



Chatbot and Artificial Intelligence



From:

<https://pandorafms.com/manual/!776/>

Permanent link:

https://pandorafms.com/manual/!776/en/documentation/10_pandora_itsm/16_pandora_itsm_ai

2024/06/10 14:34



Chatbot and Artificial Intelligence

[Back to Pandora FMS documentation index](#)

Introduction

The Integria IMS chatbot is made up of two services: a conversation or chat server (with its WEB client) and a conversational artificial intelligence engine (optional).

The artificial intelligence engine can be used to learn from the information entered into the Integria IMS knowledge base to provide quick answers to user questions.

- Automatic - Uses machine learning models to automatically provide answers to the user based on prior learning.
- Manual: an Integria IMS operator responds to the user through the chat interface.
- Mixed: Integria IMS assists the operator by showing possible answers to the user's questions.

The architecture consists of three fundamental elements:

- The chat client, with which the user interacts.
- The chat server, with which the operator interacts.
- The prediction engine, which offers answers based on the user's questions.



Both the client and the chat server are part of an Integria IMS Enterprise installation.

To add predictive capabilities to Hybrid Helpdesk you will need to install and activate the Prediction Engine as described in the following sections.

Chat server installation

[By default it is installed when using the cloud installation method. To activate it, go to the chat Setup option:](#)



[The webservice URL must be configured with the public URL of your Integria IMS installation, as configured in the main section of Setup:](#)



```
{ { :wiki:iims-07_advanced-01_ai-img040.png } }
```

In case it doesn't work as expected, reproduce the Node.js installation steps:

```
curl -sSL
http://firefly.artica.es/projects/integria/integria_deploy_enterprise.sh | sh
curl -sL https://rpm.nodesource.com/setup_12.x | sh
yum install -y nodejs
npm i -g pm2
cd /var/www/html/integria/extras/chat_server
npm update
pm2 start server.js
```

You must edit the file `/etc/systemd/system/integria-chat.service` so that the service starts automatically every time the operating system boots, adding the following:

```
[Unit]
Description=Integria-Chat-Server
After=network.target
[service]
Type=simple
ExecStart=/usr/bin/nodeserver
restart=always
# Consider creating a dedicated user for Wiki.js here:
user=root
Environment=NODE_ENV=production
WorkingDirectory=/var/www/html/integria/extras/chat_server
[Install]
WantedBy=multi-user.target
```

Save the file and exit the text editor used. Run below:

```
systemctl daemon-reload
systemctl start integria-chat.service
systemctl enable integria-chat
```

Access to the chat server database must be configured, edit the file `/var/www/html/integria/extras/chat_server/config/config.js` and modify the necessary parameters of the database (defaults are assumed, should be changed if you have a custom installation):



Installation of the artificial intelligence engine

Run the following commands as root (super user):

```

yum install python3 python36-Cython
wget http://xxxxxx/prediction_engine-latest.tgz
tar xvzf prediction_engine-latest.tgz
cd prediction_engine
./install.sh
service prediction_engine.service restart

```

This will start the service on port 6000/tcp .

AI Settings

The AI is configured independently on each channel. To do this, go to Support → Chat → Channel Management and edit the settings:



- IA url: URL where the webservice listens to the artificial intelligence engine. It must be in format

```
HTTP: IP_ADD
```

- IA Port: Default 6000. You shouldn't change this unless you had to do port forwarding somewhere in between.
- Initial response time: Seconds in which the AI will answer the first user request.
- Response time between conversations: Response time between question and question once the conversation has started.
- Progress bar timeout: Time that the system will allow the operator to choose an answer from those provided before automatically answering the best option.
- Certainly threshold: Degree of uncertainty supported by the AI when choosing an option from those offered, the value must be between 0 and 1, a value is recommended initial value of 0.5 (50%) and adjust as necessary.

Support for personalized learning models

- Before carrying out any customization, it is recommended to make a backup of the current models:

```
cp -r /opt/prediction_engine/models /opt/prediction_engine/models.bak
```

- To restore the models from the backup run:

```
cp -f /opt/prediction_engine/models.bak/* /opt/prediction_engine/models/
```

Questions and Answers (Knowledge Base)

The question and answer model is trained from a CSV file with the following format:

```
Language code;Question;Answer
```

Example:

```
"is";"question 1";"answer 1"
"is";"question 2";"answer 2"
"in";"question 1";"answer 1"
"in";"question 2";"answer 2"
...
```

It can be obtained automatically from the Integria IMS database with the commands:

```
rm -f /opt/prediction_engine/data/integria_kb.zip 2>/dev/null
echo "SELECT id_language AS lang, title as question, data as answer FROM
integria.tkb_data INTO OUTFILE '/opt/prediction_engine/data/integria_kb.csv'
FIELDS TERMINATED BY ';' ENCLOSED BY '\"' LINES TERMINATED BY '\n';" | mysql -u
integria -p integria && zip -j /opt/prediction_engine/data/integria_kb.zip
/opt/prediction_engine/data/integria_kb.csv
```

To update models:

```
cd /opt/prediction_engine/src
python kb_train.pyc
```

Dialogues (conversational)

The conversational model from YAML files with the following structure:

```
categories:
- category 1
- category 2
- ...
conversations:
- - text 1
- text 2
- ...
- - text 1
- text 2
- text 3
- ...
...
```

The `.yaml` files (the file name is not important) should be placed in the directory:

```
/opt/prediction_engine/data/chat_xx
```

Being `xx` the ISO code of the language to be updated (es for the Spanish language).

To update the models run:

```
cd /opt/prediction_engine/src
python chat_train.pyc
```

How to use the chat

The Integria chatbot uses channels to define different places to have conversations between operators (Integria IMS users with special permissions to manage conversations) and normal users (normal Integria IMS users or simple anonymous visitors, if the chat is used from outside of Integria IMS).


Manage a channel

Support Menu → Chat → Manage Channel:



Add operators to the channel

The channel must have at least one operator, who will answer the users' chat requests. You will be able to add any user with chat operator permissions and assign them a description, an avatar (different from the one in their user file) and the languages in which they can answer chats.

Click on the operators icon  to add a user and then the add operator button:



Use the screen to define an operator, example:



Once added, other users will be able to start chat:



Channel operators

In order for operators to receive audible notifications when a user opens a chat, they must be on the chat control screen. It is accessed through the menu Support → Chat → View chat.




The operator will enter one of those chats and interact with the other party:



In the event that the chat is opened by an internal user, it will indicate which user it is.

Using the chat outside of Integria IMS

To use the chat outside the Integria IMS interface, for example on a web page or in another application, press the star icon  to show the snippet of JavaScript code to embed in the application:



Note that the URL that is included in the code is the URL that is defined as the public URL in the chat server options. Said URL must be accessible from where users access it, in most cases, this means that it must be a public Internet URL.

Example:



```
{{ :wiki:iims-07_advanced-01_ai-img150.png }}
```

The conversation starts by clicking on the icon at the bottom right, Help button:



[Back to Pandora FMS documentation index](#)