstellen eines Cloudant Services

In der Suchzeile des Kataloges “Cloudant” eingeben, den Service auswählen, auf der Detail-Seite des Services alle Voreinstellungen so belassen und auf “Create” klicken. Nachdem der Datenbank-Service erfolgreich erstellt wurde, erscheint im Browser eine Seite mit den Details zum Service. Dort bitte auf den “Launch”-Button rechtsklicken und das Management Dashboard von Cloudant in einem separaten Tab öffnen.
Im Cloudant Management Dashboar mit “Create database” eine Datenbank mit dem Namen “openwhisktest” erstellen.
Wenn die Datenbank erfolgreich erstellt ist, wieder in den Bluemix-Browser-Tab und danach mit dem “Katalog” Link zurück zum Katalog wechseln.

   b) Erstellen eines Watson Language Translator Services

In der Katalog-Ansicht “Translator” in die Suchzeile eingeben, um den Watson Language Translator zu finden. Den Service anwählen, in der Service-Seite alle Voreinstellungen so belassen und auf “Create” klicken. Wenn der Service erfolgreich erstellt worden ist, wieder mit dem Link “Katalog” auf die Katalog-Seite wechseln.

3. Auf der linken Seite “Openwhisk” als Compute Modell auswählen.
4. Die Detail-Seite mit dem Openwhisk Service öffnet sich. Diesen anklicken.
5. Auf der Startseite von Bluemix Openwhisk den Link “Develop in Browser” auswählen.
6. Der Openwhisk Browser Editor öffnet sich und sollte direkt den “Develop” Tab öffnen.
7. Auf dem “Develop” Tab sieht man auf der linken Seite die Übersicht der “Actions”, die schon definiert sind, es sollten genau zwei sein: “Hello World” und “Hello World With Params”. Ebenso ist die Liste der “Sequences” und die Liste der “Rules” zu sehen.
8. Von der Liste der vordefinierten Actions “Hello World With Params” auswählen, sodass im rechten oberen Editor-Fenster der hinterlegte Code erscheint.

Für Actions steht eine einzelne Variable für den Input zur Verfügung, die in dem Code Beispiele mit params deklariert wird.

```
/**
  *
```

```
  * main() will be invoked when you Run This Action.
  *
  * @param OpenWhisk actions accept a single parameter,
  *        which must be a JSON object.
  *
  * In this case, the params variable will look like:
  *     { "message": "xxxx" }
  *
  * @return which must be a JSON object.
  *         It will be the output of this action.
  *
  */
function main(params) {
      return { "message": "you sent me " + params.message };


}
```

Input Werte können mit einem JSON Dokument als Key:Value Paare übergeben werden.

9. Das Code Beispiel soll nun wie folgt abgeändert werden, um aus dem übergebenen Input den Namen und den Ort auszulesen. Bitte die “return”-Zeile komplett durch folgenden Teil ersetzen:

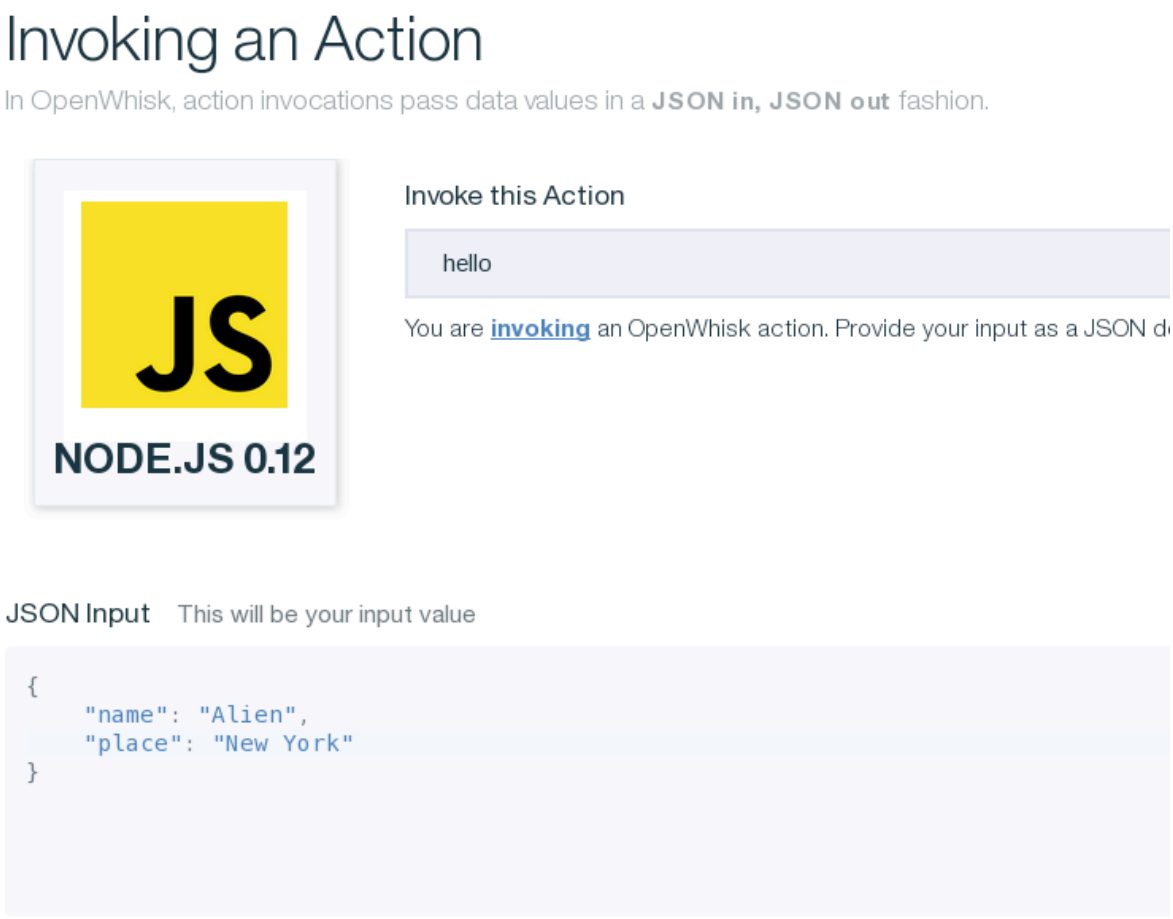
```
return { message: 'Hello, '+ params.name + ' from ' + params.place + '. It is a great pleasure to have you here.'};
```

10. Die Funktion erwartet ein JSON document mit den Schlüsseln “name” und “place”. Um die Action zu testen, geht man nun auf den “Make it live” Button rechts unten und speichert den veränderten Code im eigenen Arbeitsbereich. Anschliessend oben links auf den Button “Run this Action” klicken, um die Action zu testen.
11. Auf der folgenden Seite muss das JSON Document für den Test eingegeben werden. In dem erscheinenden Editor (“JSON Input”) das angezeigte JSON durch folgendes ersetzen:

```
{
   "name": "Alien",
    "place": "New York"
}
```

(oder eigene Werte für name und place vergeben).

Bsp.:



12. Die Eingabe mit dem Button “Run with this value” in der rechten unteren Ecke bestätigen.
13. Als Ergebnis sollte ein neues Fenster mit dem Consolen Log erscheinen, der ebenfalls wiederum ein JSON Format mit dem Schlüssel “message” und dem zusammengebauten Text ausgibt.

Bsp.:

In dem Konsolenlog sieht man auch, dass diese Action für insgesamt nur 751ms berechnet wird.

14. Von der Console kommt man wieder zurück zum Editor durch Drücken des Knopfes “Close” unten rechts.
15. Aktionen können verknüpft werden zu Ketten, sogenannten “sequences”. Dabei stehen die JSON-Objekte die eine Aktion zurückgibt, gleichzeitig als Input-Werte für die Folgeaktion

zur Verfügung. Neben selbstgeschriebenen Funktionen stellt OpenWhisk auch eine Vielzahl von vorgefertigten Actions zur Verfügung, die direkt mit den Bluemix Services arbeiten. Ein Beispiel dafür ist der Watson Language Service, mit dem wir im folgenden einen kleinen Text übersetzen wollen.

a) Erstellen Sie bitte eine neue Action mit dem Namen “translateText”. Dazu wählen Sie im Editor “Create Action”. Geben Sie den Namen ein und lassen Sie die voreingestellten Werte, wie sie sind.

b) Im Editor erscheint ein einfaches Gerüst Ihrer Function:

```
function main(params) {
    return { message: 'Hello World' };
}
```

Ersetzen Sie die Zeile, die mit “return ..” beginnt, durch die folgende Zeile:

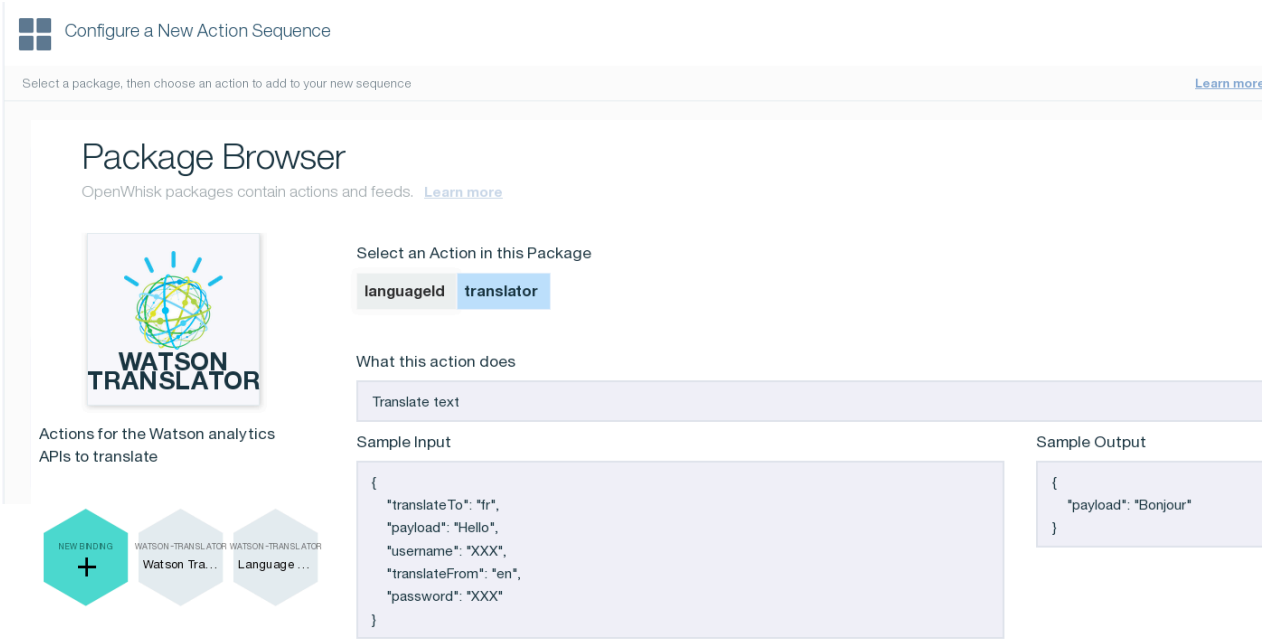
```
return ({ "translateTo": "de", "payload" : "Hello this is the message"});
```

So wird ein JSON Dokument direkt in dem Format ausgegeben, das wir für den nächsten Schritt, den Watson Language Translator, benötigen. Der Text, der zum Schlüssel “payload” gehört, wird in diesem Beispiel ins Deutsche übersetzt. Wenn Sie möchten, können sie auch ins Französische, “fr”, oder Spanische, “es” übersetzen lassen.

Wenn Sie die return Zeile angepasst haben, dann speichern Sie die Änderung mit dem “Make it live” Button unten rechts ab.

c) Im nächsten Schritt wird eine sequence erstellt. In der eben im Editor erstellten Action drücken Sie bitte den Button “Link into a Sequence” unten rechts.

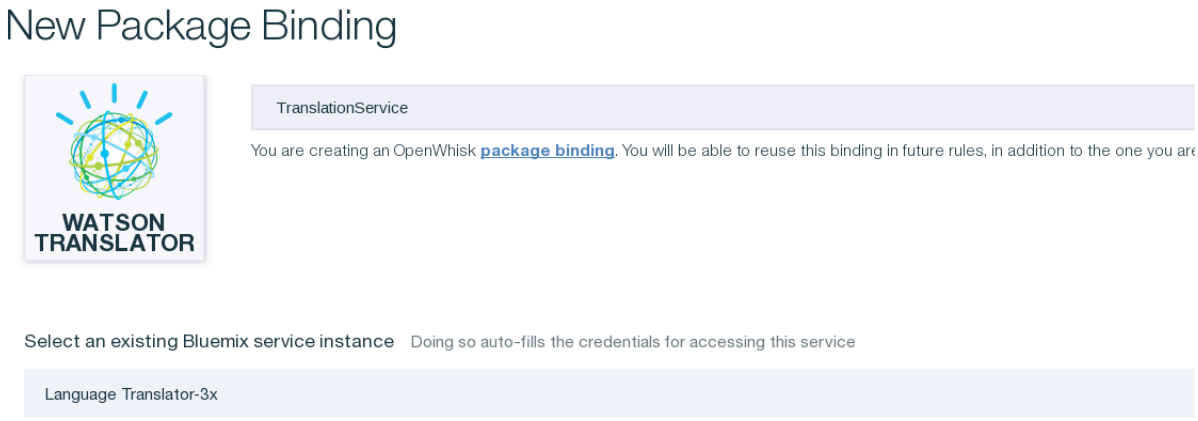
Im Folgenden können Sie aus den vorgefertigten Packages auswählen, um eine neue Sequence zu erstellen:



Wählen Sie das Package “Watson Translator” aus. Im folgenden Package Browser wählen Sie bitte die Action “translator” aus, damit wir einen übergebenen Text auch wirklich übersetzen können.

Nun müssen Sie die Action noch mit dem Watson Translator Service verbinden, ein sogenanntes “Binding” erstellen. Dazu klicken Sie bitte im Browser auf das grüne Sechseck “New Binding”:

Auf der folgenden Seite wählen Sie aus der Liste den Watson-Service aus, den Sie eben erstellt



haben und geben dem neue Binding einen aussagekräftigen Namen, z.B. “TranslationService”: Anschliessen klicken Sie unten rechts auf “Save Configuration” und gelangen wieder zurück zum Package Browser.
Um den Schritt abzuschliessen, ist es nun wichtig, das im Package Browser sowohl “translator” als auch das von Ihnen neu erstellte neue Binding “TranslationService” ausgewählt, sprich blau hinterlegt sind.

Diesen Schritt abschliessen durch drücken des Buttons “Add to Sequence”.

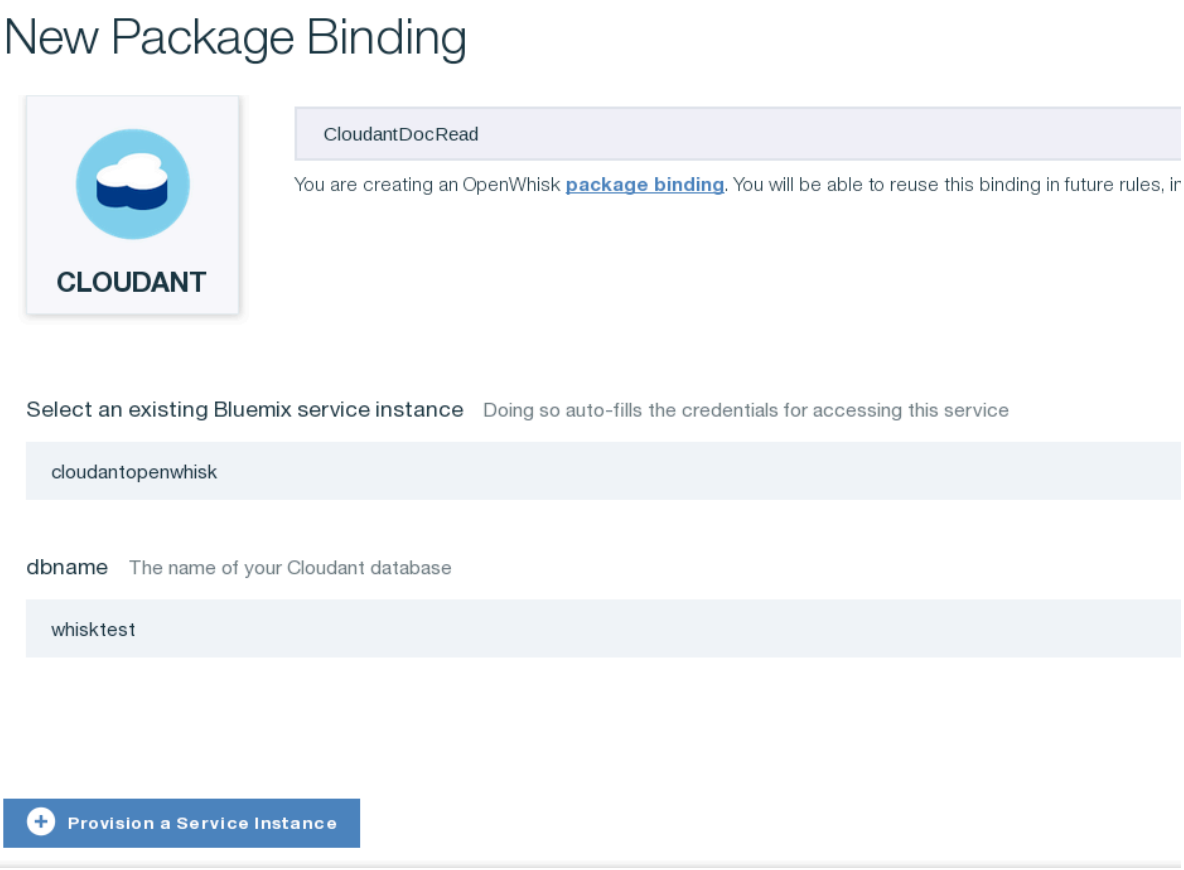
d) Im abschliessenden Schritt wird die Sequenz noch einmal dargestellt. Sie sollte aus zwei Schritten bestehen, die selbst erstellte Action zuerst und anschliessend die Watson Tanslator Action. Falls die Reihenfolge nicht stimmt, kann man die Action mit den Pfeil-Buttons nach oben oder unten verschieben. Um zum Speichern zu gelangen den Button “This looks good” rechts unten klicken und der sequence noch einen eindeutigen Namen

geben, z.B. “simpleTranslate” und den wizard mit Drücken auf den Button “Save Action sequence” rechts unten abschliessen.

e) Die Sequence erscheint im Bereich “My Sequences” auf der linken Seite. Wenn sie ausgewählt ist, kann sie mit “Run this Sequence” getestet werden.
f) Auch wenn der Input-Editor ein JSON Dokument erwartet, so ist das für unseren ersten Test nicht entscheidend, da wir hier nur eine im Code fest geschriebene Nachricht übersetzen. Deshalb direkt mit “Invoke ...” ausführen und Sie sollten die Übersetzung der Originalnachricht in der gewünschten Sprache sehen.

16. Die Idee hinter Serverless Programming ist, dass die Actions oder Sequences durch Trigger ausgelöst werden, dies kann, wie in unserem folgenden Beispiel, das Schreiben in eine Datenbank sein. Das heisst im Folgenden erstellen wir eine Rule: Immer wenn ein Document in einer Cloudant Datenbank erstellt oder verändert wird, startet die Regel die Action Sequence, die folgende Schritte umfasst:
    a) Cloudant Read Document – Action aus dem Cloudant Package das das Dokument mit der übergebenen ID liest und an das Dokument als JSON an die nächste Aktion liefert
    b) translateText – Action aus dem vorherigen Schritt, so modifiziert, dass der richtige Schlüssel an den Watson Translator übergebne wird.
    c) Watson Translator – Übersetzung des übergebenen Textes.

17. Der Trigger den wir später einführen liefert uns als Ergebnis die docID des veränderten Dokumentes in der Cloudant-Datenbank. Deshalb müssen wir die Schritt 15 erstellte Sequence nur noch mit der Aktion für das Lesen des Dokumentes erweitern. Dies geschieht mit einem vorgefertigten Action aus dem Cloudant Package.
    a) In der Liste “My Sequences” die eben erstellte Sequence “simpleTranslate” auswählen und im Editor unten rechts den “Extend” Button drücken.
    b) Der “Action Sequence Viewer” wird geöffnet und aus der Auswahl drücken Sie bitte auf die “Cloudant” Schaltfläche um den Package Browser des Cloudant Packages zu öffnen.
    c) Im Package Browser stehen Ihnen dann verschiedene vorgefertigte Actions zur Verfügung. Wählen Sie dort bitte die Action “Read document” an.
    d) Wie schon beim Watson Translator muss auch hier wieder ein “Binding” erstellt werden. Dazu bitte unten links auf “New Binding” klicken.
    e) Im “New Package Binding” Dialog geben Sie dem Binding einen Namen, z.B. “CloudantDocRead” und wählen aus der Dropdown-Liste die Cloudant Service Instanz, die Sie in Schritt 2 erstellt haben. Dabei sollte automatisch in einem weiteren Dropdown auch die Datenbank zu sehen sein die Sie erstellt haben, vergleichbar zu dem folgenden Screenshot:



Wenn die so eingestellten Werte korrekt sind, dann bitte mit “Save configuration” speichern.

f) Im Package Browser darauf achten das “Read document” und das neu erstellte Binding hellblau hinterlegt sind, erst dann auf “Add to sequence” klicken. Damit wird diese Action an das Ende eingefügt.

18. Die soeben angehängte Action muss nun in der Sequence an den Anfang der Sequenz geschoben werden, dazu die Action mit dem Pfeil Button nach oben verschieben.
19. Am Ende sollte die Sequence wie folgt aussehen:
    a) CloudantDocRead
    b) translateText
    c) TranslationService

20. Die vorgefertige Action, die wir in CloudantDocRead verwenden, stellt der Action translateText das ausgelesene Dokument als JSON zur Verfügung. Den Code von translateText müssen wir nun noch abändern, sodass der Wert für den richtigen Schlüssel and den Watson Translator übergeben wird.

Im Openwhisk Editor auf der linken Seite aus der Liste “My Actions” die Action “translateText” aufrufen, sodass der Code im Editor sichtbar ist.
Im anschliessenden Test werden wir in der Cloudant-Datenbank ein Dokument erstellen und dort einen Schlüssel “english” verwenden. Damit dieser Schlüssel richtig erkannt wird, bitte die return-Zeile durch folgenden Code ersetzen:

```
return ({ "translateTo": "de", "payload" : params.english});
```

Also insgesamt sollte dort stehen:

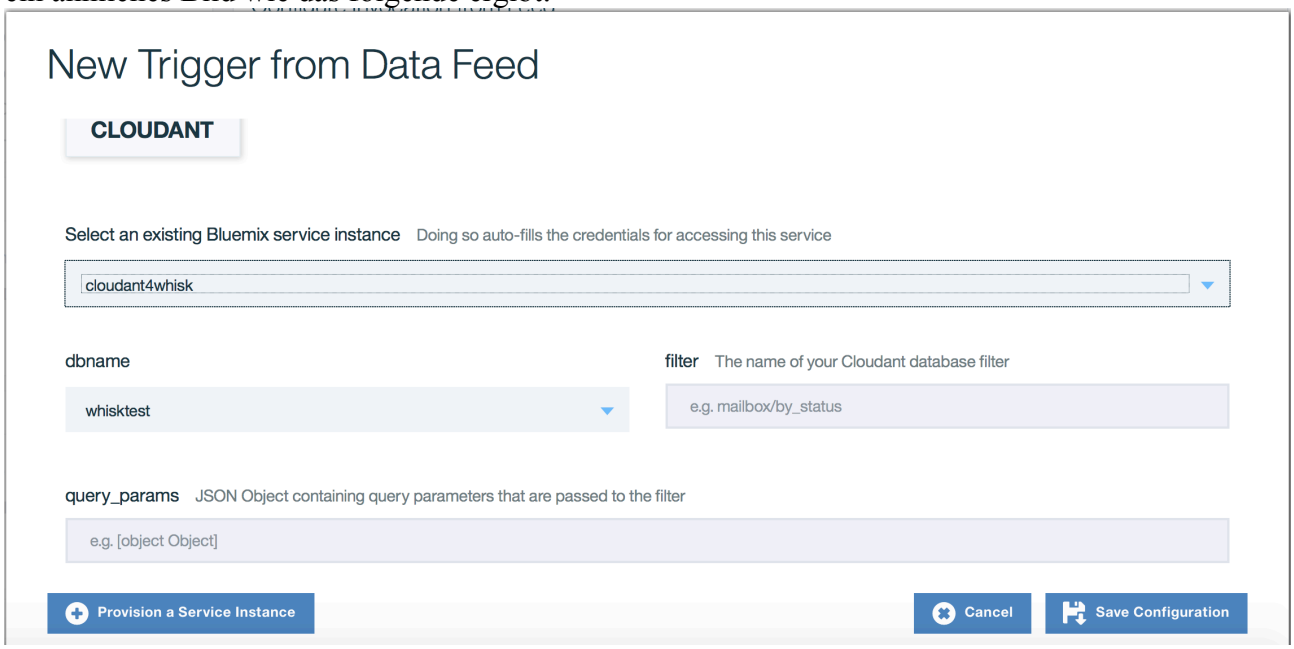
```
function main (params) {
return ({ "translateTo": "de", "payload" : params.english});
}
```

Wenn Sie fertig sind mit dem editieren der Aktion, abspeichern mit “Make it live” nicht vergessen!

21. Jetzt wo die Sequence fertig ist fehlt nur noch der Auslöser, der Trigger. Dazu aus der Liste

"My Sequences" die eben erstellte Sequence auswählen und unten rechts auf "Automate" klicken. Im folgenden Dialog die "Cloudant Changes" Schaltfläche anwählen, zum Konfigurieren auf welche Datenbankänderung reagiert werden soll.

22. Wählen Sie "New Trigger" durch klicken auf das grüne Sechseck. Im folgenden können Sie dem Trigger einen Namen geben, z.B. "translateEnglishDocs" und wählen aus der Dropdown-Liste die von Ihnen im Schritt 2 erstellte Datenbank. Wiederum wird automatisch im zweiten Dropdown die ebenfalls von Ihnen erstellte Datenbank eingestellt, sodass sich ein ähnliches Bild wie das folgende ergibt:

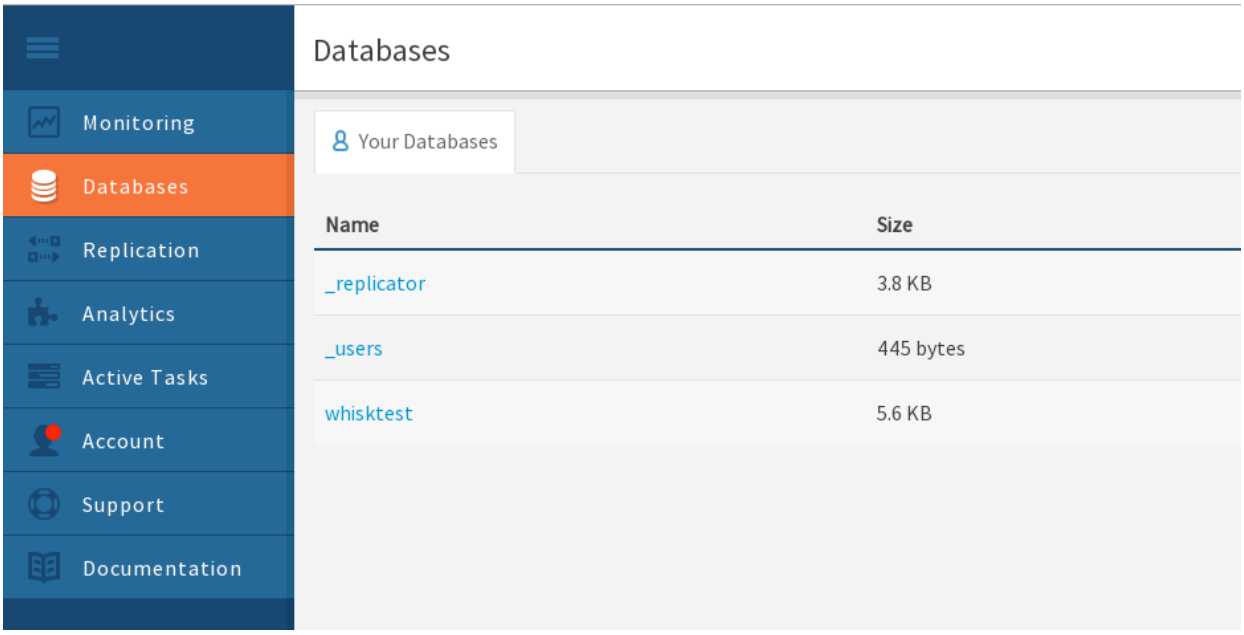


Es müssen nur Datenbank Instanz und Name ausgefüllt sein, alle anderen Einstellungen können leer bleiben. Wenn alle Einstellungen korrekt sind, bitte mit "Save configuration" abspeichern.

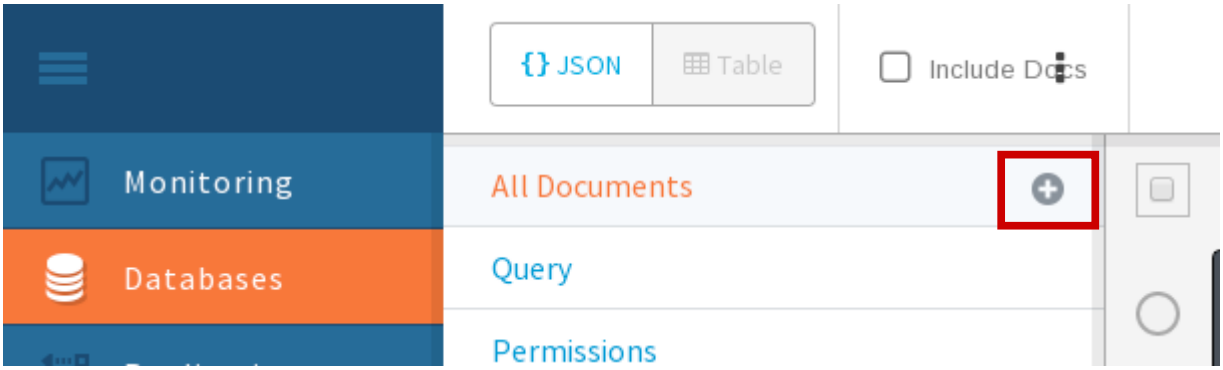
23. Das neue Binding wird als blau hinterlegtes Sechseck dargestellt. Klicken Sie auf Next.
24. Im nächsten Schritt sehen Sie wie dieser Trigger an den Anfang Ihrer Sequence gestellt wird, so daß daraus nun eine Rule geworden ist: "Wenn ein Dokument in der Datenbank geändert wird, dann starte die erste Action der Sequence."

Wenn Sie mit der Rule zufrieden sind, so gelangen Sie mit dem Button "This looks good" zum abschließenden Schritt, in dem Sie der Rule noch einen Namen geben können, bevor Sie den wizard mit "Save rule" abschliessen.

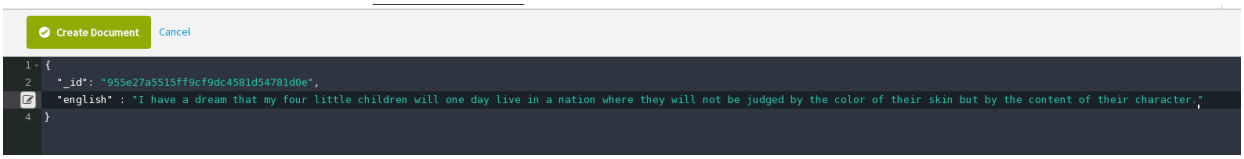
25. Mit dem Abspeichern der Rule ist diese auch gleichzeitig scharf geschaltet worden! Von diesem Moment an führt jede Veränderung eines Dokumentes in der Datenbank zum Auslösen des Triggers und zum Ausführen der Sequence.
26. Zum Testen wechseln Sie bitte zuerst von "Develop" Ansicht im OpenWhisk Editor auf die "Monitor"-Ansicht.
27. Anschliessend wechseln Sie bitte zum Cloudant-Dashboard, das Sie von Schritt 2 in einem anderen Browser Tab noch geöffnet haben sollten. Falls dies nicht der Fall ist, dann öffnen Sie bitte in einem neuen Browser Tab das Bluemix Dashboard, wählen Ihren Cloudant Service an und öffnen von dort über den "Launch"-Button das Cloudant Dashboard:



Wählen Sie die von Ihnen erstellte Datenbank aus. Nun können Sie in der Datenbank ein neues Dokument erstellen. Dazu klicken Sie bitte auf das +-Zeichen neben “All Documents” und anschliessend auf “New Doc”:



Im Editor wird ein JSON-Dokument mit einem id Schlüsserl erstellt. Fügen Sie hinter diesem Schlüssel noch den “english”-Schlüssel ein mit einem englischen Text, wie z.B. “I have a dream that my four little children will one day live in a nation where they will not be judged by the color of their skin but by the content of their character.”
So sollte das Ergebnis aussehen:



Mit dem Klick auf “Create Document” wird das neue Dokument in der Datenbank gespeichert und sollte für das Auslösen des Triggers sorgen.

28. Wechseln Sie zurück zu dem Browser Tab in dem der OpenWhisk Monitor zu sehen ist. Dort können sie in dem “Activity Log” oben rechts die Ausführung der Sequence verfolgen. Am Ende sollten Sie im Log die Ausgabe des übersetzten Textes sehen.