IBM WATSON CLOUD API

FIRST CONTACT AND API PROMENADE

Alexandre Quemy - aquemy@pl.ibm.com
April 5, 2016

IBM Analytics
IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services
IBM WATSON
What is Watson?

Difficult to answer at first sight.
What is Watson?

It is an *ecosystem*:
When a researcher or engineer develop a cool feature, it becomes part of Watson ecosystem.
WHAT IS WATSON?

Watson Equations

- AI methods + NLP = Watson Core
- Watson Core + (REST) API + Bluemix = Watson Cloud Services
- WCS + Engineer imagination = Cognitive era
IBM WATSON CLOUD SERVICES
IBM Watson

IBM Watson Cloud Services

Generalities - Development - Deployment

Programming models

A note on Bluemix

A note on AlchemyAPI

Available Services

Focus on some services
Generalities

- Each service provides a REST API.
- Some services provide additional interfaces (e.g. Websocket).
- Bluemix Cloud Service to deploy applications.
- Two privileged languages: Java and Node.js.
- But also Python and Android SDK in beta.
Development Tools

- Using the Bluemix web interface.
- Using the Eclipse IDE (dedicated plugin).
- Using the Cloud Foundry CLI (PaaS by VMWare).
IBM Watson

IBM Watson Cloud Services

Generalities - Development - Deployment

Programming models

A note on Bluemix

A note on AlchemyAPI

Available Services

Focus on some services
## Programming models

- **Server-side proxy.**
  1. Relies only on credentials.
  2. Data stored by Bluemix.
  3. Safer by default, even on HTTP.
  4. Higher latency and lower performances.

- **Direct interaction with a service.**
  1. Relies on credentials + tokens.
  2. Data stored only at client-side.
  3. One hour lifetime token.
  4. XSS vulnerable. Be careful!
IBM Watson

IBM Watson Cloud Services

Generalities - Development - Deployment

Programming models

A note on Bluemix

A note on AlchemyAPI

Available Services

Focus on some services
<table>
<thead>
<tr>
<th>Instant instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Support Java, Javascript, Go, Python, PHP, Ruby.</td>
</tr>
<tr>
<td>• Free for 30 days, up to 2Go.</td>
</tr>
<tr>
<td>• All the time free with restrictions:</td>
</tr>
<tr>
<td>1. 1 concurrent instance.</td>
</tr>
<tr>
<td>2. up to 512Mo.</td>
</tr>
<tr>
<td>3. &lt; 100 clients.</td>
</tr>
<tr>
<td>4. &lt; 300 API requests per second.</td>
</tr>
</tbody>
</table>
Containers

All the time free with restrictions:

1. 4 containers.
2. 20 Go external storage.
3. 2 public IP addresses.
4. 128Mo RAM per container.
IBM Watson

IBM Watson Cloud Services

Generalities - Development - Deployment

Programming models

A note on Bluemix

A note on AlchemyAPI

Available Services

Focus on some services
<table>
<thead>
<tr>
<th>AlchemyAPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Part of IBM Watson Services since march 2015.</td>
</tr>
<tr>
<td>• Very rich API: would require a lecture just for one service!</td>
</tr>
<tr>
<td>• <a href="http://alchemyapi.com/products/demo">http://alchemyapi.com/products/demo</a></td>
</tr>
</tbody>
</table>
To click:

- Mixing several Watson services: The API Mashup Guide.
- More about Deep Learning: deeplearning.net.
- Processing Engine: Spark.
- Data visualization: Tableau.
- State of art in ML: TAO Inria Publications.
- LuaJIT over C/CUDA: Torch.
- Internet of Things the easy way: crossbar.io and autobahn.
### Upcoming 2016 Sessions

- Social Media Monitoring Enhanced with Watson APIs - April 20
- Social Media Monitoring APIs in a BI Dashboard - May 4
- Enhancing the Results of Your BI Dashboard - May 18
- How To Partner with the Watson Ecosystem - June 1
- Social Media Monitoring in Practice - June 15

More info: [http://www.pages03.net/ibmwatson/building-with-watson-web-series](http://www.pages03.net/ibmwatson/building-with-watson-web-series)
AVAILABLE SERVICES
IBM Watson

IBM Watson Cloud Services

Available Services

Services presentation: Language

Services presentation: Speech

Services presentation: Vision

Services presentation: Data Insights

Focus on some services
<table>
<thead>
<tr>
<th>AlchemyLanguage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection of API for Natural Language Processing:</td>
</tr>
<tr>
<td>• Entity Extraction</td>
</tr>
<tr>
<td>• Sentiment Analysis</td>
</tr>
<tr>
<td>• Keyword Extraction</td>
</tr>
<tr>
<td>• Concept Tagging</td>
</tr>
<tr>
<td>• Relation Extraction</td>
</tr>
<tr>
<td>• Taxonomy Classification</td>
</tr>
<tr>
<td>• Author Extraction</td>
</tr>
<tr>
<td>• Language Detection</td>
</tr>
<tr>
<td>• Text Extraction</td>
</tr>
<tr>
<td>• ...</td>
</tr>
</tbody>
</table>
### Concept Expansion

Analyze text to create dictionary of concepts and related words.

### Usage example

Creating cross-domain dictionary to understand different words used for the same thing.
CONCEPT INSIGHT
**Concept Insights**

- Analyze a text input to extract concepts and propose related content from different sources.
- Generate concept graphs or use the default one (Wikipedia).

**Usage example**

Automatically provide further readings for articles, targeted advertisement, automatic content tagging.
I’m Watson! I can help you order a pizza. What size?

Great! Can I get a medium?

Perfect. What toppings are you in the mood for today?
Dialog

- Creation of natural language dialogues.
- Include profile tracking for learning.

Usage example

Ordering online, first line online support, intelligent home automation.
**Document Conversion**

- Convert document in another format.
- Mainly use to prepare document for other services.
مرحبا

Hello.
Language Translation

Based on Statistical Machine Translation developed by IBM:

- Bayesian approach, with maximum likelihood estimation.
- Different from Ruled-Based Machine Translation.
- Different from Example-based Machine Translation.
**Natural Language Classifier**

- Deep learning classification for Natural Language queries.
- Provide classification with confidence interval.

**Usage example**

Automatic tagging of content, classification of emails, intelligent dialogues.
Personality Insights

- Personality (Five Factors Model), needs and values extraction.
- Needs at least 3.5k words.
- Any source: twitter, book, ...

Usage example

Individual customer service or marketing, job applicant analysis, market analysis.
IBM Watson competed and
won against two human
opponents on Jeopardy!
in February 2011.
### Relationship Extraction

- Linguistic analysis combined with Machine Learning.
- Extraction of entities in non-structured documents.

### Usage example

Trend change detection, semantic search engine.
Retrieve and Rank

- Find the most relevant documents to answer a question.
- Combination of search engine and Machine Learning.

Usage example

Intelligent diagnosis system.
<table>
<thead>
<tr>
<th>Tone Analyzer</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Analyze emotion and language tones in text.</td>
</tr>
<tr>
<td>• Provide Emotion, Language and Social analysis.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usage example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study on the impact of your announcements.</td>
</tr>
</tbody>
</table>
IBM Watson

IBM Watson Cloud Services

Available Services

Services presentation: Language

Services presentation: Speech

Services presentation: Vision

Services presentation: Data Insights

Focus on some services
Hello, Watson!
<table>
<thead>
<tr>
<th>Speech to Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>• IBM speech recognition to text output.</td>
</tr>
<tr>
<td>• Stream or recorded input.</td>
</tr>
<tr>
<td>• Several languages supported.</td>
</tr>
<tr>
<td>• Keyword extraction.</td>
</tr>
</tbody>
</table>

**Usage example**

Accessibility, transcription of conferences or meetings.
What can I do for you today?
## Text to Speech

- Low latency audio synthesis.
- Several languages supported.
- Specific words pronunciation tuning through the REST API.

### Usage example

Accessibility.
AVAILABLE SERVICES

IBM Watson

IBM Watson Cloud Services

Available Services

- Services presentation: Language
- Services presentation: Speech
- Services presentation: Vision
- Services presentation: Data Insights

Focus on some services
AlchemyVision

- Face, words, persons detection.
- Age estimation, tagging and classification.
- Similar content suggestions.

Usage example

Targeted add, help for criminal recognition, image search engine.
SATURN
**Visual Recognition**

- Deep-learning methods to classify images.
- Output tags with confidence interval.

**Usage example**

Automatic organization of images, social media segmentation.
### Visual Insights

- Deep-learning methods to describe **collections** of images.
- Returns tagging profile (weighted tags).
- Tags are clustered into categories.

### Usage example

Automatic organization of images, social media segmentation.
IBM Watson

IBM Watson Cloud Services

Available Services

- Services presentation: Language
- Services presentation: Speech
- Services presentation: Vision
- Services presentation: Data Insights

Focus on some services
AlchemyData News

- NL query over news and blog articles.
- Over 250k articles indexed.

Usage example

Automatic content suggestions, trend analysis.
TRADEOFF ANALYTICS
## Tradeoff Analytics

- Provide the set of best compromises in multiobjective optimization decision problem.
- Simple Pareto Set extraction with nice interface.

## Usage example

Selection of medical treatment, sorting candidates, investment analysis.
FOCUS ON SOME SERVICES
FOCUS ON SOME SERVICES

IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

A more complex use case

Speech to text

Natural Language Classifier

Dialog

A prototype
Let’s motivate our study...
A MORE COMPLEX USE CASE

Home Assistant aka Jarvis (or almost)

1. Speech to text to get input from persons in a room.
2. Natural Language Classifier to determine the user.
3. Dialog to take order and trigger actions (turn on or off the light, put some music on, ecosystem).
4. Text to speech to communicate to the user.
5. Tone analyzer to make suggestions of music.
IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

A more complex use case

Speech to text

Natural Language Classifier

Dialog

A prototype
## Overview

1. Transcription continuously returned.
2. Retroactively updated with new text.
3. Trained by usage.
4. REST or Websocket interface.
5. Two beta mobile SDK for Android and iOS.

---

**Overview**

Websocket prefered over http protocol.
## Input configuration

1. **Language**: language to transcript.
2. **Model**: broadband (16kHz) or narrowband (8kHz).
3. **Format**: FLAC, PCM, WAV, Ogg (opus codec).
4. **Transmission**: one-shot (4MB size max), streaming (max 100MB)

### Supported languages:
- Brazilian Portuguese
- Japanese
- Mandarin
- Arabic
- Spanish
- UK and US English
### Output configuration

1. Keyword spotting with user-defined confidence level.
2. Max alternative and interims.
3. Profanity filtering.
4. Word alternatives, confidence and timestamps.
General advices

1. Use nice hardware for sound capture.
2. Boardband model for real time transcription.
3. Narrowband for sources that filter the signal (phones).
4. Tweaking is the solution to overcome sensitivity.
curl -u aquemy:yoloPassword -X POST
   --header "Content-Type: audio/flac"
   --header "Transfer-Encoding: chunked"
   --data-binary @my_record_to_transcript.flac
   "https://stream.watsonplatform.net/speech-to-text/api/v1/recognize?continuous=true"
{ "results": [ { "alternatives": [ { "confidence": 0.8691191673278809, "transcript": "this is a test" } ], "final": true } ], "result_index": 0 }
Why Websocket?

Single-socket and full-duplex communication:

1. Simpler and more powerful than REST over HTTP.
2. Lower latency, faster (full-duplex).
3. Lighter and lower bandwidth consumption (single socket).
4. Streaming directly from web browser (part of HTML5).
**Speech to Text: WebSocket Interface**

Single (secure) end-point:

wss://stream.watsonplatform.net/speech-to-text/api/v1/recognize

```javascript
var token = '<authentication-token>';
var wsURI = '<endpoint>?watson-token=' + token + '&model=es-ES_BroadbandModel';
var websocket = new WebSocket(wsURI);
websocket.onopen = function(evt) { onOpen(evt) };
websocket.onclose = function(evt) { onClose(evt) };
websocket.onmessage = function(evt) { onMessage(evt) };
websocket.onerror = function(evt) { onError(evt) };
```
### Minimal initialization

- action must be start
- content-type must be in:
  1. audio/flac
  2. audio/l16
  3. audio/wav
  4. audio/ogg;codecs=opus
var message = {
    'action': 'start',
    'content-type': 'audio/l16;rate=22050',
    'continuous': True, # Do not stop after 0.5s
    'inactivity_timeout': -1 # Unlimited
};
socket.send(blob);

```javascript
function onMessage(evt) {
    console.log('recognition result in json format : ' + evt.data);
}
```

Remarks:

- Binary data only.
- No need to wait for the answer of 'start'.
"results": [
...
"keywords_result": {
  "coffee": [
  {
    "normalized_text": "coffee",
    "start_time": 0.76,
    "confidence": 0.984,
    "end_time": 1.18,
  },
  ...
  ],
  ...
},
]
```json
{
    "results": [
        {
            "alternatives": [
                {
                    "confidence": 0.8691263198852539,
                    "transcript": "the first transcript"
                },
                {
                    "transcript": "the second transcript"
                }
            ],
        }
    ],
}
```
{
  "results": [ 
    "word_alternatives": [ 
      {
        "start_time": 0.76,
        "alternatives": [ 
          {
            "confidence": 0.9837,
            "word": "Chuck"
          }
        ],
        "end_time": 1.18,
      },
    ]
  ]
}
{ "results": [ "alternatives": [ { "timestamps": [ ["several", 1.0, 1.51], ["tornadoes", 1.51, 2.15], ["touch", 2.15, 2.5], ... ] } }, ... ] }
Stop the service:

```javascript
socket.send(null);
```

Or preferably:

```javascript
var message = {"action": "stop"}
websocket.send(JSON.stringify(message));
```

Close the connection:

```javascript
websocket.close();
```
FOCUS ON SOME SERVICES

IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

A more complex use case
Speech to text
Natural Language Classifier
Dialog
A prototype
curl -u aquemy:yoloPassword -X POST
  -F training_data=@data_train.csv
  -F training_metadata="{"language":"en","name":"TutorialClassifier"}"
  "https://gateway.watsonplatform.net/
natural-language-classifier/api/
v1/classifiers"
Answer:

```json
{
  "name": "TutorialClassifier",
  "language": "en",
  "status": "Training",
  "url": "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/10D41B-nlc-1",
  "classifier_id": "10D41B-nlc-1",
  "created": "2015-05-28T18:01:57.393Z",
  "status_description": "The classifier instance is in its training phase, not yet ready to accept classify requests"
}
```
curl -u aquemy:yoloPassword -X POST "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/<classifier_id>/classify" --data-urlencode "text=How hot will it be today?"
Answer:

```json
{
    "classifier_id": "10D41B-nlc-1",
    "url": "https://gateway.watsonplatform.net/natural-language-classifier/api/v1",
    "text": "How hot will it be today?",
    "top_class": "temperature",
    "classes": [
        {
            "class_name": "temperature",
            "confidence": 0.9998201258549781
        },
        {
            "class_name": "conditions",
            "confidence": 0.00017987414502176904
        }
    ]
}
```
NATURAL LANGUAGE CLASSIFIER: USING YOUR OWN DATA

Remarks:

- CSV format, UTF-8, escaped, double quotation marks.
- Minimum of 5 records, maximum of 15000 records.
- Maximum of 1024 characters per records.
- Supported languages: en, ar, fr, it, pt, es.
**Guidelines:**

- Avoid record with more than 60 characters.
- Avoid to use more than 100 classes.
- Do not use less than 510 records per class.
- Avoid to use more than 3 classes per record.

**For you interest**

You can use NLC Toolkit to import JSON and manage your datasets.
GET /v1/classifiers

```
{
  "classifiers": [
    {
      "classifier_id": "10D41B-nlc-1",
      "url": "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/10D41B-nlc-1",
      "name": "My Classifier",
      "language": "en",
      "created": "2015-05-28T18:01:57.393Z"
    }
  ]
}
```

Alternatively: /v1/classifiers/<classifier_id>
POST /v1/classifiers - REQUEST

{
    language: 'en',
    name: 'My Classifier',
    training_data: '<csv>', # See Node or Java example
    training_metadata: '<csv>' # Idem
}
POST /v1/classifiers - OK ANSWER

```json
{
    "classifier_id": "10D41B-nlc-1",
    "name": "My Classifier",
    "language": "en",
    "created": "2015-05-28T18:01:57.393Z",
    "url": "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/10D41B-nlc-1",
    "status": "Training",
    "status_description": "The classifier instance is in its training phase, not yet ready to accept classify requests"
}
```
GET /v1/classifiers/<classifier_id>/classify - OK ANSWER

```json
{
    "classifier_id": "10D41B-nlc-1",
    "url": "https://gateway.watsonplatform.net/natural-language-classifier/api/v1/classifiers/10D41B-nlc-1/classify?text=How%20hot%20will%20it%20be%20today?",
    "text": "How hot will it be today?",
    "top_class": "temperature",
    "classes": [
        {
            "class_name": "temperature",
            "confidence": 0.9998201258549781
        },
        ...
    ]
}
```
POST /v1/classifiers/<classifier_id>/classify - REQUEST

```
{
  "classifier_id": "10D41B-nlc-1",
  "text": "How hot will it be today?",
}
```

For the answer, see GET on the same endpoint.
DELETE /v1/classifiers/<classifier_id> - REQUEST

```json
{
    "classifier_id": "10D41B-nlc-1",
}
```
IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

A more complex use case
Speech to text
Natural Language Classifier
Dialog
A prototype
Using the Dialog Service

Prepare data
- Gather representative questions
- Understand use case
- Understand business processes

Design dialog flows
- Translate business processes to Dialog models
- Design personality
- Review potential for Watson services (NLC, STT, TTS, etc.)

Push live to users
- Connect to User Interface and publish publicly
- Monitor conversations and review activity

Iterate, optimize and add content
- Learn from real interactions
- Adjust existing content to suit user needs
- Add new content based on user activity
DIALOG: CREATING THE XML

General information

- Description in XML.
- Really rich API but also complex.

```xml
<dialog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="DialogDocFormat_v0.8_manual.xsd">
  <flow>
    ...
  </flow>
  <entities></entities>
  <constants></constants>
  <variables></variables>
</dialog>
```

Let’s start by the end...
Variables

• Store user’s inputs to be passed to the backend.
• Separated by categories in ‘var_folder’.

```xml
<variables>
  <var_folder name="Home">
    <var_folder name="Phones" type="VAR">
      <var name="Color" type="TEXT" description="Phone color preference"/>
      <var name="Storage" type="TEXT" description="Storage size preference"/>
    </var_folder>
  </var_folder>
</variables>
```
<constants>
  <var_folder>
    <var name="VA" type="TAG">Virtual Agent</var>
  </var_folder>
</constants>
<settings>
  <setting name="DISPLAYNAME" type="USER">test</setting>
  <setting name="LANGUAGE" type="USER">EN</setting>
  <setting name="RESPONSETIME" type="USER">-2</setting>
  <setting name="CONCEPTMATCHING" type="USER">0</setting>
</settings>

Exhaustive list in API documentation!
Entities

Used to define enumerations.

```xml
<entity name="COLOR" entityType=GENERIC>
  <value name="Black" value="Black" />
  <value name="White" value="White" />
  <value name="Silver" value="Silver" />
</entity>
```

General information

- LOCATION, DATE_TIME_RANGE
- CONDITION
- AMOUNT, CURRENCY_AMOUNT, PERCENTAGE_AMOUNT
- PERSONA_NAME, PHONE_NUMBER, CREDIT_CARD_NUMBER
The folders

Group similar content. First to be defined:

- Main: entry point.
- Library: dialog structure.
- Global: global functions or nodes.
- Concepts: local concept of a dialog.
<dialog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="WatsonDialogDocument_1.0.xsd">
  <flow>
    <folder label="Main"/>
    <folder label="Library"/>
    <folder label="Global"/>
    <folder label="Concepts"/>
  </flow>

  ...

</dialog>
Output

Contains the response to user’s input.

```xml
<folder label="Main">
  <output>
    <prompt selectionType="RANDOM">
      <item>Hello, welcome to Mike’s Pizza</item>
    </prompt>
    <goto ref="getUserInput_2449614"/>
  </output>
</folder>
```
<folder label="Main">
  <output>...</output>
  <getUserInput id="getUserInput_2449614">
    <search ref="folder_2449611"/>
    <default>
      <output>
        <prompt selectionType="RANDOM">
          <item>
            I am sorry, I did not understand your question. Please try asking another one.
          </item>
        </prompt>
        </output>
    </default>
  </getUserInput>
</folder>
<folder label="Library">
  <folder label="Live Content" id="folder_2449611">
    <input>
      <grammar>
        <item>What type of toppings do you have?</item>
        <item>$ What type of toppings do you have?</item>
        <item>$ list of toppings</item>
        <item>What * toppings * have</item>
      </grammar>
    </input>
    <output>
      <prompt selectionType="RANDOM">
        <item>We have Pepperoni, Mushrooms, and Sausage</item>
      </prompt>
    </output>
  </folder>
</folder>
Search
Contains the response to user’s input.

```xml
<output>
  <prompt>
    <item>Hi, how can I help you?</item>
  </prompt>

  <getUserInput>
    <search ref="about"></search>
  </getUserInput>

</output>

</folder>

  <folder label="Library">
    <folder label="About insurance" id="about"></folder>
  </folder>
```
Concept

Store some variations around a single word.

```
<concept id="123456">
  <grammar>
    <item>Hello</item>
    <item>Hi</item>
    <item>Hola</item>
    <item>Hey</item>
  </grammar>
</concept>
```

Tip

Use Concept Expansion to build concepts.
Conditional Structures:

```xml
<if>
  <cond varName="Topic" operator="Equals">
    X-Phone 7
  </cond>
</if>

Some operators:

- EQUALS, EQUAL_TO_YES, EQUAL_TO_NO
- CONTAINS
- MATCHES_PATTERN
- LESS_THAN, GREATER_THAN
### Action

Used to set variable for profile variables

```
<action varName="PostalCode" operator="SET_TO">
  33801
</action>
```

### Some operators:

- DO_NOTHING_STR
- SET_TO_USER_INPUT(_CORRECTED)
- INCREMENT_BY, DECREMENT_BY
- SET_TO_YES, SET_TO_NOT
<function>
  <script name="Calculate">Name=Calculate Line="\{Phone_Number\}".replace(/-/g, "").replace("(", "").replace(")", "").replace(/ \s/g, """></script>
  <output>
    <prompt/>
    <action varName="Phone_Number" operator="SET_TO">{MCT:CUSTOM:Calculate:value}</action>
  </output>
</function>

Good to know

Support type NLC_MODEL to call NLC!
Create and get the dialogs

- **POST /v1/dialogs:**
  send the XML file and return the id a new dialog.

- **GET /v1/dialogs:**
  get the list of dialogs.
Create and get a specific dialog

- **DELETE /v1/dialogs/dialog_id**: remove a specific dialog.
- **GET /v1/dialogs/dialog_id**: get a specific dialog.
- **PUT /v1/dialogs/dialog_id**: set another XML file.
Create and get a specific dialog

- **POST /v1/dialogs/dialog_id/conversation:** start a new conversation or get response.
- **GET /v1/dialogs/dialog_id/conversation:** get history for a given range.
IBM Watson

IBM Watson Cloud Services

Available Services

Focus on some services

A more complex use case
Speech to text
Natural Language Classifier
Dialog
A prototype
A PROTOTYPE

Browser

mobile

Node.js

PHP

Python

IoT device
A PROTOTYPE
https://github.com/aquemy/watson-homeautomation
THE END
Thank you for your attention!
Questions?