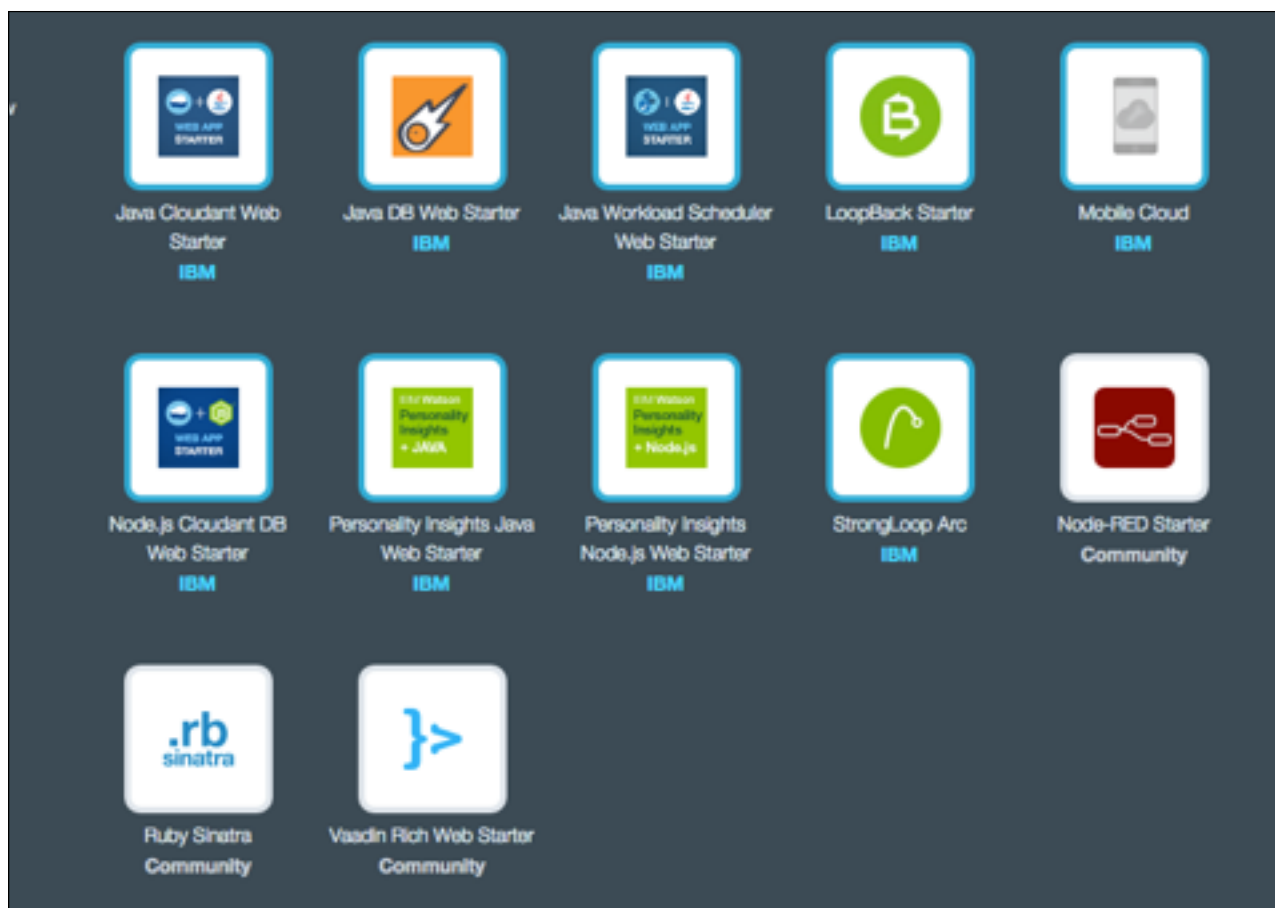


Hva mener Watson om dine instagram bilder?

En Node-Red oppgave...

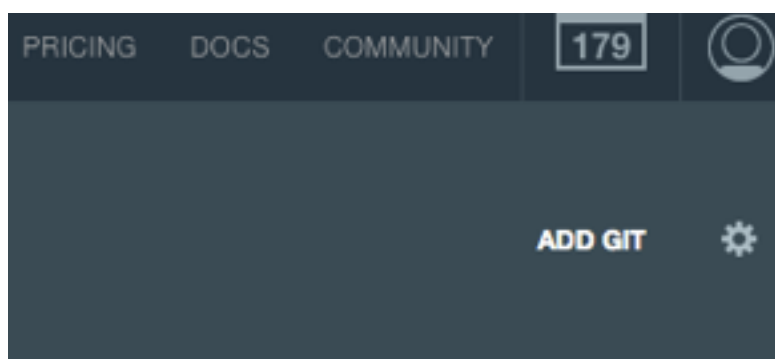
Oslo - 18. november 2015



I denne oppgaven skal vi bruke Node-Red, Instagram noden og Watson Visual Analyzer.

Det første du må gjøre er å deploye en Node-Red Starter i Bluemix - hvis du ikke allerede har en du vil bruke...

Når den er ferdig staget velg "add gitt" i oversikt siden.



Dette gir deg en mulighet til å linke prosjektet dir mot et git - og IBM DevOps services....

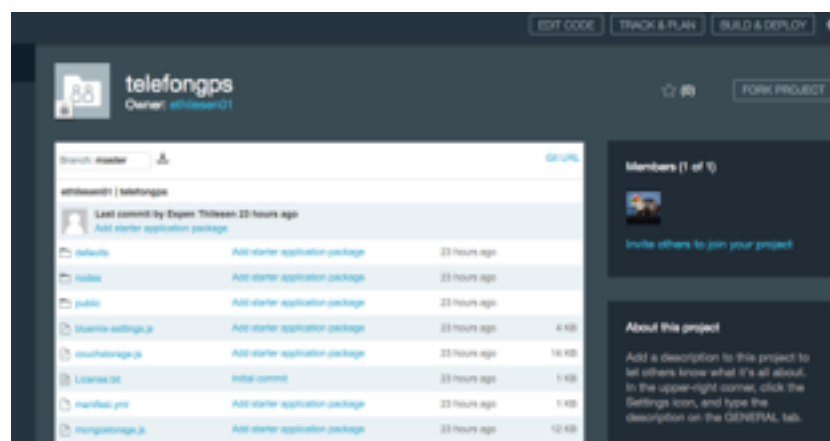
Du vil få opp en dialog boks hvor du må logge deg inn - med samme bruker / passord du bruker på Bluemix. Videre settes det opp et prosjekt i DevOps som du siden finner link til på samme sted.



Klikk på linken til ditt prosjekt...

Dette vi åpne en ny fane og bringe deg inn i prosjektet.

Noe slikt:



Trykk på EDIT CODE...

Det vi nå skal gjøre er å utvide Node-Red med en ny node. Det finnes mange noder til Node-Red som ikke er default installert i Bluemix, som du kan legge til ved behov.

Se <http://flows.nodered.org>

Søk etter Instagram og du finner en link til den noden..

node-red-node-instagram 0.1.0

Node-RED nodes that get photos from Instagram

 npm install node-red-node-instagram

Node-RED nodes that get photos from Instagram.

Install

Run the following command in the root directory of your Node-RED install

npm install node-red-node-instagram

Node Info

Version: 0.1.0
Updated 4 months, 1 week ago
License: Apache-2.0
[view on npm](#)

Downloads

11 in the last day
28 in the last week
179 in the last month

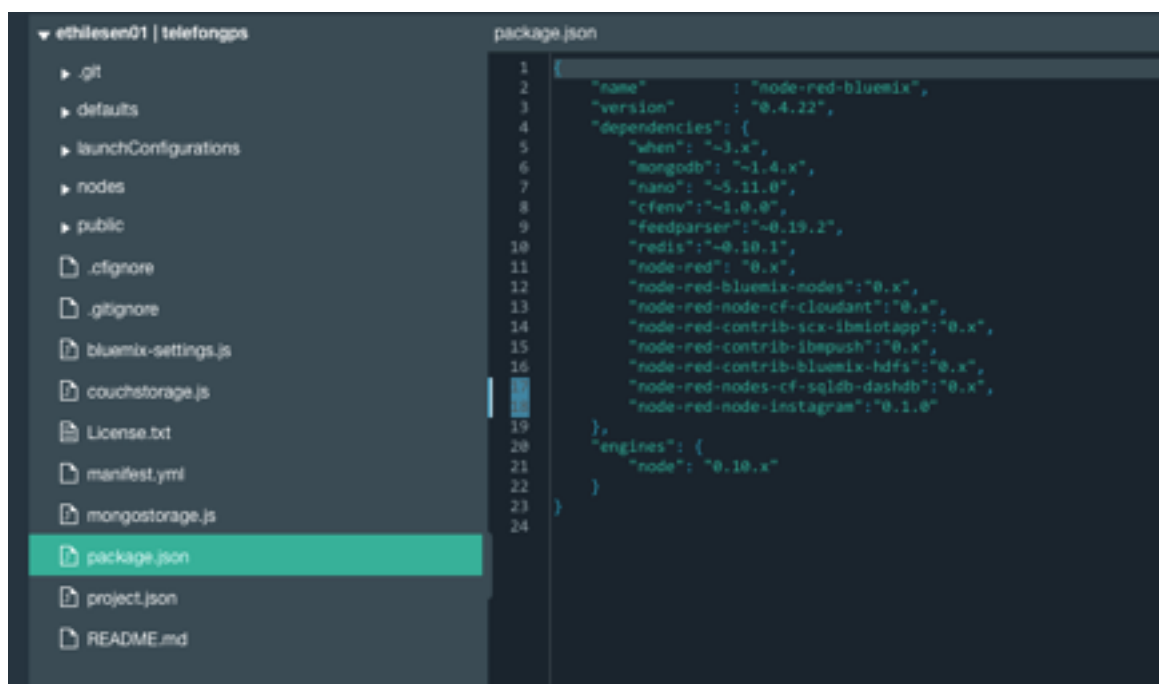
Nodes

instagram-credentials
instagram in

Merk deg navnet og versjonen - node-red-node-instagram - 0.1.0

Dette må vi nå legge inn i DevOps i en fil som heter package.json - en fil som beskriver dine dependencies..

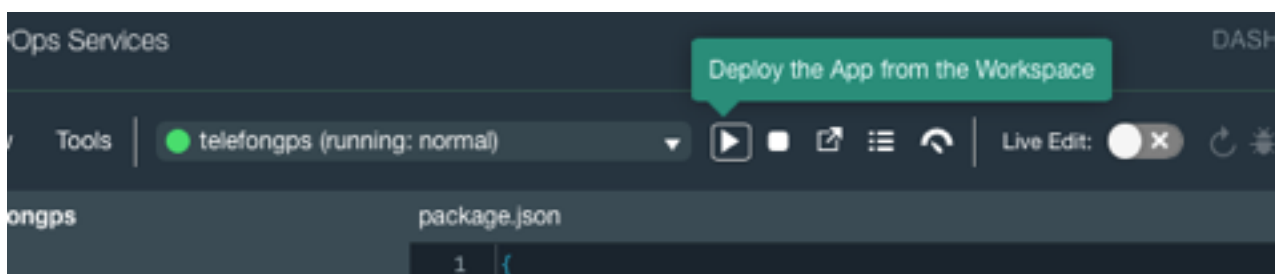
Legg inn "node-red-node-instagram":"0.1.0" i filen som vist under



```
1 {
2   "name": "node-red-bluebrix",
3   "version": "0.4.22",
4   "dependencies": {
5     "when": "~3.x",
6     "mongodb": "~1.4.x",
7     "nano": "~5.11.0",
8     "cfenv": "~1.0.0",
9     "feedparser": "~0.19.2",
10    "redis": "~0.10.1",
11    "node-red": "0.x",
12    "node-red-bluebrix-nodes": "0.x",
13    "node-red-node-cf-cloudant": "0.x",
14    "node-red-contrib-scx-ibmiotapp": "0.x",
15    "node-red-contrib-ibpush": "0.x",
16    "node-red-contrib-bluebrix-hdfs": "0.x",
17    "node-red-nodes-cf-sql&dash;dashdb": "0.x",
18    "node-red-node-instagram": "0.1.0"
19  },
20  "engines": {
21    "node": "0.10.x"
22  }
23 }
```

Husk å sette , på linjen over...

Nå du har lagret filen trykker du på knappen for å deploye endringene/applikasjonen til bluebrix.

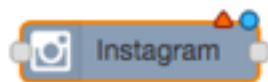


Når prosessen er ferdig kan du åpne urlen til Node-red prosjektet ditt og se om du kan finne en ny node under social:



Det skal være to - en som henter bilder hvert 15 minutt og en som henter på kommando.

Vi skal bruke den siste. Dra en slik ut på arbeidsflaten din.



Det neste vi må gjøre er å sette opp en kobling mot din Instagram konto..

Klikk på noden og du får opp dette:

Add new instagram-credentials config node Sheet 2

1. Create your own app at [instagram.com](https://www.instagram.com/developer/apps/)

2. Copy the app details here:

Client Id

Client Secret

Redirect URI

Instagram User

Klikk på linken til [instagram.com](https://www.instagram.com)

Registrer en ny app:

The screenshot shows the 'New App' form on the Instagram Developer Portal. It has two tabs: 'Details' (selected) and 'Security'. The form fields are as follows:

- Application Name:** MinApp
- Description:** bla bla...
- Company Name:** myCompAny.com
- Website URL:** myCompAny
- Valid redirect URIs:** http://dinNode-Red-server-url
- Privacy Policy URL:** (empty)
- Contact email:** (empty)

Below the form is a reCAPTCHA challenge with the text 'Type the words above:' and a button to 'Register'.

Du vil da
få en side
med

lignende informasjon:

MyNodeR		DELETE	RESET SECRET	EDIT
CLIENT INFO				
CLIENT ID	d8c12ef344b			
CLIENT SECRET	cd9df0681d5			
WEBSITE URL	http://telefon			
REDIRECT URI	http://telefon			
SUPPORT EMAIL	ethillesen@m			
node-red				

Denne informasjon legger du nå inn i instagram noden.

Add new instagram-credentials config node

Sheet 2

1. Create your own app at [instagram.com](https://www.instagram.com/developer/apps/)

Please configure Instagram to accept the following **Redirect URL** for authentication of your application:
`http://telefongps.eu-gb.mybluemix.net/red/instagram-credentials/auth/callback`

2. Copy the app details here:

Client Id

ethilesen@mac.com

Client Secret

Authenticate with Instagram

Add

Cancel

I dette eksempel ønsker vi bare linken til de bildene du trykker like på.

Edit instagram node

User

espenhilesen

Type

Liked photos

Output

the photo url

Name

Name

Ok

Cancel



Trykk på inject og se at du får en response tilbake fra instagram.

```
18.11.2015, 13.30.24 f5fc3b24.0a03c8  
msg.payload : string [121]  
https://scontent.cdninstagram.com/hphotos-  
xtf1/t51.2885-  
15/s640x640/sh0.08/e35/12269885_170758847
```

Da er vi klare for å sende dette videre til Watson. Watson er en kognitiv datamaskin - den samme som ble brukt til å vinne jeppardy i 2011. Mye har skjedd siden den dag - idag brukes Watson over hele verden - med mange ulike funksjoner og tjenester.

IBM tilbyr mange kognitive tjenester på Bluemix - katalogen finner dere en list over alle, med gode eksempler dere kan prøver videre når dere er ferdig med denne oppgaven. Mere infor finner dere også på Watson developer cloud.

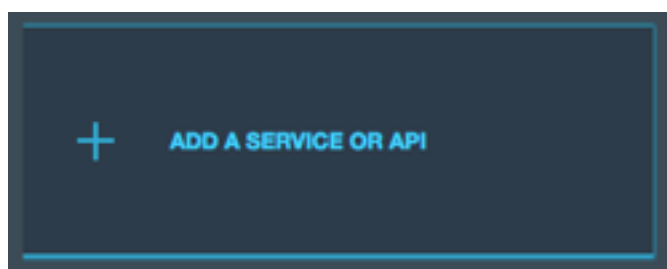
Vi skal bruke en Watson tjeneste som analysere bilder. I motsetning til andre tjenester dere bruker - som stort sett viser meta informasjon(tags) på bildene analyserer Watson pixler (selve bilde) ved hjelp av maskin lærende algoritme(r). Watson vil gi deg en liste over hva han/hun/den tror det er sammen med en sannsynlighets scoring.

Feks Annimal - score 0,872121 - betyr at den tror det er et dyr med 87% sikkerhet.

Denne Watson maskinen vi har tilgjengelig i Bluemix er ikke spesielt flink enda (den er under læring) men tenk likevel på hva dette kan brukes til hvis du har hundrevis av bilder du må prøve å finne litt system i. Feks hva har jeg av bilder - utendørs - innendørs - dyr - personer etc... De av dere som bruker FaceBook, Apple Bilder ol - vil se hva denne type teknologi vil bringe i fremtiden.

Gå tilbake til Bluemix dashbord og velg den Node-Red prosjektet ditt.

Legge til en ny tjeneste ved å klikke på denne:



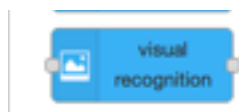
Velg under Watson - Visual Recognition



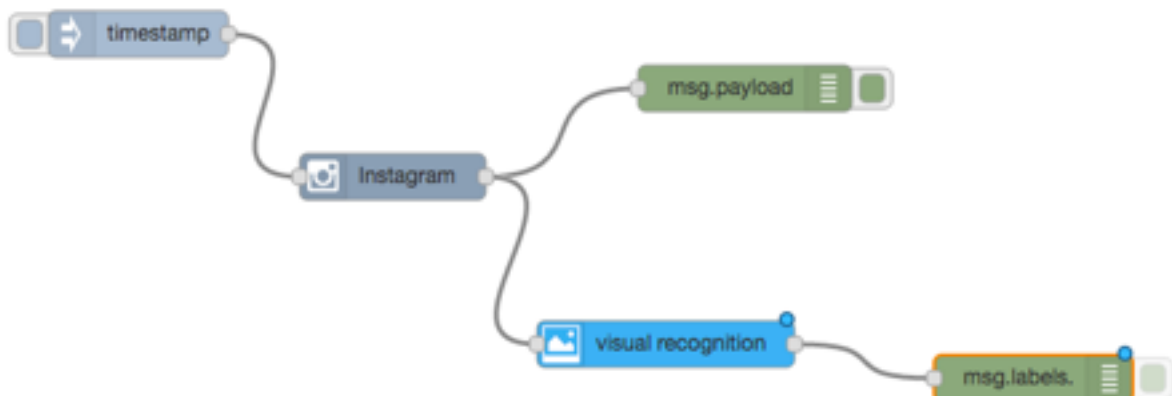
Velg ok for å re-stage applikasjonen din.

Gå så tilbake til Node-Red arbeidsflaten.

I node biblioteket ditt finner du Visual Recognition - dra denne ut på arbeidsflaten.



Koble sammen slik:



Endre den ene debug noded fra å logge msg.payload til msg.labels

Trykk på inject knappen og se hva du får i loggen !

Trolig noe slikt:

```
18.11.2015, 13.52.54  fa793a110567c8  
msg.labels : array [7]  
[ { "label_name": "Indoors", "label_score": "0.723468" }, {  
  "label_name": "Object", "label_score": "0.65725" }, {  
  "label_name": "Person", "label_score": "0.642142" }, {  
  "label_name": "Human", "label_score": "0.603131" }, {  
  "label_name": "Room", "label_score": "0.55694" }, {  
  "label_name": "People Activity", "label_score": "0.53638"  
}, { "label_name": "Rollerskating", "label_score":  
  "0.511304" } ]
```

Stemmer det ? -:)

Bruk instgram appen din - trykk liker på et annet bilde og se at det virker igjen...

Nå har du tatt i bruk kognitive tjenester så gratis!

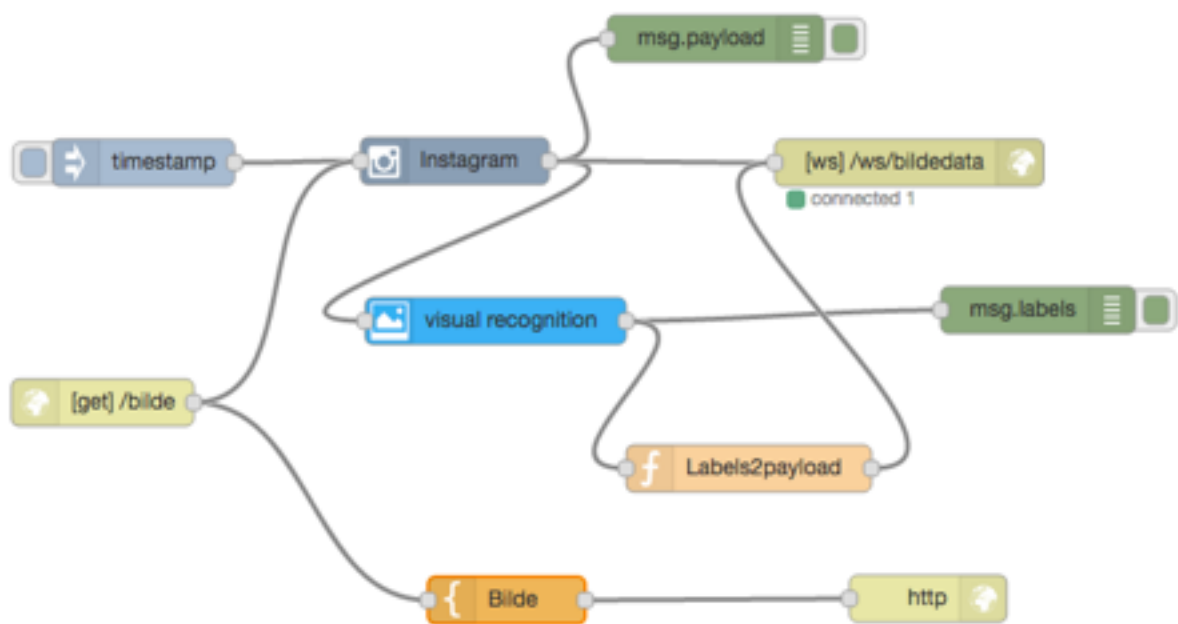
For de som har tid lyst - kan vi jo utvide dette slik at jeg har en webside som viser blide + resultat fra Watson... Dette klarer dere sikker på egenhånd. Hvis ikke er her en kjapp oppskrift...

Legg til en http input node, en template og en http response for å lage en webside.

Jeg tenkte bare å lage en websocket klient (som dere har gjort i tidligere oppgaver) og bare sende blide linken + Watson svare ut med ws.send.

Lables2payload bare endre msg objectet...

```
msg.payload = msg.labels;  
return msg;
```



Bilde siden: - her er det sote rom for forbedringer men det virker -:-)

<https://github.com/ethilesen/kodeklubben/blob/master/insta2watson.html>

```

<html>
<head>
  <title>Instagram2Watson</title>
</head>
<body>
  <div>
    <img id="bilde" src="">
  </div>
  <div>
    
    <p id="text"></p>
  </div>
  <script type="text/javascript">
    var wsUri = "ws://{{req.headers.host}}/ws/bildedata";
    var map;
    var ws;

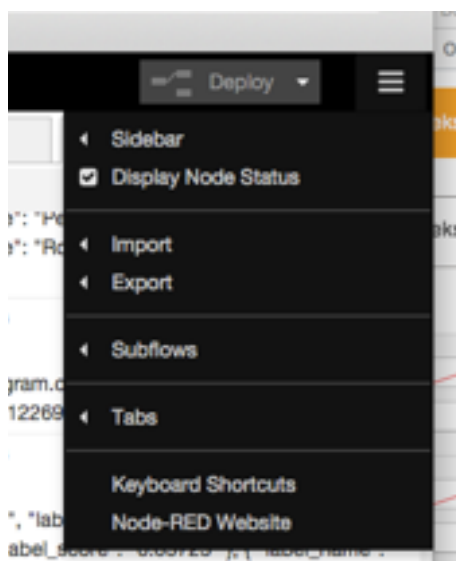
    ws = new WebSocket(wsUri);
    ws.onopen = function(){ console.log("Connected websocket");};
    ws.onerror = function(){ console.log("Websocket error"); };
    ws.onmessage = function(evt){
      console.log("Message recived!", evt.data);
      var str = evt.data;

      var n = str.search(/([http])/g); // her gjør jeg en test på om det er linken fra instagram eller scoring fra
      watson som kommer
      console.log("n= ", n);
      if (n<1){

        document.getElementById('bilde').src=(evt.data);
      }
      else{
        var arr = JSON.parse(evt.data);
        var ansvers = arr.length;
        var x = arr.map( function(item) {
          draw(item.label_name+" - score: "+item.label_score+'\n');
          function draw(items){
            var svar = document.createTextNode(items);
            document.getElementById("text").appendChild(svar);
          }
        });
        document.getElementById('spinner').src="";
      }
    }
  </script>
</body>
</html>

```

I Node-Red kan vi også eksportere / importere en hel flyt...



Her er flowen som json og du kan også importere denne:

```
{["id":"dd5e4674.22a1b8","type":"websocket-listener","z":"a6b734d.f5948c8","path":"/ws/bildedata","wholesmsg":"false"},"id":"a16c31da.5e93d","type":"inject","z":"a6b734d.f5948c8","name":"","topic":"","payload":"","payloadType":"date","repeat":"","crontab":"","once":false,"x":138,"y":340,"wires":[["b2a06524.4d5f98"]]],{"id":"f5fc3b24.0a03c8","type":"debug","z":"a6b734d.f5948c8","name":"","active":true,"console":"false","complete":"false","x":502,"y":261,"wires":[]},{"id":"b2a06524.4d5f98","type":"instagram","z":"a6b734d.f5948c8","instagram":"","inputType":"like","outputType":"link","name":"","x":336,"y":339,"wires":[["f5fc3b24.0a03c8","85689eb2.7a976","ff092cf4.00f6d"]]],{"id":"85689eb2.7a976","type":"watson-visual-recognition","z":"a6b734d.f5948c8","name":"","x":362,"y":441,"wires":[["fa783a1f.0587c8","12362158.edc9df"]]],{"id":"fa783a1f.0587c8","type":"debug","z":"a6b734d.f5948c8","name":"","active":true,"console":"false","complete":"labels","x":707,"y":434,"wires":[]},{"id":"64df67fb.9b2098","type":"template","z":"a6b734d.f5948c8","name":"Bilde","field":"payload","format":"handlebars","template":"<html>\n<head>\n\n <title>Instagram2Watson</title>\n</head>\n<nbody>\n\n <div>\n\n   <img id=\"bilde\" src=\"\">>\n\n </div>\n\n <div>\n\n <img id=\"spinner\" src=\"http://telefondps.eu-gb.mybluemix.net/images/watson.gif\">\n\n   <p id=\"text\"></p>\n\n </div>\n\n <script type=\"text/javascript\">\n\n   var wsUri = \"ws://{req.headers.host}/ws/bildedata\";\n\n   var map;\n\n   var ws;\n\n   \n\n   ws = new WebSocket(wsUri);\n\n   ws.onopen = function(){ console.log(\"Connected websocket\");};\n\n   ws.onerror = function(){ console.log(\"Websocket error\");};\n\n   ws.onmessage = function(evt){\n\n     console.log(\"Message received!\", evt.data);\n\n     var str = evt.data;\n\n     \n\n     var n = str.search(/([http])/g); // her gjør jeg en test på om det er linken fra instagram eller scoring fra watson som kommer\n\n     console.log(\"n= \", n);\n\n     if (n<1){\n\n       \n\n       document.getElementById('bilde').src=(evt.data);\n\n       }\n\n     else{\n\n       \n\n       var arr = JSON.parse(evt.data);\n\n       \n\n       var answers = arr.length;\n\n       \n\n       var x = arr.map( function(item) { \n\n         draw(item.label_name+\" - score: \" +item.label_score+'\\n');\n\n       });\n\n       function draw(items){\n\n         \n\n         var svar = document.createTextNode(items);\n\n         \n\n         document.getElementById(\"text\").appendChild(svar);\n\n         \n\n       });\n\n       \n\n       document.getElementById('spinner').src=\"\";\n\n       }\n\n     }\n\n   }\n\n </script>\n\n </body>\n\n </html>\",\"x\":369,\"y\":619,\"wires":[[\"1712c8ca.e8ed37\"]]],{"id":"961f58d2.69e0a8","type":"httpin","z":"a6b734d.f5948c8","name":"","url":"/bilde","method":"get","swaggerDoc":"","x":112,"y":493,"wires":[[\"64df67fb.9b2098\",\"b2a06524.4d5f98\"]]],{"id":"1712c8ca.e8ed37","type":"http response","z":"a6b734d.f5948c8","name":"","x":637,"y":618,"wires":[]},{"id":"ff092cf4.00f6d","type":"websocket output","z":"a6b734d.f5948c8","name":"","server":"dd5e4674.22a1b8","client":"","x":625,"y":340,"wires":[]},{"id":"12362158.edc9df","type":"function","z":"a6b734d.f5948c8","name":"Labels2payload","func":"msg.payload = msg.labels;\\nreturn msg;","outputs":1,"noerr":0,"x":523,"y":535,"wires":[[\"ff092cf4.00f6d\"]]]
```