

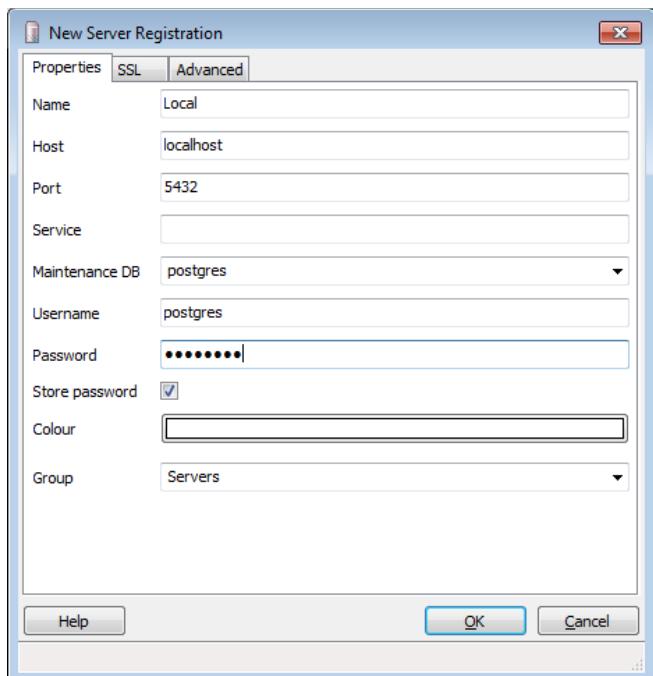
## TD-Tracker Installation Guide

The first step to install TDR is related to database creation.

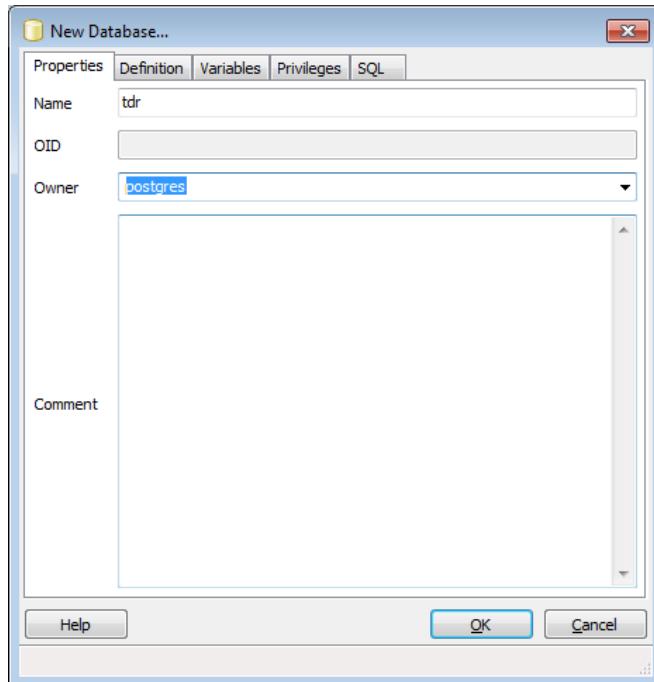
It is necessary to create TDR database in Postgres (currently supported database) to install the application. The installation is described below.

### Installing Postgres Database (after install postgres on your equipment):

- 1) Open your pgAdmin tool
- 2) Create a server on your local equipment



- 3) Connect to the created server.
- 4) Right-click on the server -> New Database
- 5) Name this new database as TDR and put as Owner postgres user with postgres password (default).



After installing postgres and configure the database it is time to run the application, following the steps bellow:

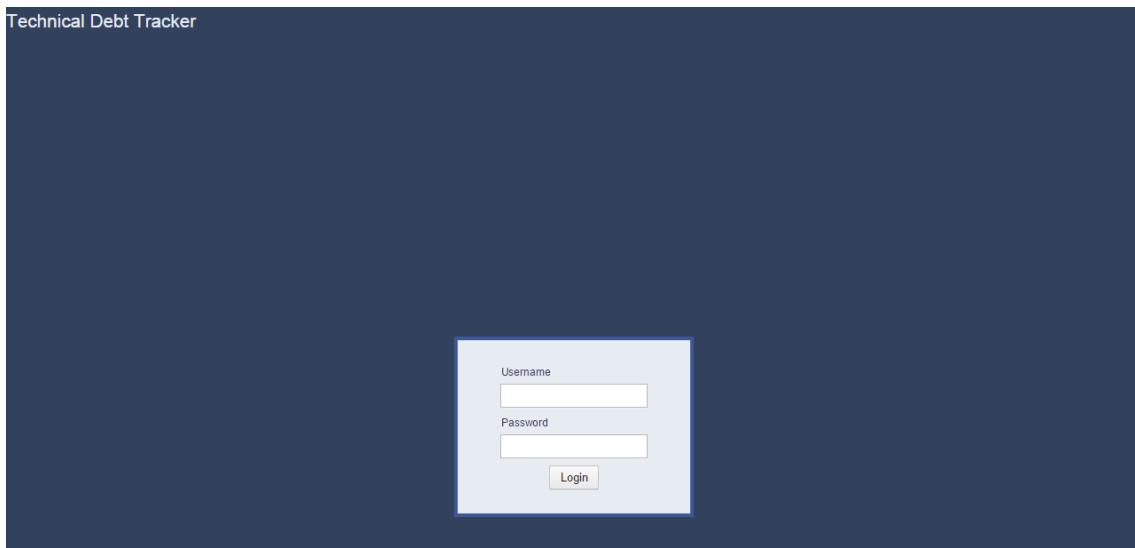
## Installing application

- 1) Download and save tdr.war file in any folder on your computer
- 2) Open the command prompt and locate the folder with tdr.war
- 3) Execute command java -jar tdr.war

```
C:\Windows\system32\cmd.exe - java -jar tdr.war
D:\tdr>java -jar tdr.war
:: Spring Boot ::      (v1.1.5.RELEASE)

2015-08-07 18:47:29.282 INFO 5860 --- [           main] br.unesp.tdr.Application          : Starting Application on notelucas with PID 5860
2015-08-07 18:47:29.374 INFO 5860 --- [           main] o.s.w.e.EmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@53333333: startup date [2015-08-07T18:47:29.374+0000]; root of context hierarchy
2015-08-07 18:47:31.203 INFO 5860 --- [           main] o.s.b.f.s.DefaultListableBeanFactory : Overriding bean definition for bean 'org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WebMvcAutoConfiguration$HandlerMethodArgumentResolverFilterRegistrationBean': was [org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WebMvcAutoConfiguration$HandlerMethodArgumentResolverFilterRegistrationBean@53333333]; is [org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WebMvcAutoConfiguration$HandlerMethodArgumentResolverFilterRegistrationBean@53333333]
2015-08-07 18:47:31.903 INFO 5860 --- [           main] o.s.b.f.s.DefaultListableBeanFactory : Overriding bean definition for bean 'org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WebMvcAutoConfiguration$HandlerMethodReturnValueHandlerFilterRegistrationBean': was [org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WebMvcAutoConfiguration$HandlerMethodReturnValueHandlerFilterRegistrationBean@53333333]; is [org.springframework.boot.autoconfigure.web.WebMvcAutoConfiguration$WebMvcAutoConfiguration$HandlerMethodReturnValueHandlerFilterRegistrationBean@53333333]
log4j:WARN No appenders could be found for logger (org.jboss.logging).
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
2015-08-07 18:47:32.445 INFO 5860 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.interceptor.TransactionInterceptor' of type [org.springframework.transaction.interceptor.TransactionInterceptor] is not managed by the current BeanFactory. This means that this bean will not be destroyed when its containing application context is closed.
2015-08-07 18:47:32.649 INFO 5860 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.interceptor.TransactionAttributeSource' of type [org.springframework.transaction.interceptor.TransactionAttributeSource] is not managed by the current BeanFactory. This means that this bean will not be destroyed when its containing application context is closed.
2015-08-07 18:47:32.679 INFO 5860 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.interceptor.TransactionInterceptor' of type [org.springframework.transaction.interceptor.TransactionInterceptor] is not managed by the current BeanFactory. This means that this bean will not be destroyed when its containing application context is closed.
2015-08-07 18:47:33.025 INFO 5860 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.interceptor.TransactionAttributeSource' of type [org.springframework.transaction.interceptor.TransactionAttributeSource] is not managed by the current BeanFactory. This means that this bean will not be destroyed when its containing application context is closed.
2015-08-07 18:47:33.310 INFO 5860 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.interceptor.TransactionInterceptor' of type [org.springframework.transaction.interceptor.TransactionInterceptor] is not managed by the current BeanFactory. This means that this bean will not be destroyed when its containing application context is closed.
2015-08-07 18:47:33.520 INFO 5860 --- [           main] o.apache.catalina.core.StandardService : Starting service Tomcat
2015-08-07 18:47:33.580 INFO 5860 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.0.22
2015-08-07 18:47:38.482 INFO 5860 --- [lost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext: initialization started
2015-08-07 18:47:38.483 INFO 5860 --- [lost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization started
2015-08-07 18:47:38.711 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Registering servlet of type [org.vaadin.viritin.servlet.VaadinServlet]
2015-08-07 18:47:38.714 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Servlet will be mapped to URL [/UAAIDI]
2015-08-07 18:47:38.720 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Servlet was not found in the application
2015-08-07 18:47:38.724 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Looking for servlet init parameters
2015-08-07 18:47:38.724 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Found no servlet init parameters
2015-08-07 18:47:38.757 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Registering servlet of type [org.vaadin.viritin.servlet.VaadinServlet]
2015-08-07 18:47:38.757 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Servlet will be mapped to URL [/UAAIDI]
2015-08-07 18:47:38.763 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Servlet was not found in the application
2015-08-07 18:47:38.764 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Looking for servlet init parameters
2015-08-07 18:47:38.765 INFO 5860 --- [lost-startStop-1] o.v.s.b.c.UaadinServletConfiguration : Found no servlet init parameters
```

4) TD-Tracker will start and use the url <http://localhost:8080/> to start.



5) To the first login use the installing user "admin@tdr.com" password "admin".

## Creating Integration Connection

- 1) Click Integration Connection on left side menu
- 2) A list with all created connections will be displayed.
- 3) It is possible to edit or create a new connection

A screenshot of the 'Integration Connection' list screen. The left sidebar has a dark blue background with white text and icons. It shows the user 'admin@tdr.com' and a dropdown arrow. Below that are links: 'Dashboard', 'Technical Debt', 'Integrations' (which is underlined), 'Integration Connection', 'Integration Sync', 'Users', and 'Users'. The main area has a light gray background. At the top, it says 'Integration Connection' and has a green 'New' button with a '+' icon. Below that is a table with two columns: 'ID' and 'Alias'. There are two rows: row 2 with alias 'GH\_ElasticSearch' and row 3 with alias 'Sonar\_OpenRefine'. Each row has edit ('pencil') and delete ('X') icons in the last column.

## GitHub Connection

- 1) For GitHub connection all fields above are required.
- 2) Use a valid user , password and project(existing in gitHub)

A screenshot of the 'Integration Connection' creation form. The left sidebar is the same as the previous screenshot. The main area has a light gray background. It starts with a heading 'Integration Connection'. Below that are five input fields: 'Alias' (with value 'GH\_ElasticSearch'), 'Integration Type' (a dropdown menu with 'GITHUB' selected), 'Project' (with value 'elastic/elasticsearch'), 'User' (with value 'borante'), and 'Password' (with value '\*\*\*\*\*'). Below these fields is a 'Test Connection' button. At the bottom are 'Save' and 'Cancel' buttons.

## Database Integration Connection

- 1) For database direct connection all fields above are required.
- 2) After fulfill and test the connection save the integration
- 3) Use connection table/fields to map the relationship, after manually inform the table and field, select the existent field for the relationship.
- 4) Click on Add Relationship and the field will be stored on the grid bellow.

Technical Debt Tracker

admin@ldr.com

Dashboard

Technical Debt

Integrations

Integration Connect...

Integration Sync

Users

Users

Integration Connection

Alias: Sonar\_OpenRefine

Integration Type: DATABASE

Server: localhost

Database: sonar

User: postgres

Password: \*\*\*\*\*

Test Connection

Connection Table/Fields

Table	Foreign Field	Internal Field	
issues	rule_id	EXTERNAL_ID	X
issues	message	DESCRIPTION	X
issues	line	LOCATION	X

Add Relationship

Check Results

Save Cancel

- 5) After insert a relationship you can check results to see if data is being retrieved. Click "Check Results" button and expand the presented dialog.

### Integration Check Results

Integration Query

```
SELECT rule_id as EXTERNAL_ID, message as DESCRIPTION, line as LOCATION, " as TITLE, " as TYPE, " as RESPONSIBLE, " as ESTIMATED_PRINCIPAL, " as ESTIMATED_INTEREST_AMOUNT, " as ESTIMATED_INTEREST_PROBABILITY from issues
```

Title Location Description Responsible

		Replace all tab characters in this file by sequences of white-spaces.	
		Replace all tab characters in this file by sequences of white-spaces.	
		Replace all tab characters in this file by sequences of white-spaces.	
		Replace all tab characters in this file by sequences of white-spaces.	
22		A atribuição ao parâmetro 'local' não é permitida.	
		Replace all tab characters in this file by sequences of white-spaces.	
60		'5' é um número mágico.	
		Replace all tab characters in this file by sequences of white-spaces.	
8		Rename this constant name to match the regular expression "[A-Z][A-Z0-9]*_[A-Z0-9]+\\$".	
		Replace all tab characters in this file by sequences of white-spaces.	
6		Utility classes should not have a public or default constructor.	
		Replace all tab characters in this file by sequences of white-spaces.	
		Replace all tab characters in this file by sequences of white-spaces.	
38		Rename this generic name to match the regular expression '[A-Z]\$.	
51		The String literal "unchecked" appears 7 times in this file; the first occurrence is on line 51	

## Using Integration Connection to Synchronize

- 1) Click Integration Sync on left side menu
- 2) A list with all created connections will be displayed.
- 3) Click on right button to start the synchronization.
- 4) Select all desired registers (candidates) that really are Technical Debt and move from left grid to right grid.
- 5) Use the fields bellow to complete the information from candidates list which are not presented or in a case of information to be overwritten.
- 6) Save to save all selected items existing in the right grid.

The screenshot shows the 'Integration Synchronization' page. On the left, a sidebar lists 'Technical Debt Tracker' with a user icon and 'admin@tdr.com'. Below it are links: Dashboard, Technical Debt, Integrations, Integration Connect..., Integration Sync, and Users. The main area has a title 'Integration Synchronization'. It contains two tables. The left table has columns 'Title', 'Locator', and 'Description'. The right table has columns 'Title', 'Location', 'Description', and 'Response'. Both tables have many rows of technical debt candidates. At the bottom, there are filters for 'TD Type', 'Responsible', 'Is Intentional?', and buttons for 'Save', '<<', '<->', and '>>'. A scroll bar is visible between the two tables.

## Creating a manual TD

- 1) Click Technical Debt on left side menu
- 2) A list with all existing TD of your database is presented

The screenshot shows the 'Technical Debt' page. The sidebar is identical to the previous one. The main area has a title 'Technical Debt' and a green 'Add Technical Debt' button. Below it is a table with columns 'Responsible', 'Type', and 'Title'. One row is shown: 'br.unesp.tdr.entity.User@6fc4dbac', 'DEFECT', and 'Example of TD'. To the right of the table are edit ('pencil') and delete ('x') buttons. At the bottom, there are pagination controls for 'Items per page' (set to 10), 'Page' (set to 1), and navigation arrows.

- 3) It is possible to edit with "pencil" blue button, exclude with "x" red button and create a new manual TD with "Add Technical Debt" button.
- 4) Click on "Add Technical Debt" button on superior left side of the screen to go to Technical Debt Register form.

5) Fulfill all required fields

6) Click on "Save" button

Technical Debt Tracker

admin@tdr.com ~

Dashboard

Technical Debt

Technical Debt

Integrations

Integration Connecti...

Integration Sync

Users

Users

Technical Debt Register

Title	Example of TD
Identification Date	<input type="text"/>
Responsible	Administrator
TD Type	DEFECT
Description	Description example
Project	
Location	LoginUI
Is Intentional?	
Estimated Principal	LOW
Estimated Interest Amount	LOW
Estimated Interest Probability	LOW
Fixed By	
Fixed Date	<input type="text"/>
Realized Principal	
Realized Interest Amount	

**Save** **Cancel**

The screenshot shows a user interface for managing technical debt. On the left is a dark sidebar with navigation links: Dashboard, Technical Debt, Integrations, Integration Connecti..., Integration Sync, and Users. The main area is titled 'Technical Debt Register' and contains a form with the following fields:

- Title: Example of TD
- Identification Date: (input field)
- Responsible: Administrator
- TD Type: DEFECT
- Description: Description example
- Project: (input field)
- Location: LoginUI
- Is Intentional?: (dropdown menu)
- Estimated Principal: LOW
- Estimated Interest Amount: LOW
- Estimated Interest Probability: LOW
- Fixed By: (input field)
- Fixed Date: (input field)
- Realized Principal: (input field)
- Realized Interest Amount: (input field)

At the bottom right of the form are 'Save' and 'Cancel' buttons.