



## 1. Stanford CoreNLP Capability

A Blue Prism skill that implements the Web API interface to the [Stanford CoreNLP server](#).

Whats Included:

- Stanford CoreNLP Skill v1.0.bpskill - Blue Prism skill
- Standford CoreNLP - Basic Actions v1.0.bprelease - A VBO to expose the Stanford CoreNLP Skill via Blue Prism actions.
- Standford CoreNLP - Example Process v1.0.bprelease - A sample process that uses the Stanford CoreNLP Skill and VBO for simple NLP tasks.

### Documentation

- [Blue\\_Prism\\_Stanford\\_CoreNLP\\_Capability\\_Skill\\_User\\_Guide.pdf](#)

### Pre-Requisites

- Open JDK 8 or equivalent (64-bit preferred due to the VM memory requirements)
- Stanford CoreNLP 3.9.2

## 2. Getting Started:

- Install the JDK:
  - I installed [OpenJDK 8 \(LTS\)](#)
- Download the [CoreNLP v3.9.2 package](#) and unzip it to the directory of you choice.
  - I installed it in `C:\BluePrism\Projects\Stanford\CoreNLP` but it should not matter since the VBO accesses the server API via the URL `http://localhost:9000`
- Install the Blue Prism skill from the Digital Exchange:
  - Install the **Stanford CoreNLP Skill v1.0.bpskill** to your Blue Prism database.
- Optionally install the additional Blue Prism assets from the Digital Exchange:
  - Install the **Standford CoreNLP - Basic Actions v1.0.bprelease** to your Blue Prism database.
  - Install the **Standford CoreNLP - Example Process v1.0.bprelease** to your Blue Prism database.

### 2.1 Verifying the CoreNLP Server:

NOTE: Once you have the JDK installed, the CoreNLP is unzipped and ready and the bprelease installed and ready type the following in a command window...

- To start the server, open a command line window or use a batch file to execute the following commands:

```
cd <Enter Your CoreNLP install Location here >
java -cp "*" -Xmx8g edu.stanford.nlp.pipeline.StanfordCoreNLPServer -annotators
"tokenize" -preload "tokenize, ssplit, pos, parse, lemma, depparse"
```

Once the server has started, using the browser of your choice and enter the following URLs to test if the server is alive and ready to take requests:

- <http://localhost:9000/live> - server started successfully
- <http://localhost:9000/ready> - the default configurations are loaded and ready to take requests

If this all goes well you should be able to run the provided **Stanford CoreNLP - Example Process v1.0.bprelease** with the **Stanford CoreNLP - Basic Actions v1.0.bprelease**.

For more detailed information on how to use the Stanford CoreNLP API please visit:

- [Stanford CoreNLP](#)
- [Stanford CoreNLP FAQ](#)

**!** Important Note: Once the CoreNLP server is setup correctly it runs very well. If for some reason you run into issues starting the server or with slow performance please refer to the following sections for help:

- [CoreNLP Server](#)
- [Understanding memory and time usage](#)

## 3. Using the Skill

There is essentially two ways to use the Stanford CoreNLP API in Blue Prism.

- Using the [skill](#) directly
- Using and modifying the additional [VBO](#) asset.

Below is a list of available actions from the Web API:

### 3.1 Using the Skill

Annotate:

This is the general action to annotate text using one of the provided "annotators" included in the Stanford CoreNLP API.

[Complete Annotator Listing](#)

Input Parameter	Type	Required	Description
properties	Text	Y	The properties JSON string to pass to the CoreNLP server.
text	Text	Y	The text to apply the annotators list.

Output Parameter	Type	Description
Response_Content	Text	A JSON value containing response information related to the request.
HTTP Status Code	Text	HTTP Status code returned in response.
Response Headers	Collection	Collection of headers data returned in response.

## TokenRegex:

A simple, rule-based NER over token sequences using Java regular expressions.. This can also be run as an annotator using Annotate action with the annotator 'tokensregex'

Input Parameter	Type	Required	Description
pattern	Text	Y	The TokensRegex pattern to annotate.
properties	Text	Y	The properties JSON string to pass to the CoreNLP server
text	Text	Y	The text to apply the annotators list.

Output Parameter	Type	Description
Response_Content	Text	A JSON value containing response information related to the request.
HTTP Status Code	Text	HTTP Status code returned in response.
Response Headers	Collection	Collection of headers data returned in response.

## Semgrex:

An implementation of the SemgrexPattern pattern for matching node and edge configurations of a dependency graph.

Input Parameter	Type	Required	Description
pattern	Text	Y	The Semgrex pattern to annotate.
properties	Text	Y	If true, entire sentences must match the pattern, rather than the API finding matching sections.
text	Text	Y	The text to apply the annotators list.

Output Parameter	Type	Description
Response_Content	Text	A JSON value containing response information related to the request.
HTTP Status Code	Text	HTTP Status code returned in response.
Response Headers	Collection	Collection of headers data returned in response.

## Tregex:

A utility for matching patterns in trees, based on tree relationships and regular expression matches on nodes (the name is short for "tree regular expressions")

Input Parameter	Type	Required	Description
pattern	Text	Y	The Tregex pattern to annotate.
properties	Text	Y	The properties JSON string to pass to the CoreNLP server
text	Text	Y	The text to apply the annotators list.

Output Parameter	Type	Description
Response_Content	Text	A JSON value containing response information related to the request.
HTTP Status Code	Text	HTTP Status code returned in response.
Response Headers	Collection	Collection of headers data returned in response.

## Live:

Check server liveness - is online

Input Parameter	Type	Required	Description
none			

Output Parameter	Type	Description
Response_Content	Text	A JSON value containing response information related to the request.
HTTP Status Code	Text	HTTP Status code returned in response.
Response Headers	Collection	Collection of headers data returned in response.

---

Ready:

Check server readiness - annotators are loaded and the server is ready to accept connections

Input Parameter	Type	Required	Description
none			

Output Parameter	Type	Description
Response_Content	Text	A JSON value containing response information related to the request.
HTTP Status Code	Text	HTTP Status code returned in response.
Response Headers	Collection	Collection of headers data returned in response.

## 3.2 Using the VBO

The **Stanford CoreNLP - Basic Actions VBO** is an additional asset that makes the usage of the Stamford CoreNLP API a little more Blue Prism process friendly. It includes all the same functionality of the Skill but with some added encapsulation, basic error handling. The VBO additionally calls the (/ready) action to verify the server is ready before each request so that the user can more appropriately handle retries and connection issues.

Below is a list of available actions from the **Stanford CoreNLP - Basic Actions VBO**:

Annotate Text:

This is the general action to annotate text using one of the provided "annotators" included in the Stanford CoreNLP API.

Input Parameter	Type	Required	Description
Options	Collection	Y	The collection of (Key=Value) pairs that represent the properties to pass to the CoreNLP server.
Text	Text	Y	The text to apply the annotators list.

Output Parameter	Type	Description
JSON	Text	A JSON value containing response information related to the request.
Collection	Collection	A JSON value containing response information related to the request.

Output Parameter	Type	Description
Status	Text	HTTP Status code returned in response.
Message	Text	Status Message returned by the request.

### TokenRegex:

A simple, rule-based NER over token sequences using Java regular expressions.. This can also be run as an annotator using Annotate action with the annotator 'tokensregex'

Input Parameter	Type	Required	Description
Options	Collection	Y	The collection of (Key=Value) pairs that represent the properties to pass to the CoreNLP server.
Text	Text	Y	The text to apply the annotators list.
Pattern	Text	Y	The TokensRegex pattern to annotate.

Output Parameter	Type	Description
JSON	Text	A JSON value containing response information related to the request.
Collection	Collection	A JSON value containing response information related to the request.
Status	Text	HTTP Status code returned in response.
Message	Text	Status Message returned by the request.

### Semgrex:

An implementation of the SemgrexPattern pattern for matching node and edge configurations of a dependency graph.

Input Parameter	Type	Required	Description
Options	Collection	Y	The collection of (Key=Value) pairs that represent the properties to pass to the CoreNLP server.
Text	Text	Y	The text to apply the annotators list.
Pattern	Text	Y	The TokensRegex pattern to annotate.

Output Parameter	Type	Description
JSON	Text	A JSON value containing response information related to the request.
Collection	Collection	A JSON value containing response information related to the request.
Status	Text	HTTP Status code returned in response.
Message	Text	Status Message returned by the request.

## Tregex:

A utility for matching patterns in trees, based on tree relationships and regular expression matches on nodes (the name is short for "tree regular expressions")

Input Parameter	Type	Required	Description
Options	Collection	Y	The collection of (Key=Value) pairs that represent the properties to pass to the CoreNLP server.
Text	Text	Y	The text to apply the annotators list.
Pattern	Text	Y	The TokensRegex pattern to annotate.

Output Parameter	Type	Description
JSON	Text	A JSON value containing response information related to the request.
Collection	Collection	A JSON value containing response information related to the request.
Status	Text	HTTP Status code returned in response.
Message	Text	Status Message returned by the request.

## Live:

Check server liveness - is online

Input Parameter	Type	Required	Description
none			

<b>Output Parameter</b>	<b>Type</b>	<b>Description</b>
Live	Boolean	The status of the server (online)