

**COME CREARE UN
CHATBOT**

L'AIML

<http://www.alicebot.org/> - A.L.I.C.E Artificial Intelligence Foundation website.

Il testo che segue è una traduzione del manuale “Basic AIML User Manual”, di Anupama Seshagiri, reperibile in Rete.

Ovviamente si riportano, qui, solo gli argomenti principali che ci serviranno per comprendere il funzionamento del linguaggio.

L'Artificial Intelligence Markup Language è un linguaggio XML compatibile che rende davvero semplice creare un Alice bot anche ai principianti.

L'AIML prevede TAG come gli altri markup language. I più importanti sono:

- . **<aiml>**: è il tag che apre e chiude un document AIML
- . **<category>**: racchiude una “unità di conoscenza” nella base di conoscenza di un Alicebot
- . **<pattern>**: racchiude un pattern semplice che corrisponde a cosa dice o scrive un utente ad un Alicebot
- . **<template>**: racchiude la risposta ad un input utente

Tag AIML

Le *categorie* sono specificate dall'informazione compresa tra i due tag *category*:

```
<category> ...information... </category>
```

In modo simile, si ha per i *pattern*:

```
<pattern> ...information... </pattern>
```

La risposta è invece specificata da:

```
<template> and </template>.
```

Gli elementi posti tra "<" e ">" si dicono **tag AIML** o elementi. Ogni tag deve essere aperto e chiuso.

A volte succede che non vi sia informazione tra un tag aperto ed uno chiuso. In questo caso, invece di `<some-tag></some-tag>` si può scrivere direttamente `<some-tag/>`

tag AIML personalizzati

l'AIML si può estendere creando un numero infinito di nuovi tag, ad esempio per i pronomi in lingue diverse, predicati specifici, o proprietà specifiche dell'applicazione.

I "Predicate tags" hanno un metodo "set" e un "get" specifico. Pronomi quali "it" hanno predicate tag quali `<set name="it"></set>`.

L'AIML prevede un certo numero di built-in tag per i pronomi inglesi.

Usando `<set name='xxx'>` e `<get name='xxx'>` si possono ampliare a piacere le capacità del robot.

TAG Category

Un chatbot risponde a domande.

La domanda specifica può essere visualizzata come una specie di percorso, che comincia al centro della knowledge web e viene costruita attraverso una serie di link e nodi, e che termina in un Response Template.

Il percorso attraverso il quale il (response) template viene chiamato, si dice *Category*.

Considerando solo i tag <pattern> e <template>, ci sono in generale tre tipi di Categorie: atomic, default, recursive.

Atomic

Le categorie *Atomic* sono quelle contenenti pattern specifici, indivisibili (atomici in questo senso), univoci, cioè quelli che non contengono i simboli * o _

Sono le categorie più più semplici e facili da aggiungere in AIML. Esempio:

```
<category>
```

```
<pattern>WHAT IS A CIRCLE</pattern>
```

```
<template><set_it>A circle</set_it> is the set of points  
equidistant from a common point called the center.
```

```
</template>
```

```
</category>
```

nota: é sempre meglio indentare i tag per individuare eventuali errori (es. tag non chiusi)

Default

La categoria *Default* si ha quando il suo pattern contiene i simboli * o _

La categoria Default estrema è `<pattern>*</pattern>`, che fornisce la risposta a qualsiasi input.

La categoria Default più comune ha pattern che combinano poche parole con altri simboli. Per esempio:

`<category>`

`<pattern>I NEED HELP *</pattern>`

`<template>Can you ask for help in the form of a question?</template>`

`</category>`

Recursive

La categoria *Recursive* è quella che “mappa” gli input su altri input, per semplificare il linguaggio o per identificare pattern sinonimi.

L'AIML implementa la ricorsione con l'operatore <srai>.

Molti input sinonimi possono avere la stessa risposta. Per esempio <srai> è molto utilizzato per le forme di saluti. A “GOODBYE” possono corrispondere risposte quali GOODBYE, BYE, CYA CIAO, BYE BYE, ecc.

```
<category>
```

```
<pattern>BYE BYE</pattern>
```

```
<template><srai>GOODBYE</srai></template>
```

```
</category>
```

Può essere usato anche per la semplificazione o la riduzione di pattern di input complessi.

Ad esempio, in Inglese, la domanda "**What is X**" può essere posta in molti modi differenti: "Do you know what X is?", "Tell me about X", "Describe X", "What can you tell me about X?", e "X is what?" sono solo alcuni esempi.

Di solito si cerca di immagazzinare la conoscenza nel modo più conciso possibile, o nella forma più comune. Il tag <srai> riconduce tutte queste forme a quella base:

<category>

<pattern>* DO YOU KNOW WHAT * IS</pattern>

<template><srai>WHAT IS <star/></srai></template>

</category>

Il tag `<star/>` sostituisce il valore indicato da * prima della chiamata ricorsiva a `<srai>`.

Questa categoria trasforma "Do you know what a circle is?" in "WHAT IS A CIRCLE", e poi cerca la miglior corrispondenza per l'input trasformato.

certe forme di ramificazione possono essere realizzate con `<srai>`.

TAG Response

Default Response

Si realizza spesso quando un pattern di input non viene trovato nella kw, ma vogliamo che il robot risponda ugualmente con qualcosa di sensato.

Random Response

Uno strumento molto potente ma ancora di base che fornisce la possibilità di replicare in modo casuale ad una richiesta. E' spesso utilizzata per far sembrare il bot più intelligente e meno prevedibile.

<random>

** What is your name? **

** What is your favorite movie? **

** Will you buy me a drink? **

</random>

Una delle tre risposte sarà scelta in modo casuale.

TAG per esprimere Conditional Elements

L'elemento condizionale istruisce l'interprete AIML a restituire un contenuto specifico a seconda del risultato di una condizione espressa.

NB: Ve ne sono di tre tipi differenti, distinti da un attributo **xsi:type** che permette la validazione da parte di un XML Schema processor, Due di questi tipi possono contenere elementi ``. In pratica un interprete AIML può permettere l'omissione dell'attributo **xsi:type** e determinare, invece, euristicamente, che tipo di condizione (e di azione ``) è in uso.

Block Condition

E' un tipo di condizione che ha un nome che specifica un predicato AIML, oltre a un valore attributo che contiene un'espressione pattern semplice.

```
<aiml:condition  
  xsi:type = "blockCondition"  
  name = aiml-predicate-name  
  value = aiml-simple-pattern-expression >  
  <!-- Contents: aiml-template-elements -->  
</aiml:condition>
```

Se il contenuto dell'attributo valore corrisponde al valore del predicato specificato dal nome, l'interprete AIML restituirà il contenuto della condizione. In caso contrario, restituirà una stringa vuota "".

UN ESEMPIO: Single-predicate Condition

Il Single Predicate Condition deve avere un nome di attributo che specifica un predicato AIML. Deve contenere almeno un element che potrebbe essere di tipo valueOnlyListItem type. Zero o un può essere del tipo defaultListItem.

```
<!-- Category: aiml-template-elements -->  
<aiml:condition  
  xsi:type = "singlePredicateCondition"  
  name = aiml-predicate-name >  
  <!-- Contents: value-only-list-item*, default-list-item{0,1} -  
->  
</aiml:condition>
```

L'istruzione è interpretata come segue:

Gli `` vengono letti nell'ordine

1. Se `li` è un `valueOnlyListItem`, allora il contenuto viene confrontato con il valore del predicato specificato
2. Se corrispondono, restituisce il contenuto di `li` e si ferma
3. Se non corrispondono, continua
4. Se `li` è un `defaultListItem`, allora restituisce il contenuto di `li` e si ferma

Vi sono anche:

- Multi-predicate Condition
- Condition List Items
- Default List Items
- Value-only List Items
- Name and Value List Items

(il meccanismo è simile, ma cambia il risultato. Sono riportati tutti nel manuale che vi ho citato all'inizio e sulle dispense)

Esempio di script aiml

```
<aiml>
```

```
  <category>
```

```
    <pattern> Dobbiamo * </pattern>
```

```
    <template>
```

```
      <srai> Dovremmo <star/>
```

```
      </srai>
```

```
    </template>
```

```
  </category>
```

```
  <category>
```

```
    <pattern> Dovremmo * </pattern>
```

```
    <template> Non sono sicuro di essere pronto per questo.
```

```
    </template>
```

```
  </category>
```

```
<category>
  <pattern> BYE </pattern>
  <template>
    <random>
      <li> Bye. </li>
      <li> Goodbye. </li>
      <li> Bye Bye. </li>
      <li> Sayonara. </li>
      <li> Ciao. </li>
      <li> Alla prossima. </li>
      <li> A più tardi <get name="name" />.</li>
    </random>
  </template>
</category>
</aiml>
```

Ovviamente vi sono alcuni altri TAG, ma l'importante era capire il meccanismo.

I TAG possono essere combinati tra loro.

COME CREARE UN CHATBOT?

Si possono prendere in considerazione diverse opzioni

1) **Verbots** www.verbots.com

È una piattaforma opensource per sviluppare chatbot molto versatili utilizzando una interfaccia semplice e l'editor della knowledge base. Le opzioni sono molte e il costo si aggira sui 30\$/anno per poter condividere il bot con tutti. È possibile fornirlo di voce e animazioni.

2) **Personality Forge** www.personalityforge.com

È una piattaforma on line per creare virtual chatterbot, dopo aver creato un proprio account. Il costo si aggira attorno ai 50\$/anno

3) Pandorabots www.pandorabots.com

Permette di creare bot basati su ALICE/AIML . Non ha opzioni voce o animazione ma prevede un piano gratuito. (per informazioni su ALICE: www.alicebot.blogspot.com)

4) Program your own chatbot from scratch

È ovviamente possibile programmare da capo il proprio chatbot, utilizzando qualsiasi linguaggio... ma bisogna essere un po' più esperti...

CREARE UN WEB-BASED CHATBOT - PANDORABOTS

Creare un bot con Pandorabots è piuttosto semplice e bisogna considerare che Pandorabots è anche una delle compagnie di hosting di chatbot più conosciuta.

Inoltre è particolarmente adatta per i neofiti, e prevede un piano free.

Vediamo passo per passo come creare un bot.

Creare un account

Per prima cosa si vada su

<http://www.pandorabots.com/botmaster/en/home>, e si crei un account (“Account Sign-up”). E’ gratis e si può operare da subito.

Attenzione: il link di cui sopra è stato sostituito da questo:

<http://www.pandorabots.com>

Che è la nuova versione della piattaforma

Anche in questo caso, si deve creare un account.

Progettare il chatbot

Non mi stancherò mai di ripetere che, come in tutte le cose, ma in particolare nell'IT, la prima cosa da fare è pianificare! Una progettazione accurata determina il successo di un progetto. Nel nostro caso occorre pensare alla personalità che vogliamo dare al chatbot, in che lingua parlerà, per quale scopo lo stiamo creando, e quali interazioni realizzerà con gli altri. Occorre giusto un foglio e una penna, per schiarirsi le idee, ma, credetemi, è un'operazione assolutamente necessaria!

The Work Begins: Create Your Bot

Pronti? Allora fate login con l'account creato. Tra le opzioni che appaiono, vedrete “create a pandorabot”. Con un click accederete alla schermata in cui specificare il nome del chatbot, una serie di opzioni ed il set AIML che vorrete (eventualmente) usare. Il mio suggerimento è di creare prima un file AIML, oppure farvi prima “un giro” ed aggiungere poi il file AIML (è possibile modificare le impostazioni anche successivamente). Inizialmente potete anche scegliere il set AIML standard (o, ad esempio, quello italiano). Se selezionate “no initial content”, il bot non avrà nulla da dire 😊

- MY BOTS
- CLUBHOUSE
- HELP

Interactions: 0
View Past 30 Days

Clients: 0
View Past 30 Days

Sessions: 0
View Past 30 Days

Interactions/Session: -
View Past 30 Days

Create Bot

Please specify your bot's name, language, and optional base content.

Name:

Language:

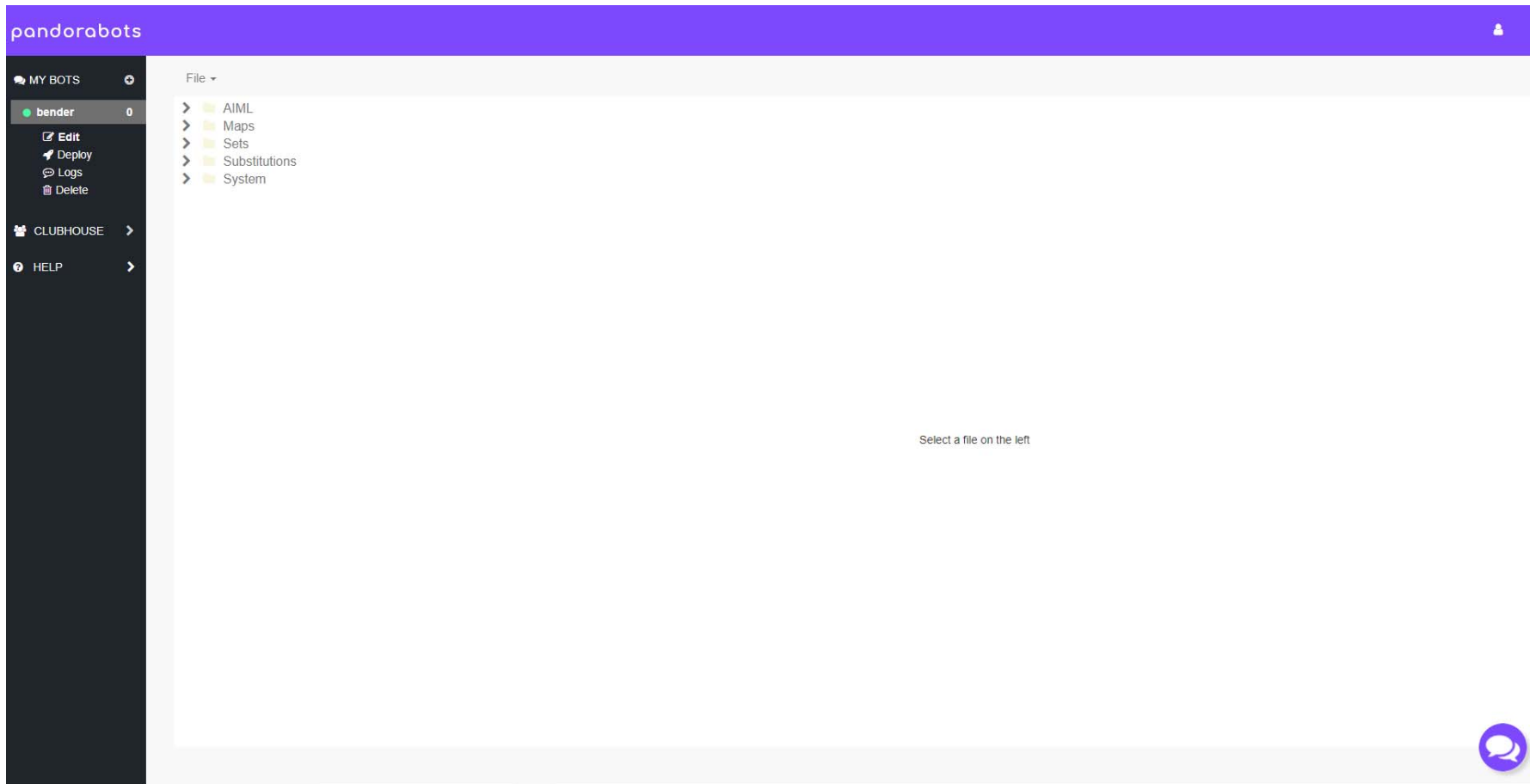
Content:

Cancel Create Bot

Personalizzazione del chatbot

Ok, ora siete gli orgogliosi creatori del vostro primo chatbot.... Però non l'avete ancora pubblicato e poi bisogna lavorarci ancora un po'... Progettare è importante perché occorre rendere il bot credibile. Dovete, cioè, decidere chi sarà il nostro bot, ovvero le sue informazioni di base (sesso, età, localizzazione, e così via). Una volta che avrete progettato sul foglio la personalità del bot, dovrete tradurla in sintassi AIML.

Potete anche aggiungere attribute personalizzati: fissazioni, stranezze... se poi creerete un altro bot, potete anche copiare queste caratteristiche e modificarle per il nuovo. Una volta finite, fate click su “Salva” e su “Submit Changes” se il bot è pubblicato (ma lo vedremo poi)



Se volete creare nuove categorie (il vostro file AIML) dovete fare click su FILE, NEW, AIML



MY BOTS +

- raff 0
- **prova** 0
- ✎ Edit
- 🚀 Deploy
- 🗨️ Logs
- 📦 Push to Repo

📁 DIRECTORY >

- 📖 HELP
- Docs
- FAQ
- Tutorial
- Quickstart

File ▾ Insert ▾

- Save (ctrl-s)
- New
 - AIML
 - Maps
 - Sets
 - Substitutions
 - Pdefaults
- Upload
- Import from Repo
- Download Zip
- Delete

📦 Push to Repo

📦 Publish Changes

Select a file on the left



Se, ad esempio, copiate il testo seguente:

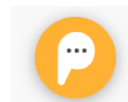
```
<category>  
  <pattern>HI</pattern>  
  <template>Hello, world!</template>  
</category>
```

Una volta fatto, salvate il file.

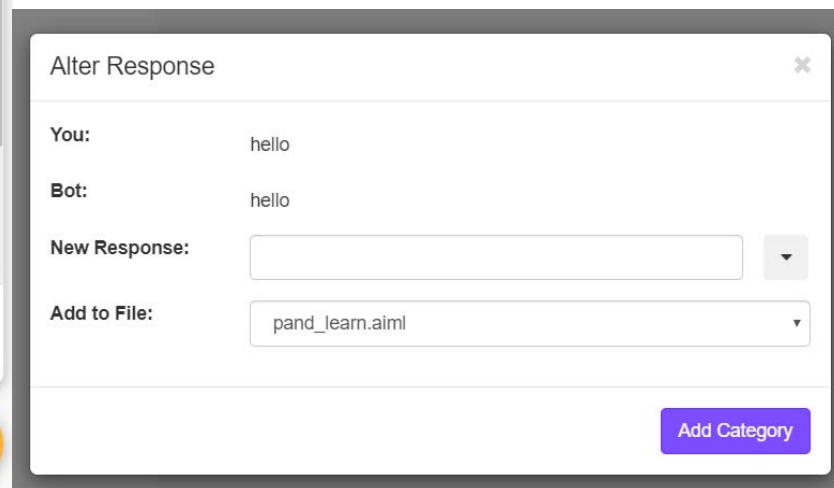
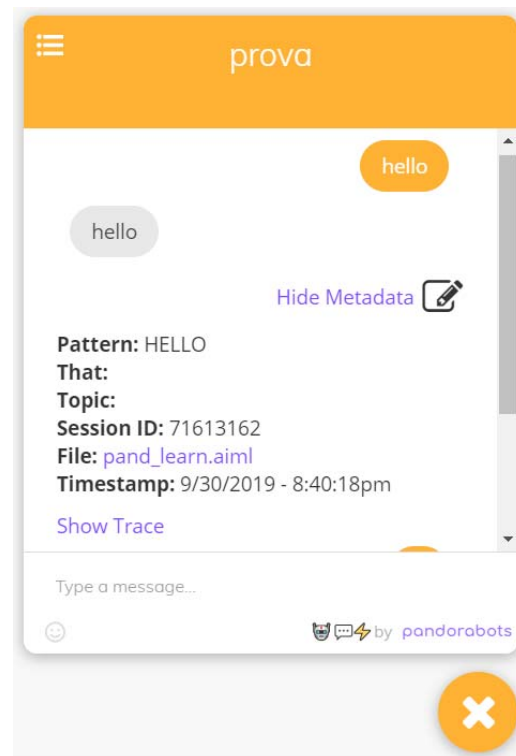
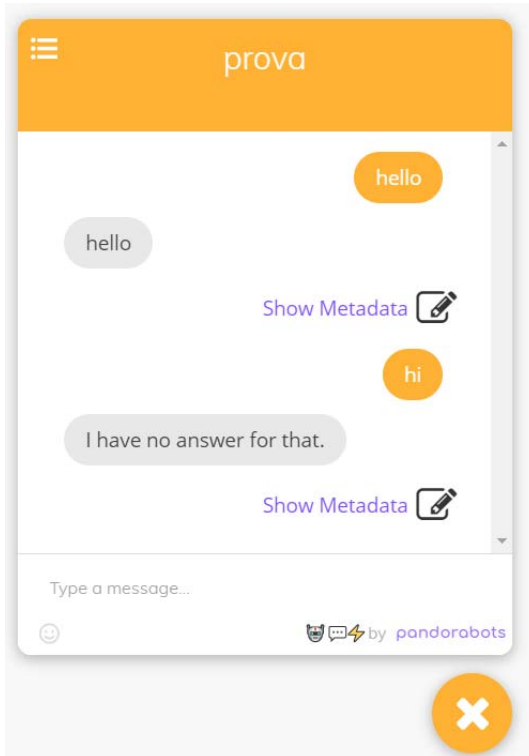
Il bot risponderà “hello world!”

Come controllare se è vero?

Fate click sulla chat, che compare in basso a destra:



Avete ora la possibilità di interrogare il bot: (potete anche visualizzare i metadati e aggiornare le risposte, facendo click sulla matita)



Con il clic sulla matita compare l'edit e potete specificare la risposta desiderata alla domanda e decidere a quale file aggiungerla (per default è pand_learn.aiml)

Ricordatevi di salvare sempre il bot.

Ora provate a divertirvi ad aggiungere le categorie che desiderate.

Un altro modo di aggiornare le risposte dei bot è utilizzare la funzione log. Vale quando il bot è pubblicato (ad esempio nella community) e altri utenti vi accedono per effettuarne il test. Come vedete, accanto ad ogni domanda fatta al bot appare la “matita di edit”... con un clik, si può specificare la risposta (desiderata o mancante) ed aggiungerla all’apprendimento del bot.

The screenshot shows the Pandorabots interface. On the left, a sidebar contains navigation options: 'MY BOTS' (with a sub-menu for 'bender' containing 'Edit', 'Deploy', 'Logs', and 'Delete'), 'CLUBHOUSE', and 'HELP'. The main area displays a chat log for the bot 'bender' from 9/26/2017 to 10/26/2017. The log shows a message from 'undbcc2a (3)' with three responses: 'Hello, world!', 'Hello, world!', and 'I'm still in training and don't have an answer yet.'. Each response has a blue button with an edit icon and a 'Show Metadata' link. The chat log also shows user messages: 'hi', 'hi there bender!', and 'yo yo yo'.

Anche in questo caso è possibile mostrare i metadata.

Il bot che abbiamo creato è nella sandbox (cioè nell'area di test cui accedete solo voi).

E' utile condividere il bot con altri utenti, in modo che anche loro lo testino (e voi potete testare i loro) aiutandovi a perfezionarlo.

Per fare ciò, click su DEPLOY e scegliere INTERNAL CHANNEL, seguendo le istruzioni (occorre fornire qualche informazione)

MY BOTS

raff 0

prova 0

- Edit
- Deploy
- Logs
- Push to Repo

DIRECTORY

HELP

- Docs
- FAQ
- Tutorial
- Quickstart

Sandbox



Internal Bot

Developer Cha



API



Chat Widge



Landing Pa

3rd Party Char

Upgrade to access 3rd

Discovery Tools

Upgrade to access more developer tools.



Amazon .Bot Domain

Set up

Add to Bot Directory

Adding your Sandbox bot to the Pandorabots Bot Directory makes it available for other botmasters to discover and chat with, and is a great, free way to collect chatlogs during development. Note: Bot Directory interactions do not count against your monthly limit.

Type: Assistance

Name:

Description:

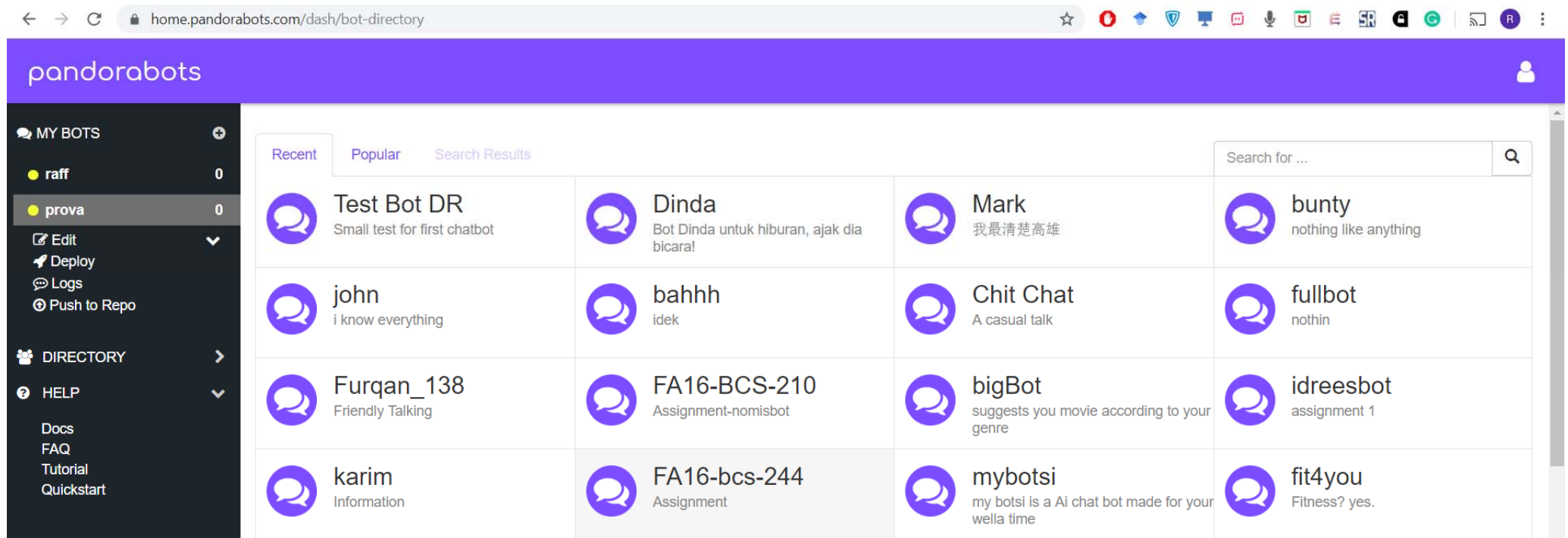
In publishing my bot to the Pandorabots' Bot Directory, I represent and warrant that my bot does not violate the Pandorabots [Terms of Service](#), contain offensive or inappropriate content or spam, and is not a boring FrankenZombieBot with no content at all. I understand that my bot may be removed if it is deemed to be in violation or at Pandorabots' sole discretion.

Cancel Add



















Ovviamente si possono anche scegliere anche altri canali, esterni, ma conviene farlo se il bot è più sviluppato. Inoltre, per pubblicare su questi canali, occorre avere un accesso premium.

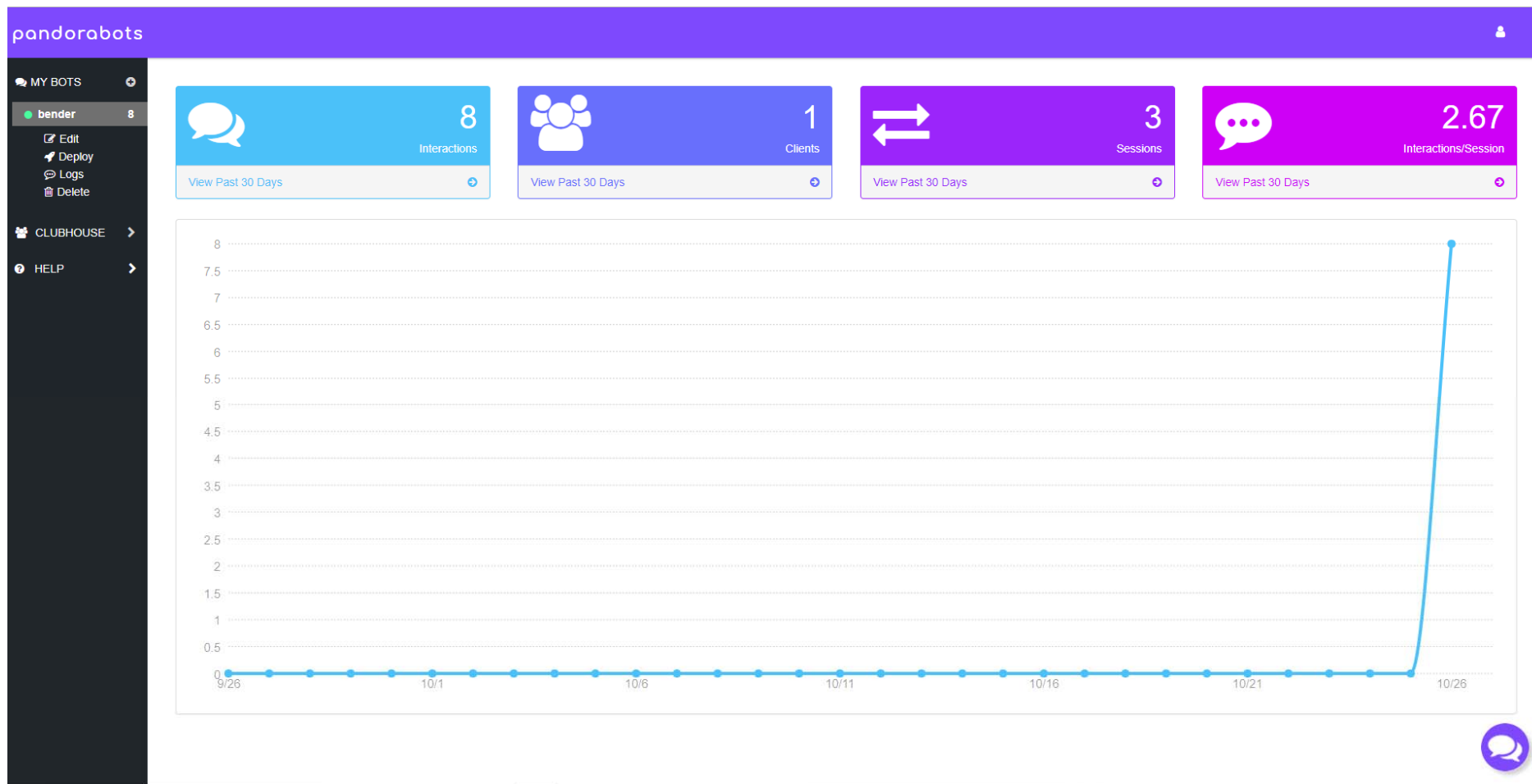
Il vostro bot comparirà nella directory di pandorabot, come quelli degli altri utenti:



The screenshot shows the Pandorabots dashboard at the URL `home.pandorabots.com/dash/bot-directory`. The interface features a purple header with the 'pandorabots' logo and a user profile icon. A dark sidebar on the left contains navigation options: 'MY BOTS' (with a plus icon), 'raff' (0), 'prova' (0), 'Edit', 'Deploy', 'Logs', 'Push to Repo', 'DIRECTORY' (with a right arrow), and 'HELP' (with a down arrow). Under 'HELP', there are links for 'Docs', 'FAQ', 'Tutorial', and 'Quickstart'. The main content area has tabs for 'Recent', 'Popular', and 'Search Results'. A search bar is located at the top right of the main area. Below the tabs is a grid of 12 bot cards, each with a purple chat icon, a name, and a short description. The bots listed are: Test Bot DR (Small test for first chatbot), Dinda (Bot Dinda untuk hiburan, ajak dia bicara!), Mark (我最清楚高雄), bunty (nothing like anything), john (i know everything), bahhh (idek), Chit Chat (A casual talk), fullbot (nothin), Furqan_138 (Friendly Talking), FA16-BCS-210 (Assignment-nomisbot), bigBot (suggests you movie according to your genre), idreesbot (assignment 1), karim (Information), FA16-bcs-244 (Assignment), mybotsi (my botsi is a Ai chat bot made for your wella time), and fit4you (Fitness? yes.).

Recent	Popular	Search Results	
 Test Bot DR Small test for first chatbot	 Dinda Bot Dinda untuk hiburan, ajak dia bicara!	 Mark 我最清楚高雄	 bunty nothing like anything
 john i know everything	 bahhh idek	 Chit Chat A casual talk	 fullbot nothin
 Furqan_138 Friendly Talking	 FA16-BCS-210 Assignment-nomisbot	 bigBot suggests you movie according to your genre	 idreesbot assignment 1
 karim Information	 FA16-bcs-244 Assignment	 mybotsi my botsi is a Ai chat bot made for your wella time	 fit4you Fitness? yes.

Avrete, infine, anche accesso alle statistiche riguardanti il bot:



Per imparare qualcosa di più avanzato, conviene seguire il tutorial **Bot Building 101**

Un'ultima funzione utile è l'INTENTS TREE che trovate nel menu edit. Da qui, potete vedere come il bot è costruito.

The screenshot displays the Pandorabots web interface. At the top, a purple header contains the text "pandorabots" and a user profile icon. Below the header, a dark sidebar on the left lists navigation options: "MY BOTS" (with a plus icon), "raff" (0), "prova" (2), "Edit", "Deploy", "Logs", "Push to Repo", "DIRECTORY", and "HELP" (with a dropdown arrow). Under "HELP", there are links for "Docs", "FAQ", "Tutorial", and "Quickstart".

The main content area features a "Refresh Map" button, a "Select File(s)" dropdown, and an "Add Category" dropdown. Below these are several intent nodes, each represented by a purple circle connected to a bot icon and a response bubble:

- An empty intent node with a response bubble containing "I have no answer for that."
- Two intent nodes labeled "come stai" and "come ti senti" both connected to a central node labeled "How are you", which has a response bubble containing "I am fine. What about you?"
- An intent node labeled "hello how do you do" connected to a node labeled "ciao", which has a response bubble containing "ciao".
- An empty intent node.
- An intent node labeled "Ciao" connected to a node containing the code snippet `<srai>CIAOHELLOHOW DO YOU DO</srai>`, which has a response bubble containing the same code snippet.

Inoltre, se fate click su un nodo, potete aggiungere le reduction (cioè domande dell'utente che conducono alla stessa risposta) molto facilmente (una per riga).

The screenshot displays the Pandorabots web interface. On the left is a dark sidebar with navigation options: MY BOTS (raff: 0, prova: 2), Edit, Deploy, Logs, Push to Repo, DIRECTORY, and HELP (Docs, FAQ, Tutorial, Quickstart). The main workspace features a flowchart with nodes and connecting lines. Nodes include 'come stai', 'come ti senti', 'How are you', 'hello how do you do', 'ciao', and 'Ciao'. A modal dialog titled 'Edit Terminal Pattern' is open, showing a text input field with 'How are you', a 'Reductions' section with an 'Add Reduction(s)' input, and buttons for 'Delete', 'Open File', and 'Save'. A 'Pro tip' message is visible below the reductions input. The background shows a chat window with a message 'I have no answer for that.' and a terminal output showing the raw pattern: `<srai>CIAOHELLOHOW DO YOU DO</srai>`. The top navigation bar includes 'Refresh Map', 'Select File(s)', and 'Add Category' buttons.

Un altro modo per effettuare un test o aggiungere domande è utilizzare CHAT DESIGN, sempre sotto il menu edit.

Il funzionamento è intuitivo: a sinistra si specifica l'input dell'utente e a destra l'output (risposta) desiderato.

The screenshot displays the Pandorabots interface. On the left, a dark sidebar menu contains the following items: "MY BOTS" with a plus icon, "raff" with a yellow dot and "0", "prova" with a yellow dot and "2", "Edit" with a pencil icon, "Deploy" with a rocket icon, "Logs" with a speech bubble icon, "Push to Repo" with a circular arrow icon, "DIRECTORY" with a right arrow, "HELP" with a question mark icon, "Docs", "FAQ", "Tutorial", and "Quickstart". The main area features a purple header with "pandorabots" and a user icon. In the center, a mobile phone mockup shows a chat conversation with a purple header "udc.ai ml". The chat history includes: a grey bubble "How are you", a blue bubble "I am fine. What about you?", a grey bubble "ciao", a blue bubble "ciao", a grey bubble "Ciao", and a blue bubble containing the code: `<srai>CIAOHELLOHOW DO YOU DO</srai>`. Below the phone, there are two editors: "User Input" on the left and "Bot Response" on the right. Each editor has a text input field, a "Delete" button, and an "Update" button. The "Bot Response" editor also includes a "Show Reductions" link and an "Open File" link. A floating orange chat bubble icon is visible in the bottom right corner.

La cosa pi semplice, con un po' di pratica, è comunque operare direttamente sul file AIML. Qui di seguito avete un file di esempio:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<aiml version="1.0">
```

```
<category>
```

```
  <pattern>QUANDO È IL TUO COMPLEANNO</pattern>
```

```
  <template>Il mio compleanno è <bot  
name="compleanno"/>.</template>
```

```
</category>
```

```
<category>
```

```
  <pattern>QUANDO È IL TUO COMPLEANNO</pattern>
```

```
  <template>Il mio compleanno è il <bot  
name="compleanno"/>.</template>
```

```
</category>
```

```
<category>
```

```
<pattern>QUAL È IL TUO COLORE PREFERITO</pattern>
```

```
<template>Il mio colore preferito è il <bot name="colore  
preferito"/>.</template>
```

```
</category>
```

```
<category>
```

```
<pattern>HAI QUALCHE HOBBY</pattern>
```

```
<template>Amo studiare le <bot name="hobby"/>!</template>
```

```
</category>
```

```
<category>
```

```
<pattern>SEI UN MASCHIO O UNA FEMMINA</pattern>
```

```
<template>Sono una <bot name="sesso"/>... non si  
capisce?</template>
```

```
</category>
```

```
<category>
```

<pattern>DI CHE SESSO SEI</pattern>

<template>solo un uomo poteva fare questa domanda!...
secondo te???

</category>

</aiml>

Visto che sono sicura che avete capito l'AIML e che vi sentite più sicuri di voi stessi, a questo punto potete divertirvi a scrivere un file molto più complesso (scaricando quello in parte già creato ed ampliandolo, o intervenendo sul file on line, o, meglio, creandone uno tutto vostro che implementi l'AI di cui avete deciso di dotare il chatbot).

Si possono anche associare più file AIML (ad esempio uno con risposte standard, che poi si possono utilizzare anche in altri bot; uno con risposte random e così via).

That's all, Folks! ... ma... c'è sempre qualche trucco!

COME RENDERE “PIÙ CONVINCENTE” UN CHATBOT

Uso dei proverbi

I proverbi possono rispondere a molte domande... 😊

Pensate a: “non ci sono più le mezze stagioni”, “tanto va la gatta al lardo...”, “tra il dire e il fare c’è di mezzo il mare...” ecc. ecc., magari introdotti da un “Come ben sai: ...”

Aggiungere “intelligenza” al chatbot

L'intelligenza del chatbot dipenderà da due fattori: una buona logica (torna il fatto che deve essere progettato bene) e un gran numero di record (i casi delle risposte e delle domande).

Si può pensare di programmare qualcosa che apprenda dalle conversazioni che il chatbot tiene con gli utenti.

In pandorabots esiste questa possibilità: tramite i log, si può accedere ad una conversazione ed utilizzare le domande per addestrare (in modalità “train”) ulteriormente il bot.

Più Avanti, quando saprete crearle, potrete anche pensare ad un meccanismo basato su algoritmi di apprendimento più evoluti.

Chiamare l'utente per nome in modo casuale

Se ogni tanto, nelle risposte, il bot chiamerà l'utente per nome, la conversazione sembrerà più naturale.

Per realizzare questa opzione, ricordate l'uso di *get* e *set* visti quando abbiamo parlato dell'AIML.

Inoltre... qui trovate alcuni tutorial:

http://www.chatterbotcollection.com/category_content_s.php?id_cat=20

REPETITA IUVANT!

So che i chatbot sembrano semplici e entusiasmano tutti, ma meglio ripetere **PERCHE' SCONSIGLIO DI CREARE UN CHATBOT COME PROGETTO DI FINE CORSO:**

Il chatbot è nato per superare il test di Turing, quindi la progettazione deve essere così accurata che un utente deve credere che dietro al chatbot vi sia una persona!

Occorre una grande padronanza della logica (che va progettata perfettamente), tanto tempo e soprattutto una progettazione accuratissima!