



Turing AI

Viglet Turing AI ***Installation Guide***

Viglet Team

Version 0.3.4, 05-08-2021

Table of Content

Preface	1
1. Installing Java	2
2. Turing AI Requirements	3
2.1. Hardware Requirements	3
3. Installing and Configuring Databases	4
3.1. Turing AI Database Table Requirements	4
3.1.1. Oracle Database Requirements	4
3.1.2. Microsoft SQL Server Database Requirements	4
Configuring the SQL Server Database and Schemas	5
3.1.3. PostgreSQL Database Requirements	6
Installing and configuring the PostgreSQL Database	7
Configuring the Turing user, database, and schemas	7
4. Turing Utils	9
5. Solr Configuration	10
5.1. OOTB	10
5.2. OpenText InfoFusion	11
6. Installing Turing AI	13
6.1. Turing AI Download	13
6.2. Database	13
6.2.1. MySQL	13
6.2.2. Oracle Database	13
6.2.3. PostgreSQL	15
6.3. Encrypting the Database Password	15
6.4. Creating Turing AI Service on Linux	16
6.4.1. Red Hat and CentOS	16
6.5. Starting Turing AI	17
6.6. Accessing the Turing Console	18

Preface

Viglet Turing AI (<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. Installing Java

Turing AI requires Java.

NOTE Turing AI only supports Java 14.

During installation, Turing AI reads the `JAVA_HOME` and `PATH` variables to install the necessary services. Therefore, you must modify the variables as specified here:

- On Windows:
 - Set the `JAVA_HOME` system variable to point to the JAVA location.
 - Attach the `PATH` system variable with `%JAVA_HOME%/bin`.
- On Unix:
 - Create an environment variable called `JAVA_HOME`.
 - Provide the JAVA bin path in the `PATH` variable.

Chapter 2. Turing AI Requirements

The information below describes the resource modes available to run Turing AI.

NOTE Use a screen resolution of at least 1024 x 768 for optimum clarity.

This section describes the minimum and suggested features for systems on which you can configure Turing AI.

2.1. Hardware Requirements

Minimum	Recommended
Single 1.5 GHz processor 1 GB available memory	Dual 2 GHz processors with Hyper-Threading enabled 2 GB available memory

In addition, the following is required for all resource modes in Windows:

- 1GB of disk space for software.
- 300 MB additional disk space per component for registry and configuration files.
- One NTFS disk partition.
- A compatible version of a database.

Chapter 3. Installing and Configuring Databases

Turing AI supports multiple database platforms, each with specific unique requirements to maximize effectiveness with Turing AI. The following sections describe these requirements.

3.1. Turing AI Database Table Requirements

Turing AI uses multiple tables in multiple databases (also called database schemas by some vendors such as Oracle). You must create new databases / schemas that will contain the sets of tables described below.

3.1.1. Oracle Database Requirements

This section describes configuration requirements for an existing database.

To set up an Oracle database:

1. Install and configure a single database instance for use by Turing AI.
2. Create a database schema to contain the Turing AI system tables.
3. Configure the Turing AI admin user for the database. For Oracle, the database user with the grant of the CONNECT and RESOURCE roles. Make sure the user has UNLIMITED TABLESPACE and CREATE VIEW privileges.

3.1.2. Microsoft SQL Server Database Requirements

This section discusses the steps required to configure Microsoft SQL Server for your Turing AI databases.

Creating the Turing AI Database before you begin, be sure to refer to the following guides:

The following procedure describes how to create and configure a Microsoft SQL Server database for use by Turing AI.

When creating the Turing AI database, use the following options:

- You must install the Database Engine feature and Management Tools are recommended. All other features are optional.
- Set the Collation configuration variable to SQL_Latin1_General_CP1_CS_AS or Latin1_General / Case-sensitive / Accent-Sensitive.
- Set authentication mode to mixed mode.
- Turing AI system user cannot be sa.

Configuring the SQL Server Database and Schemas

This section discusses the steps required to create and configure a SQL Server database for use by Turing AI.

NOTE The database and usernames provided are examples only.

To create the SQL Server database:

1. Select Databases > New Database.
2. Enter the database name (eg turing) and the size of the data and log files according to the environment.
3. Go to the Options page and set the following values:
 - Collation = Latin1_General_CS_AS or SQL_Latin1_General_CP1_CS_AS
 - Miscellaneous > Allow Snapshot Isolation = True
 - Miscellaneous > Is Read Committed Snapshot On = True

```
USE master
GO
ALTER DATABASE turing
SET SINGLE_USER
WITH ROLLBACK IMMEDIATE
GO
ALTER DATABASE turing
SET ALLOW_SNAPSHOT_ISOLATION ON
GO
ALTER DATABASE turing
SET READ_COMMITTED_SNAPSHOT ON
GO
ALTER DATABASE turing
SET MULTI_USER
GO
```

1. Under Security > Logins, select New Login and set the following attributes:
 - Enter the Login name (for example, turing)
 - Deselect the Enforce Password Expiration option.
 - Set the Default database.
2. On the User Mapping page, set the following attributes for the turing database:
 - Default user = turing
 - Default schema = turing
 - With the turing database is selected, select the option for db_owner
3. Click OK.

3.1.3. PostgreSQL Database Requirements

This section discusses the steps required to configure PostgreSQL for your Turing AI databases:

- Installing and configuring the PostgreSQL Database
- Configuring the Turing AI user, database, and schema
- Installing and configuring the PostgreSQL Database

Installing and configuring the PostgreSQL Database

This section describes the steps required to install and configure a PostgreSQL database for your Turing AI environment. You will also need to configure the Turing user, database and schemas.

To install the PostgreSQL database:

1. Navigate to the following URL and follow the instructions provided for your supported version of PostgreSQL: <https://www.postgresql.org/docs/manuals/>

To configure the PostgreSQL instance:

- Open the postgresql.conf file (located in <PostgreSQL_installDir>\<version>\data\) and set the following parameters:

```
max_connections = 10
max_prepared_transactions = 10
shared_buffers = 512MB
autovacuum = off
```

IMPORTANT

The autovacuum daemon is used to reclaim memory/storage space the database is no longer using. You must re-enable this utility after you have finished installing and configuring Turing AI.

NOTE

Configuration values will vary for higher transaction environments and numbers of users. max_prepared_transactions should be at least as large as max_connections so that every session can have a prepared transaction pending.

Configuring the Turing user, database, and schemas

This section discusses the steps required to configure the users, database and schemas for your Turing AI environment.

NOTE

You must be a PostgreSQL super-user or have createuser privileges to create Turing users. To create and configure the Turing AI database, you must be a super-user or have create privileges.

IMPORTANT

When you create a database table in PostgreSQL, the field name must be lowercase.

To configure the Turing AI user:

1. Create a user as the owner of the Turing AI database:

```
CREATE USER turing PASSWORD <password> with createuser;
```

1. Create a Turing AI database:

```
CREATE DATABASE turing_db;  
ALTER DATABASE turing_db OWNER TO turing;
```

To create and access the Turing AI user and schema:

- Log in to the turing (database) using the turing account. For example, on the operating system command prompt:

```
$psql -U turing -d turing_db
```

- Create the Turing schema and grant ownership to the Turing user:

```
CREATE SCHEMA turing;  
ALTER SCHEMA turing OWNER TO turing;
```

- Set the PostgreSQL search_path variable to the user to allow direct access to the respective schemas:

```
ALTER USER turing SET SEARCH_PATH = turing;
```

Chapter 4. Turing Utils

First you need install Turing Utils, that contains sample configurations and script to facilitate the use of Turing AI, so go to <https://viglet.org/turing/download/> and click on "Integration > Utils" link to download it.

Extract the turing-utils.zip to /appl/viglet/turing/utils

```
mkdir -p /appl/viglet/turing/utils  
unzip turing-utils.zip -d /appl/viglet/turing/utils
```

Chapter 5. Solr Configuration

You need to configure Solr to be able to store Semantic Navigation and ChatBot data.

5.1. OOTB

With Solr running, go to Turing Utils and create the new cores that will be used by Turing AI.

1. Install the cores into solr:

```
cd <SOLR_DIR>/bin
## For Semantic Navigation
./solr create_collection -c turing -n turing -d
/appl/viglet/turing/utils/solr/<VERSION>/turing/<LANGUAGE>
## For ChatBot
./solr create_collection -c converse -n converse -d
/appl/viglet/turing/utils/solr/<VERSION>/converse/<LANGUAGE>
```

2. Started Solr Service

5.2. OpenText InfoFusion

1. Create the cores into Solr:

```
cd <SOLR_DIR>/bin
## For Semantic Navigation
mkdir <SOLR_DIR>/server/solr/turing
cp -Rf /appl/viglet/turing/utils/solr/<VERSION>/turing/<LANGUAGE>/
<SOLR_DIR>/server/solr/turing/
## For ChatBot
mkdir <SOLR_DIR>/server/solr/converse
cp -Rf /appl/viglet/turing/utils/solr/<VERSION>/turing/<LANGUAGE>/
<SOLR_DIR>/server/solr/converse/
```

2. If the core name of Semantic Navigation Site will be different, you need modify the `core.properties` file and change the `name` and `collection` attributes, for example:

```
cd <SOLR_DIR>/bin
mkdir <SOLR_DIR>/server/solr/sample
cp /appl/viglet/turing/utils/solr/<VERSION>/turing/<LANGUAGE>/
<SOLR_DIR>/server/solr/sample/
chown otif:otif <SOLR_DIR>/server/solr/sample

## Modify the name and collection fields
sed -i 's/turing/sample/g' <SOLR_DIR>/server/solr/sample/core.properties

cat <SOLR_DIR>/server/solr/sample/core.properties

numShards=1
name=sample
shard=shard1
collection=sample
coreNodeName=core_node1
```

3. Edit the `<IF_API>/bin/otif.sh` and add the new cores to sync with Zookeeper.

```
# Push Solr configurations in ZooKeeper
pushSolrConfigs() {
    ...
    cd "$BASE_DIR"
    # For Semantic Navigation
    ./zk.sh -cmd upconfig -zkhost "$SOLR_ZK_ENSEMBLE" -confdir
"<IF_SOLR>/server/solr/turing/conf" -confname turing
    ./zk.sh -cmd upconfig -zkhost "$SOLR_ZK_ENSEMBLE" -confdir
"<IF_SOLR>/server/solr/<OTHER_CORE>/conf" -confname <OTHER_CORE>
    # For ChatBot
    ./zk.sh -cmd upconfig -zkhost "$SOLR_ZK_ENSEMBLE" -confdir
"<IF_SOLR>/server/solr/converse/conf" -confname converse
    # Default InfoFusion cores (Do not modify)
    ./zk.sh -cmd upconfig -zkhost "$SOLR_ZK_ENSEMBLE" -confdir "
$BASE_DIR/../../etc/solr/otif/en/conf" -confname otif_en
    ...
    cd "$CURRENT_DIR"
}
```

4. With Zookeeper running, execute the command below:

```
cd <IF_API>/bin
./otif.sh push-solr-configs
```

5. Restart the Solr.

Chapter 6. Installing Turing AI

6.1. Turing AI Download

Go to <https://viglet.org/turing/download/> and click on "Download Turing AI" button to download the viglet-turing.jar executable.

Copy the viglet-turing.jar to /appl/viglet/turing/server

```
mkdir -p /appl/viglet/turing/server
cp viglet-turing.jar /appl/viglet/turing/server
chmod 770 /appl/viglet/turing/server/viglet-turing.jar
```

6.2. Database

By default, Turing uses H2 as its embedded database, but if you prefer another database you can follow the procedure below, creating `/appl/viglet/turing/server/viglet-turing.conf` file with following lines:

6.2.1. MySQL

Below is an example of how to configure Turing for MySQL.

viglet-turing.conf for MySQL

```
JAVA_OPTS="-Xmx1g -Xms1g
-Dspring.datasource.url=jdbc:mariadb://localhost:3306/turing
-Dspring.datasource.username=turing -Dspring.datasource.password=turing
-Dspring.datasource.driver-class-name=org.mariadb.jdbc.Driver
-Dspring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
"
```

6.2.2. Oracle Database

Below is an example of how to configure Turing for Oracle Database.



viglet-turing.conf for Oracle Database

```
JAVA_OPTS="-Xmx1g -Xms1g  
-Dspring.datasource.url=jdbc:oracle:thin:@localhost:1521/turing  
-Dspring.datasource.username=turing -Dspring.datasource.password=turing  
-Dspring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver  
-Dspring.jpa.properties.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect"
```


6.2.3. PostgreSQL

Below is an example of how to configure Turing for PostgreSQL.

viglet-turing.conf for PostgreSQL

```
JAVA_OPTS="-Xmx1g -Xms1g  
-Dspring.datasource.url=jdbc:postgresql://localhost:5432/turing  
-Dspring.datasource.username=turing -Dspring.datasource.password=turing  
-Dspring.datasource.driver-class-name=org.postgresql.Driver  
-Dspring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQL94Dialect  
"
```

6.3. Encrypting the Database Password

For more security, it is possible to encrypt the Database password in `viglet-turing.conf` following the procedure below:

```
cd /appl/viglet/turing/utils/scripts  
./turGenerateDBPassword.sh /appl/viglet/turing/server  
Password:  
Retype Password:  
  
Thanks, the password was successfully generated in /appl/viglet/turing/server/db-  
encrypted.properties  
  
cat /appl/viglet/turing/server/db-encrypted.properties  
clqp6QxceJQIp6lnaiKelQc9DvdNKNxg
```

Copy the content of `/appl/viglet/turing/server/db-encrypted.properties` and replace the
`-Dspring.datasource.password=<PLAIN_TEXT_PASSWORD>` value of
`/appl/viglet/turing/server/viglet-turing.conf` file with
`-Dspring.datasource.password=ENC(<ENCRYPTED_PASSWORD>).`

For example: `-Dspring.datasource.password=ENC(clqp6QxceJQIp6lnaiKelQc9DvdNKNxg)`

6.4. Creating Turing AI Service on Linux

6.4.1. Red Hat and CentOS

As root, create a `/etc/systemd/system/turing.service` file with the following lines:

```
[Unit]
Description=Viglet Turing AI
After=syslog.target

[Service]
User=viglet
ExecStart=/appl/viglet/turing/server/viglet-turing.jar
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

This way, you can interact with the Turing AI service:

```
./systemctl start turing.service
./systemctl stop turing.service
./systemctl restart turing.service
./systemctl status turing.service
```




```
Downloading pt-pos-maxent.bin Model...[ OK ]
Path: /appl/viglet/turing/server/models/opennlp/pt/pt-pos-perceptron.bin
Downloading pt-pos-perceptron.bin Model...[ OK ]
Path: /appl/viglet/turing/server/models/opennlp/pt/pt-sent.bin
Downloading pt-sent.bin Model...[ OK ]
Path: /appl/viglet/turing/server/models/opennlp/pt/pt-token.bin
Downloading pt-token.bin Model...[ OK ]
OpenNLP Models downloaded...
Configuration finished.
Viglet Turing started
```

6.6. Accessing the Turing Console

Turing provides remote access to administration, configuration, and management through its Web application interfaces. Once setup is complete, the Console become browser-accessible through the following URL: <http://<host>:<port>/console> where <host>:<port> is the listening host and port for the Turing AI that you created in the application server. The default port is 2700 for the Console.

By default the login/password are: admin/admin