

TensorFlow Java API with Spring Framework



This page presents you a demo application what integrates the TensorFlow and the Spring frameworks together, creating an object detection web application in this way. It uses the TensorFlow Java API to run the frozen graphs with TensorFlow and Spring Framework to create the web user interface. In this example a YOLOv2 model was used to detect the objects on the uploaded pictures. Please read the previous article if you would like to find more details about the object detection part: [TensorFlow Java API example](#).

TensorFlow Java API with Spring Framework built by Gradle

On the image below you can see how the application looks like. On the web user interface it is possible to upload an image to detect objects on it. After running the object detection algorithm, the UI presents the input and the output image. Under the result you can see the list of the recognized objects and its probabilities e.g. in this example the person was recognized with 0.9429456 probability.

TensorFlow Java API with YOLO

Image: horses-and-woman.jpg has been successfully uploaded!

Input image	Result
	 <ul style="list-style-type: none">• horse - 0.95632315• person - 0.9429456• horse - 0.88018584

TensorFlow Java API with Spring Framework example project.

In the [live demo page](#) there is a limit for the uploaded images, they should have in **image/jpeg** format and their size cannot exceed the **300kb** limit. The optimal resolution for the images is **416x416**. If you upload an image with larger resolution, it will be rescaled to the mentioned resolution.

The source code of the project is available on Github: <https://github.com/szaza/tensorflow-java-examples-spring>.

Installation of the TensorFlow Java API with Spring Framework example project

Preconditions:

- Java JDK 1.8 or greater;
- TensorFlow 1.6 or greater;
- Git version control system;

Strongly recommended to install:

- [nVidia CUDA Toolkit 8.0](#) or higher version;
- [nVidia cuDNN](#) GPU accelerated deep learning framework;

Do the following steps to compile and run the project:

1. Clone the repository from Github: <https://github.com/szaza/tensorflow-java-examples-spring.git>;
2. Download the protobuff and label files from this link:
<https://drive.google.com/drive/folders/1GfS1Yle7Xari1tRUEi2EDYedFteAOaoN>;
3. Place them into the `/graph/YOLO/` directory;
4. Modify the [application.yml](#) configuration file in case if it is necessary;
5. The file upload limit can be increased also in the [application.yml](#) by changing the value of the `max-file-size` and `max-request-size`;
6. Build the application with Gradle by running the `./gradlew clean build` command from the root directory;
7. Start the server application with the `./gradlew bootRun` command;
8. Open the <http://localhost:8080> address in any browser and you have to see the application;