ATIPXpress

CoreNLP AI Implementation

v11.11.0

February 2025

ATIPXpress v11.11.0 CoreNLP AI Implementation

Contents

1 AI Assistant Implementation

Welcome to the AI Assistant Implementation guide. This document provides comprehensive instructions on how to set up and configure the AI Assistant feature in our latest release.

Note: These requirements were added as of v11.7.0. Any customers using AI Assistant who are upgrading from any version under 11.7.0 must complete these steps for the AI Assistant to function.

With the updated implementation process, you will be able to leverage advanced natural language processing capabilities to enhance your case management workflows.

1.1 Implementation Process

To implement the AI Assistant, you're required to install CoreNLP, then apply the resulting URLs within configuration. This guide is structured into three major sections to streamline the setup process:

1. *Introduction*: An overview of the document, highlighting the new implementation process for version 11.7.0.
2. *Install Core NLP*: Instructions for setting up the Core NLP application, which provides the necessary language processing tools for the AI Assistant.
3. *Application Configuration*: Guidance on configuring the application settings to ensure optimal performance and integration of the AI Assistant.

By following this guide, you will be able to effectively implement the AI Assistant, providing users with an enhanced and efficient document management experience.

1.2 Prerequisites

To install the AI Assistant, you must fulfill the following prerequisites:

1. The ATIPXpress API must be installed. This is a requirement for using the application in all versions v11.7.0 and above.
2. You must have AI Assistant license(s) for your organization. Contact OPEXUS support for details.

2 Install CoreNLP

You must install Standford CoreNLP to use the AI Assistant. Two options are provided: Docker installation, and installation using Java. Each process is covered in the following sections.

2.1 CoreNLP for Docker

Follow the steps in the section below to deploy CoreNLP using Docker.

2.1.1 Install Docker Desktop for Windows

You must have Docker Desktop installed. If you don't have Docker Desktop installed, follow the steps below:

1. **Download Docker Desktop**: Visit the official Docker website and download the Docker Desktop version for Windows. Docker Desktop
2. **Run the Installer**: Execute the installer and follow the instructions. You may need to enable virtualization in your BIOS if it's not already activated.
3. **Restart Your Computer**: Restarting ensures all necessary features are activated and functioning.
4. **Start Docker Desktop**: Open Docker Desktop from the **Start** menu. It might take a while to initiate for the first time as it sets everything up.

2.1.2 Configure Docker Desktop

Check Settings: Click on the **Docker** icon in the system tray and open **Settings** or **Preferences**. Make sure to adjust the memory and CPUs allocated according to the software's needs you plan to run CoreNLP.

2.1.3 Get the Docker Image for Stanford CoreNLP

1. **Open Command Prompt or PowerShell**: You can do this by searching for "cmd" or "PowerShell" in the **Start** menu and selecting the application.
2. **Download the CoreNLP Docker Image**: Run the following command to download the official image:

> Bash:

```
docker pull nlpbox/corenlp
```

This command pulls the Docker image containing Stanford CoreNLP from Docker Hub.

2.1.4   Run CoreNLP Inside a Docker Container

Start the Container with Custom Settings: To use CoreNLP with extended timeout settings and a custom port, execute the following command:

Bash:

```
docker run -p 59000:59000 -e TIMEOUT_MILLISECONDS=600000 -d
nlpbox/corenlp
```

This command starts the Docker container exposing port 59000 for local use and sets the server's timeout to 600,000 milliseconds (10 minutes). The -d option ensures the container runs in "detached" mode, freeing up the command line for other uses.

2.1.5   Verify Execution

To check if the CoreNLP server is running correctly, you can try accessing http://localhost:59000 in your browser. If configured correctly, you should be able to see the Stanford CoreNLP web interface.

With these steps, Docker and Stanford CoreNLP are now set up on your Windows system with specific customizations to better suit your analysis needs and extended timeout settings.

2.2   Install CoreNLP Using Java

2.2.1   Install Java

Stanford CoreNLP requires Java 8 or later. If you do not already have this application installed on the application server, follow the steps below:

1. Download Java:
    a. Visit the Oracle website to download the Java JDK: Oracle JDK Downloads.
    b. Choose the appropriate Windows installer for your system architecture (32-bit or 64-bit).
2. Install Java:
    a. Run the downloaded installer and follow the installation instructions.
3. Verify Java Installation:
    a. Open Command Prompt and type:
        Bash:

        ```
        java -version
        ```

b. This should display the installed version of Java. Ensure it is Java 8 or higher.

2.2.2 Download and Set Up Stanford CoreNLP

Download Stanford CoreNLP:

1. Go to the Stanford CoreNLP official website:
   https://stanfordnlp.github.io/CoreNLP/index.html
2. Download the latest version of Stanford CoreNLP zip file.
3. Once downloaded, extract the zip file to a directory of your choice, for example, C:\StanfordNLP.

2.2.3 Run Stanford CoreNLP Server

1. Launch the Stanford CoreNLP Server:
2. Open **Command** Prompt.
3. Navigate to the directory where you extracted Stanford CoreNLP:

   Bash (example):

   ```
   cd C:\StanfordNLP
   ```

4. Use the following command to start the server:

   Bash:

   ```
   java -mx4g -cp "*"
   edu.stanford.nlp.pipeline.StanfordCoreNLPServer -port 59000 -
   timeout 15000
   ```

Note: Port 59000 is an example, you are not required to use this port. If you change this value.

5. This command starts the CoreNLP server on port 59000 with a timeout of 15000 milliseconds and allocates 4 GB of memory to the process.

2.2.4 Verify the Server

1. Open a web browser and go to http://localhost:59000

Note: The "59000" value above is based on the port configured in Section 3.2.3. Use your configured port value in this URL.

You should see the Stanford CoreNLP server interface, indicating that the server is running and ready to process text. Leave this browser tab open, you'll need to retrieve this URL and add it to ATIPXpress configuration later in this process.

3   Application Configuration

3.1   General Configuration

1. Log in to the application as an administrator and access **Administration > System Configuration > General Configuration**.
2. Under *Application Next-Gen URL*, copy and paste the API This should follow the format https://<IP>/ATIPXpressAPI where <IP> is the installed IP for the API.
3. In *the AI API URL* field, copy and paste the AI URL (from step 2 in section 3.2.4).
4. Click **Save** to save the configuration.

3.2   AI Roles and Permissions

There are three types of AI user permissions within the application:

1. AI Config Application Role Permission: Users with the AI Config application role permission can view and edit three AI configuration pages. Click the links below to learn more about these configuration pages:
    a. [AI Redaction Templates](#)
    b. [Redaction Knowledge Base](#)
    c. [AI Trainers](#)
2. AI User Role: Application users who are allowed to use the AI Assistant feature within document management. This is a license-based role, and all AI users must have an active license.
3. AI Trainer Role: A subset of the AI Users, AI Trainers are selected to feed redaction data into the Redaction Knowledge Base, which contains a repository of all redactions performed by AI Trainer to identify patterns and frequently encountered data.

3.3   Using the AI Assistant

There are two main components to the AI Assistant feature. AI Redaction Templates, and applying these templates to documents in Document Management. Follow the links below to learn more about using the AI Assistant with CoreNLP:

[Using the AI Assistant](#)

[AI Redaction Templates](#)