Viglet Turing ES
Developer Guide

Viglet Team

Version 0.3.6, 19-02-2022

Table of Content

Preface

Viglet Turing ES (https://viglet.com/turing) is an open source solution (https://github.com/openturing), which has Semantic Navigation and Chatbot as its main features. You can choose from several NLPs to enrich the data. All content is indexed in Solr as search engine.

Chapter 1. More Documentation

Technical documentation on Turing ES is available at https://docs.viglet.com/turing.

Chapter 2. Development Structure

2.1. Frameworks

Turing ES was developed using Spring Boot (https://spring.io/projects/spring-boot) for its backend.

The UI is currently using AngularJS (https://angularjs.org), but a new UI is being developed using Angular 12 (https://angular.io) with Primer CSS (https://primer.style/css).

In addition to Java, you also need to have Git (https://git-scm.com/downloads) and NodeJS (https://nodejs.org/en/download/) installed.

2.2. Databases

By default it uses the H2 database (https://www.h2database.com), but can be changed to other databases using Spring Boot properties. It comes bundled with OpenNLP (https://opennlp.apache.org/) in the same JVM.

2.3. Language and Deploy

It uses Java 14 (https://adoptopenjdk.net/archive.html?variant=openjdk14&jvmVariant=hotspot) and its deployment is done with Gradle 7.2 (https://gradle.org/) and works on Linux and Windows.

2.4. Docker

To use Semantic Navigation and Chatbot you must have a Solr (https://solr.apache.org) service available. If you prefer to work with all the services Turing depends on, you can use docker-compose (https://docs.docker.com/compose/install) to start these services, we use the Docker Desktop (https://www.docker.com/products/docker-desktop) installed on computer.

2.5. IDE

You can use Spring Tools 4 for Eclipse (https://spring.io/tools) or Eclipse (https://www.eclipse.org/downloads/) or Visual Studio Code (https://code.visualstudio.com/) or IntelliJ (https://www.jetbrains.com/pt-br/idea/) as IDEs.

Chapter 3. Download

Use the git command line to download the repository to your computer.

3.1. Turing Server and Connectors

```
git clone https://github.com/openturing/turing.git
```

3.2. Turing Java SDK

```
git clone https://github.com/openturing/turing-java-sdk.git
```

Chapter 4. Run during Development

To run Turing ES, execute the following lines:

4.1. Turing Server

4.1.1. Development

With UI

```
cd turing
./gradlew turing-app:bootrun
```

Without update UI

```
cd turing
./gradlew turing-app:bootrun -Pno-ui
```

4.1.2. New Turing UI

Start the Turing Server using dev-ui profile

```
cd turing
./gradlew turing-app:bootrun --args='--spring.profiles.active=dev-ui' -Pno-ui
```

And start one of the components of turing-ui

```
cd turing/turing-ui

## Console
ng serve console

## Search
ng serve sn

## Chatbot
ng serve converse
```

```
## Chatbot
ng serve welcome
```

IMPORTANT | You need start the Turing Server and Solr first.

4.1.3. Build

```
cd turing
./gradlew turing-app:build
```

4.2. Java SDK

4.2.1. Development

```
cd turing-java-sdk
./gradlew shadowJar
java -cp build/libs/turing-java-sdk-all.jar
com.viglet.turing.client.sn.sample.TurSNClientSample
```

IMPORTANT | You need start the Turing Server and Solr first.

4.3. Build

```
cd turing-java-sdk
./gradlew build
```

Or use the jitpack into your project at https://jitpack.io/#openturing/turing-java-sdk

4.4. WEM Listener

```
cd turing
./gradlew turing-wem:shadowJar
```

For development, copy the turing-wem/build/libs/turing-wem-all.jar into WEM_DIR/libs and test the listener using turing-wem command line.

| IMPORTANT | You need start the Turing Server and Solr first and restart WEM |

4.5. Database Connector

```
cd turing
./gradlew turing-jdbc:shadowJar
```

| IMPORTANT | You need start the Turing Server and Solr first and restart WEM |

4.6. Filesystem Connector

```
cd turing
./gradlew turing-filesystem:shadowJar
```

| IMPORTANT | You need start the Turing Server and Solr first and restart WEM |

4.7. Nutch

4.7.1. Nutch 1.12

```
cd turing/
./gradlew turing-nuch:nutch1_12:packageDistribution
```

For development, copy the files of turing-nutch/nutch1_12/build/extracted_dist to APACHE_NUTCH1_12/plugins/indexer-viglet-turing

4.7.2. Nutch 1.18

```
cd turing/
./gradlew turing-nuch:nutch1_18:packageDistribution
```

For development, copy the files of turing-nutch/nutch1_18/build/extracted_dist to APACHE_NUTCH1_18/plugins/indexer-viglet-turing

IMPORTANT	You need start the Turing Server and Solr first.

Chapter 5. Docker Compose

You can start the Turing ES using MariaDB, Solr and Nginx.

```
./gradlew turing-app:build -x test -i --stacktrace
docker-compose up
```

NOTE	If you have problems with permissions on directories, run `chmod -R 777 volumes`

5.1. Docker Commands

5.1.1. Turing

```
docker exec -it turing /bin/bash
```

5.1.2. Solr

```
docker exec -it turing-solr /bin/bash
```

Check logs

```
docker-compose exec turing-solr cat /opt/solr/server/logs/solr.log
# or
docker-compose exec turing-solr tail -f /opt/solr/server/logs/solr.log
```

5.1.3. MariaDB

```
docker exec -it turing-mariadb /bin/bash
```

5.1.4. Nginx

```
docker exec -it turing-nginx /bin/bash
```

Chapter 6. URLs

6.1. Turing Server

- Administration Console: http://localhost:2700. (admin/admin)

- Semantic Navigation Sample: http://localhost:2700/sn/Sample.

6.2. New Turing UI

- Welcome http://localhost:4200/welcome

- Console http://localhost:4200/console

- Search Page http://localhost:4200/sn/template?_setsite=Sample&_setlocale=en_US

- Converse http://localhost:4200/converse

6.3. Docker Compose

- Administration Console: http://localhost. (admin/admin)

- Semantic Navigation Sample: http://localhost/sn/Sample.

- Solr: http://localhost:8983

6.4. Code Quality

You can check the quality of Turing Code at:

- SonarCloud at https://sonarcloud.io/organizations/viglet-turing/projects

- Github Actions at https://github.com/openturing/turing/actions

- Github Security at https://github.com/openturing/turing/security/code-scanning

- Codecov at https://app.codecov.io/gh/openturing/turing

Chapter 7. Installation Modes

7.1. Turing ES Server

7.1.1. Simple.

Turing ES will be installed only using OpenNLP and H2 database embedded in Turing ES itself.

Prerequisites

1. Linux server

2. Java 14

3. 50Gb HDD

4. 2 Gb of RAM

Target Audience

Development and testing environment. Because it requires fewer components and lower memory usage.

Estimated Hours

2 hours

> **IMPORTANT** | Servers will be provided by the customer.

7.1.2. Docker Compose

Turing ES and its dependencies will be installed using Docker Compose script, including the following services:

- MariaDB – to store Turing ES system tables

- Solr – Used by Turing ES's Semantic Navigation and Chatbot

- Nginx – WebServer for Turing ES to use port 80

- Turing ES.

Prerequisites

1. Linux server

2. Docker and Docker Compose installed

3. 50Gb HDD

4. 4Gb of RAM

Target Audience

Customers who need more complex environments, but avoid the installation and configuration of each product. It can be used in an QA or Production environment.

Estimated Hours

16 hours

| **IMPORTANT** | Servers and docker configuration will be provided by the customer. |

7.1.3. Kubernetes

Turing ES and its dependencies will be installed using Kubernetes scripts, including the following services:

- MariaDB – to store Turing ES system tables

- Solr – Used by Turing ES's Semantic Navigation and Chatbot

- Nginx – WebServer for Turing ES to use port 80

- Turing ES.

Prerequisites

1. Linux Server with Kubernetes installed or Cloud that supports Kubernetes

2. 100Gb of Storage

3. 4Gb RAM

Target Audience

Customers who want to use cloud solutions like Google, AWS, Oracle, etc. It can be used in the production environment in a scalable way.

Estimated Hours

20 hours

> **IMPORTANT** Cloud infrastructure and servers will be provided by the customer.

7.1.4. Manual Installation of Services

The services will be installed individually on the servers following the Installation Guide procedure, which will include the following services:

- MariaDB – to store Turing ES system tables
- Solr – Used by Turing ES's Semantic Navigation and Chatbot
- Apache – WebServer for Turing ES to use port 80
- Turing ES.

Prerequisites

1. One Linux server or up to 4 Linux servers to install services
2. 50 - 100Gb of Storage for each server
3. Minimum 2Gb RAM for each Server
4. The services will be installed individually on the servers following the Installation Guide procedure.

Target Audience

Customers who prefer the on-premise structure and want to have the services installed directly on the servers. It can be used in Development, QA and Production.

Estimated Hours

20 hours

| **IMPORTANT** | Servers will be provided by the customer. |

7.2. Connectors

Turing ES has several connectors to allow you to index the contents in Semantic Navigation:

- Apache Nutch (Crawler)
- Wordpress
- OpenText WEM Listener
- FileSystem
- Database

7.2.1. Prerequisites

1. New linux server or existing server with content or files that will be indexed.
2. 50 of Storage for each server.

7.2.2. Estimated Hours

On average, it will take **16 hours** to configure the connector and have the first indexing version in Turing ES.

7.3. NLP

The customer can choose the NLP that will be used by Turing ES:

- Apache OpenNLP (Embedded)
- SpaCy NLP
- Stanford CoreNLP
- OpenText Content Analytics
- Poliglot

7.3.1. Prerequisites

1. Linux server

2. 50 of Storage for each server

3. Minimum 2 Gb of RAM

7.3.2. Estimated Hours

On average, it will take **4 hours** to configure NLP and configure Turing ES to use it.