

This is unreleased documentation for iDempiere **Next** version.

For up-to-date documentation, see the [latest version](#) (10.0.0).

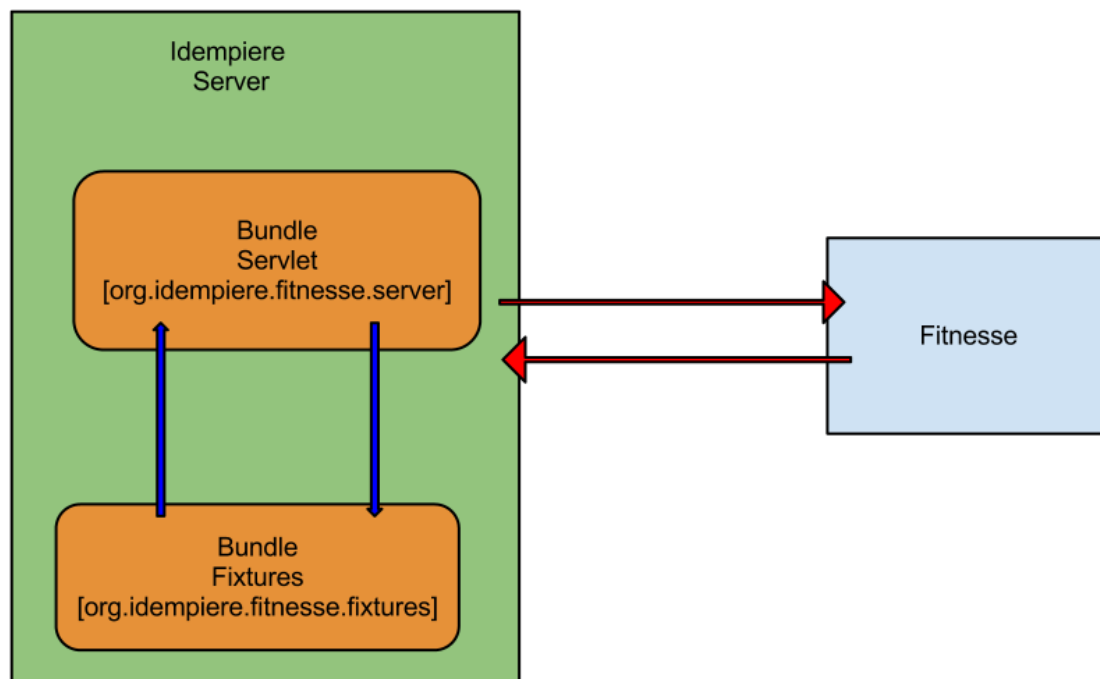
[Dev Guides](#)[Setup for automated tests with Fitnessse](#)**Version: Next**

# Setup for automated tests with Fitnessse

## Table of Content

**Goal:** Functional Testing

**Sponsor:** [Sponsor:Trek\\_Global|Trek Global](#)



# Description:

Allow user and developer create and execute fitnesse tests withing the idempiere environment.

***Please, be aware that this documentation has changed significantly from earlier versions to iDempiere Vers. 10.***

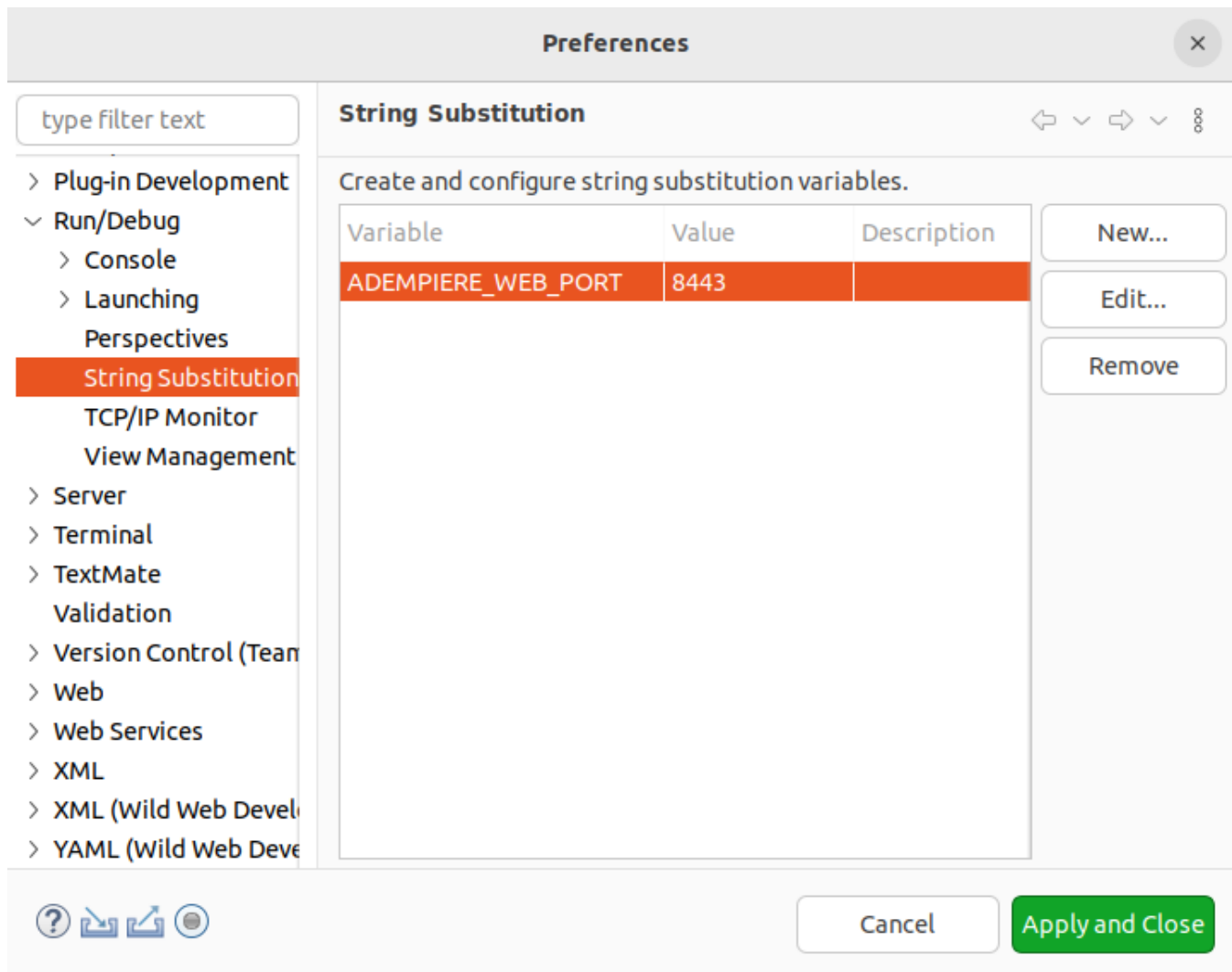
Two different cases can be distinguished:

- Running fitnesse from eclipse
- Running fitnesse on a server from comand line

## Running fitnesse from eclipse

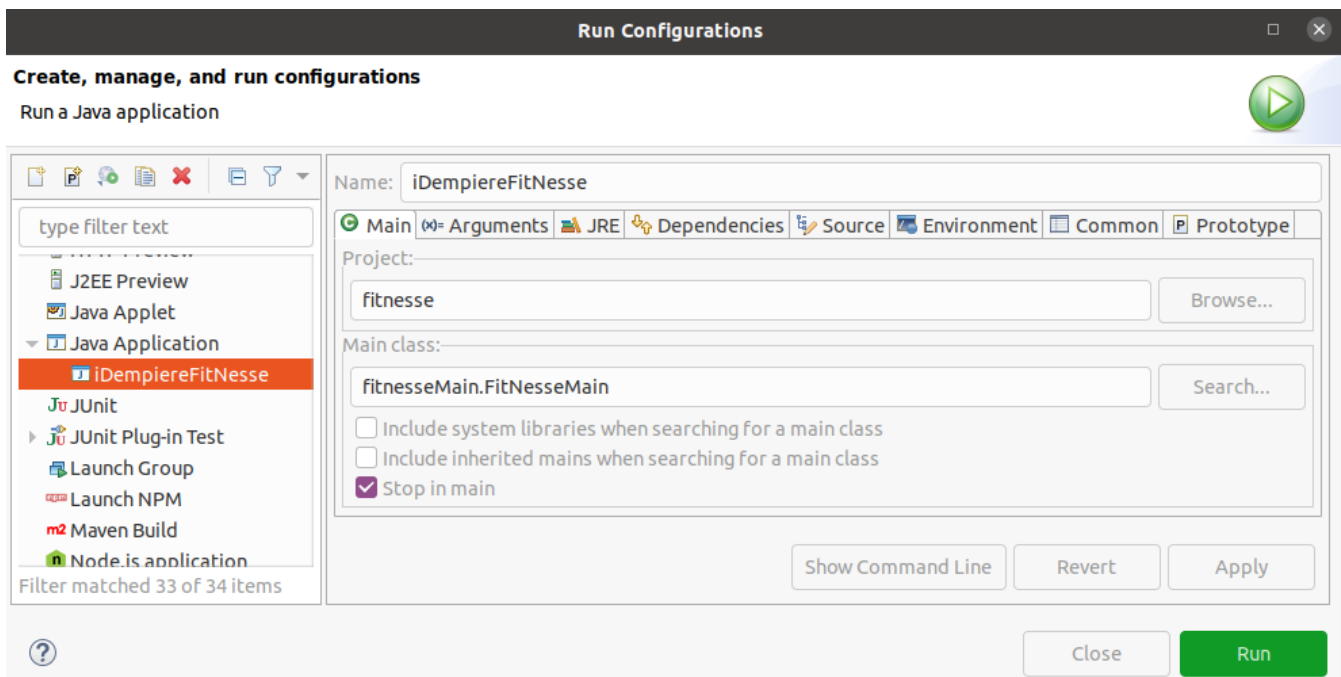
- Prerequisiste is an installed development environment.
- Get the "\"fitnesse\" java project which fits to your release into your eclipse workspace. You will find the sources on github. <https://github.com/idempiere/idempiere-fitnessse>
- following folder structure is necessary for iDempiere Dev and the fitnesse part (github standards must be renamed!):
  - idempiere
  - idempiere-fitnessse
    - org.idempiere.fitnessse.\*
    - org.idempiere.fitrecorder
    - org.idempiere.ui.zk.selenium
- Before importing the Fitnessse-plugin into the iDempiere Workspace you must run "mvn verify" in the idempiere-fitnessse folder !
  - To avoid Tycho/Maven problems, remove ~/.m2/repository/.meta/p2-artifacts.properties prior to running maven build for source using Tycho 2.6 and below, i.e. with: rm ~/.m2/repository/.meta/p2-artifacts.properties (rm ~ is helpful cause the file is difficult to find otherwise)
- Import - Existing Projects into Workspace + there tick "Search for nested Projects" (see image below)
- Set ADEMPIERE\_WEB\_PORT StringSubstitution to right Port with SSL e.g. 8443 (older versions work with 8080)
  - The value is set in Eclipse - Preferences - Run/Debug - String Substitution
  - The former port mentioned in older and following screenshots does not work anymore as SSL is mandatory for iDempiere.
  - Use 8443 or your selected SSL-Port for iDempiere Access in setup for String Substitution

!!!:



## Run IDempiere Fitnesse Bundles

- First you have to define the Fitnesse Application in the Eclipse - Configurations

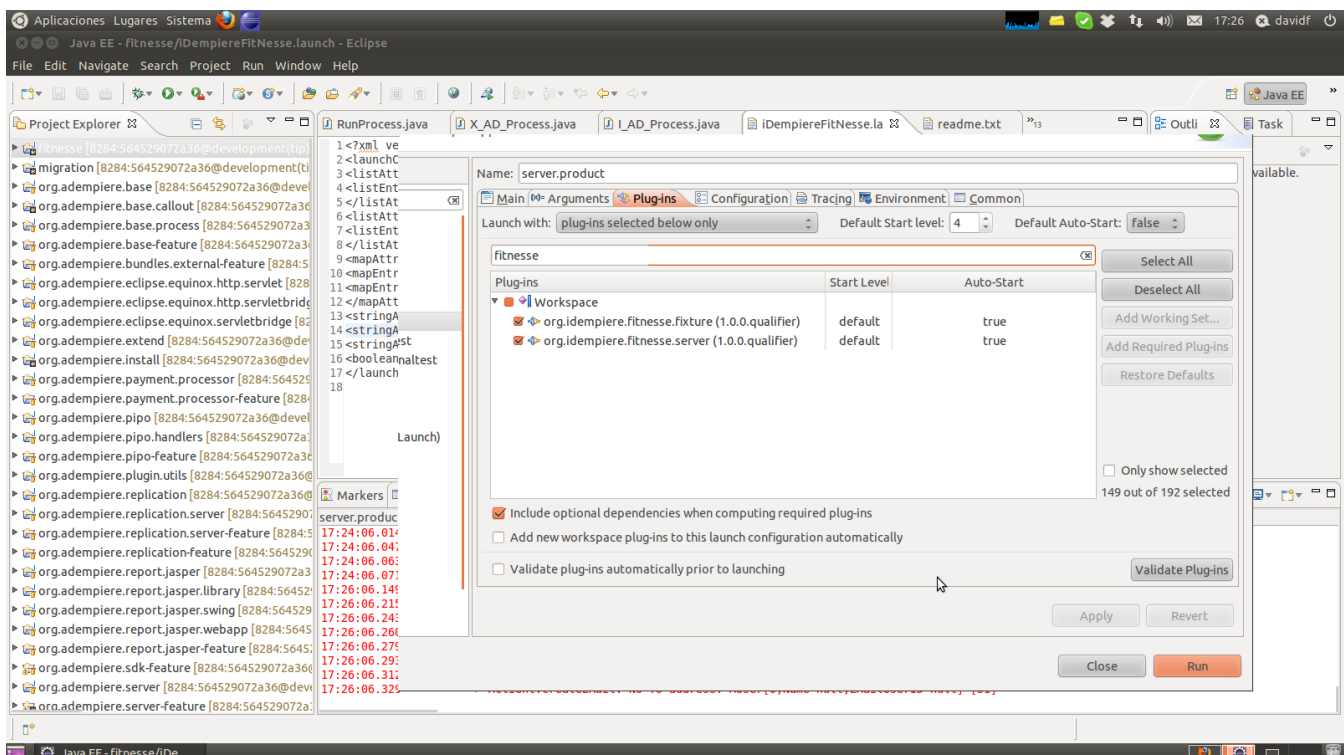


- Use \"Run\" for starting the fitnesse component. Debug was required in older versions.
- Verify fitnesse started ok.

FitNesse (v20111026) Started...

```
port: 8089
root page: fitnesse.wiki.FileSystemPage at ./FitNesseRoot
logger: /home/hengsin/workspace/idempiere-trekglobal/fitnesse/log
authenticator: fitnesse.authentication.PromiscuousAuthenticator
html page factory: fitnesse.html.HtmlPageFactory
page version expiration set to 14 days.
```

- Please start the bundles at the idempiere server\'s osgi as shown below in the screenshot:



## Run iDempiere Fitnessse Bundles

Congratulations finished! Now you can start using Fitnessse with iDempiere e.g. start some of the TestCases already prepared for Login or a new Business partner.

Some special cases are:

- If your server is running with a self-signed certificate, you must match the certificates in the respective cacert and key.pem. You find some commands to do this on one computer containing both iDempiere and Fitnessse-server below: [Installing Fitnessse.HowTo#Allow SSL-Connection for self signed certificates](#)
- Sometimes you must start the fitnessse OSGI-bundles by hand the first time they are used:
  - i. find out the numbers of the OSGI-plugins and then
  - ii. start the plugins for fitnessse (those before set to auto start) (compare comands on OSGI console below:)

1. On eclipse console write :

```
osgi> ss
```

Look for idempiere services, for example:

```
127    <<LAZY>>    org.idempiere.fitnessse.fixture_1.0.0.qualifier
128    <<LAZY>>    org.idempiere.fitnessse.server_1.0.0.qualifier
```

2. And start the services related:

```
osgi> start 127
```

```
osgi> start 128
```

- After that normally they will start automatically.

## Running iDempiere Fitnessse server From Command Line

You can also set up Fitnessse on any server e.g. on a normal test, other iDempiere server or even access iDempiere on another system.

For this you need to do in preparation:

- Build the Fitnessse for server from github

- Ensure you have SSL-connection between the Fitnessse and the iDempiere server (with public certificates this works out of the box. Another option for private keys will be outlined below).

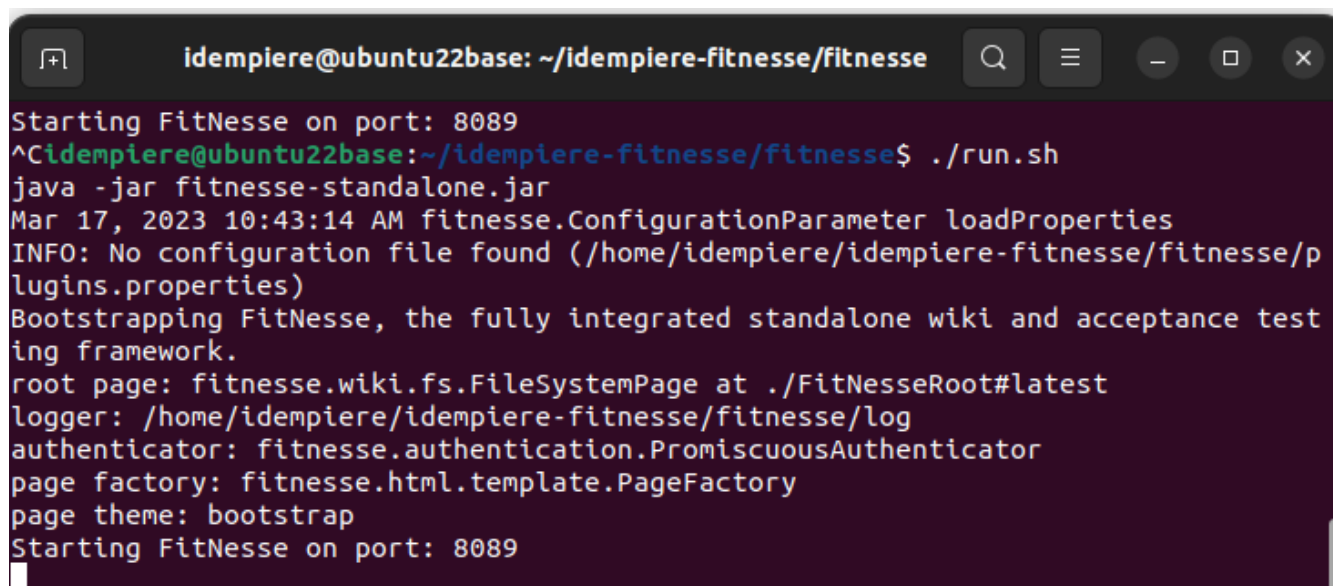
## Build Fitnessse from github

- Basic setup also in idempiere and idempiere-fitnessse folders directly on top of user's home (for links and folders see description on Dev)
  - Get the \"fitnessse\" java project which fits to your release into your eclipse workspace.  
<https://github.com/idempiere/idempiere-fitnessse>
  - following folder structure is necessary for iDempiere Dev and the fitnessse part (github standards must be renamed!):
    - idempiere
    - idempiere-fitnessse
  - You must run \"mvn verify\" in both folders
    - a. idempiere
    - b. idempiere-fitnessse
  - To avoid Tycho/Maven problems, remember to remove ~/.m2/repository/.meta/p2-artifacts.properties prior to running maven build, i.e. with: rm ~/.m2/repository/.meta/p2-artifacts.properties
- Then the server is changed to a running iDempiere-server in the folder \"idempiere\" (perhaps other options are easier??)
- command in fitnessse path:
  - ant build (ant must be installed before)
- Now move the parameters from the iDempiere-server to the Fitnessse-server
  - Copy deploy-fitnessse.sh (from idempiere-fitnessse) to main path of iDempiere-server
  - Execute deploy.sh and direct the output to .../org.idempiere.fitnessse.p2/target/repository e.g. ./deploy-fitnessse.sh <file:/home/idempiere/idempiere-fitnessse/org.idempiere.fitnessse.p2/target/repository>

If your server is running with self-signed certificates, please, you must match the certificates in the respective cacert and key.pem. You find some commands to do this on one computer containing both iDempiere and Fitnessse-server below: [Installing Fitnessse.HowTo#Allow SSL-Connection for self signed certificates](#)

## Start fitnesse server from command line

- The script run.sh sets the variable ADEMPIERE\_WEB\_PORT based on a file idempiereEnv.properties located on the folder above fitnesse\_home
  - In case the properties file is not copied automatically by the deploy.sh, copy it from idempiere-Folder to idempiere-fitness-Folder manually
  - Anycase make sure that the contents of the file for ADEMPIERE\_WEB\_PORT-Parameter (from Version 10 with SSL prescribed) is ADEMPIERE\_WEB\_PORT=8443
- Open a Terminal and go to Idempiere Fitness Location in your Computer (.... idempiere-fitness/fitness)
- Start Fitness executing: ./run.sh

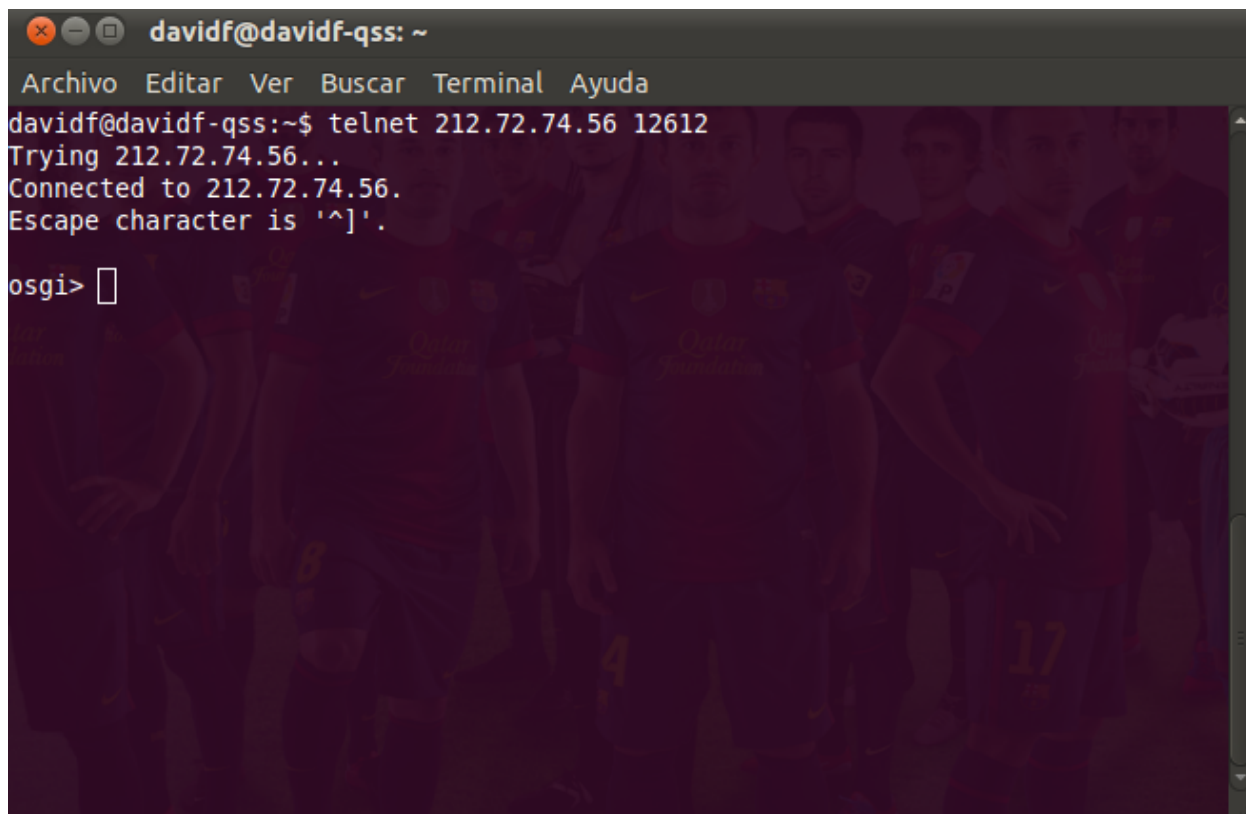


```
idempiere@ubuntu22base: ~/idempiere-fitness/fitness
Starting FitNesse on port: 8089
^Cidempiere@ubuntu22base:~/idempiere-fitness/fitness$ ./run.sh
java -jar fitness-standalone.jar
Mar 17, 2023 10:43:14 AM fitness.ConfigurationParameter loadProperties
INFO: No configuration file found (/home/idempiere/idempiere-fitness/fitness/p
ugins.properties)
Bootstrapping FitNesse, the fully integrated standalone wiki and acceptance test
ing framework.
root page: fitness.wiki.fs.FileSystemPage at ./FitNesseRoot#latest
logger: /home/idempiere/idempiere-fitness/fitness/log
authenticator: fitness.authentication.PromiscuousAuthenticator
page factory: fitness.html.template.PageFactory
page theme: bootstrap
Starting FitNesse on port: 8089
```

- In the iDempiere-Server you must also start the OSGI-bundles for
  - org.idempiere.fitness.fixture ... and
  - org.idempiere.fitness.server (details for a start via telnet you find below)

## Running Bundles From Demo Idempiere

- Go to idempiere demo and connect.



```
davidf@davidf-qss: ~  
Archivo  Editor  Ver  Buscar  Terminal  Ayuda  
davidf@davidf-qss:~$ telnet 212.72.74.56 12612  
Trying 212.72.74.56...  
Connected to 212.72.74.56.  
Escape character is '^['.  
osgi> 
```

- To Install bundles in idempiere demo in the console type this:

```
install http://jenkins.idempiere.com/job/iDempiereDaily  
/ws/buckminster.output/org.adempiere.server\_2.0.0-eclipse.feature/site.p2  
/plugins/org.idempiere.fitnessse.fixture\_2.0.0.v20140610-0232.jar
```

```
install http://jenkins.idempiere.com/job/iDempiereDaily  
/ws/buckminster.output/org.adempiere.server\_2.0.0-eclipse.feature/site.p2  
/plugins/org.idempiere.fitnessse.server\_2.0.0.v20140610-0232.jar
```

- You must start the services

```
osgi> ss
```

- Look for idempiere services, for example:

```
127    <<LAZY>>    org.idempiere.fitnessse.fixture_1.0.0.qualifier  
128    <<LAZY>>    org.idempiere.fitnessse.server_1.0.0.qualifier
```

- And start the services related:

```
osgi> start 127
```

```
osgi> start 128
```



# Allow SSL-Connection for self signed certificates

Outline of procedure - there are other options for ssl and cacerts handling experts.

In iDempiere-setup (1<sup>st</sup>/first field in Keystore window) computer must be named "localhost" (, otherwise name of computer same as alias)

- Commands to be executed in folder .../fitnesse-server/fitnesse (Command after bullet, comment next line in brackets):
  - `rm cacerts` (other cacert inherited from the repository in Github)
  - `cp /etc/ssl/certs/java/cacerts .` (Copy of certifikate from computer(iDempiere-web-server) to fitnesse (command incl. `\".\"`)
  - `keytool -list -v -keystore cacerts` (for inspection - Standard `\'no password\'` or `\'changeit\'` or `\'your password\'` !best )
  - `echo | openssl s_client -connect localhost:8443 2>&1 | sed -ne \'/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p\'> key.pem` (Please note: For this your iDempiere server (to be connected to fitnesse) must run)
  - `keytool -import -alias localhostnew -file key.pem -keystore cacerts` (if import is done several times localhost must change name, e.g. localhostN with N=1,2,... so simply numbering the term)

## Edit global setting

You must edit global setting

- In your browser type <http://localhost:8089/> (With the port you already configure in this case 8089)
- Go to root



**FrontPage** [add child]

**WELCOME TO IDEMPIERE [FitNesse!](#)**

*THE FULLY INTEGRATED STAND-ALONE ACCEPTANCE TESTING FRAMEWORK AND WIKI.*

Suites:  
[IdempiereSuite](#)  
[AvgCostSuite](#)

Zk Examples:  
[ZkSingleClientLogin](#)  
[ZkMultiClientLogin](#)

To Learn More...	
<a href="#">A One-Minute Description</a>	<i>What is <a href="#">FitNesse</a>? Start here.</i>
<a href="#">A Two-Minute Example</a>	<i>A brief example. Read this one next.</i>
<a href="#">User Guide</a>	<i>Answer the rest of your questions here.</i>
<a href="#">Acceptance Tests</a>	<i><a href="#">FitNesse's</a> suite of Acceptance Tests</i>

Release v20111026

[Front Page](#) | [User Guide](#)  
[root](#) (for global !path's, etc.)

- Write your own workspace location here
- Change for the same Port as you run idempiere in this case 8089



[add child]

**Edit** Define the global path:

Where to find the fixtures classes: ( i.e. /home/hengsin/workspace/idempiere/fitnesse/bin )

**Refactor** `variable defined: fitnesse_home=/home/davidf/sources/myexperiment/fitnesse`

**Where Used** `classpath: /home/davidf/sources/myexperiment/fitnesse/fitnesse.jar:/home/davidf/sources/myexperiment/fitnesse/lib/*:jar:/home/davidf/sources/myexperiment/fitnesse/bin`

**Search** Define the command to access iDempiere fitnesse servlet

**Files** There are some important variables here:

- \* LOG4J\_LEVEL to set the desired log level
- \* Include "-Xdebug -Xrunjdpw:transport=dt\_socket,server=y,suspend=y,address=1044" to enable remote debugging of the test
- \* URL to access the iDempiere fitnesse servlet

**Versions** This variables can be redefined specifically at page level.

**Recent Changes** `variable defined: TEST_RUNNER=fitnesse.client.FitServerServletInvoker`

**User Guide** `variable defined: COMMAND_PATTERN=java -Xms32m -Xmx512m -DLOG4J_LEVEL=CONFIG -cp %p %m http://localhost:8089/fitnesse/FitServlet`

**Test History** To enable remote debugging the tests will stop until you connect remotely via eclipse using RemoteADempiereFitnesse.launch

You must add this to the URL ?responder=test&remote\_debug=true

`variable defined: REMOTE_DEBUG_COMMAND=java -Xms32m -Xmx512m -DLOG4J_LEVEL=ALL -Xdebug -Xrunjdpw:transport=dt_socket,server=y,suspend=y,address=1044 -cp %p %m http://localhost:8089/fitnesse/FitServlet`

[Front Page](#) | [User Guide](#)  
[root](#) (for global !path's, etc.)

## More screenshots

- The following are the screenshots of the tests layout that you can find when you click on the items above.

# IdempiereSuite [add child]

## Contents:

- [Quick Test +](#)
- [Test Bug 1733602 +](#)
- [Test Cash Pos Order +](#)
- [Test Complete Payment +](#)
- [Test Create Business Partner +](#)
- [Test Create Reference +](#)
- [Test Initial Client Setup +](#)
- [Test Login Garden Admin +](#)
- [Test Login System +](#)
- [Test Post Internal Inventory +](#)

## [AvgCostSuite.](#)

# BasicTest [add child]

This test is intended to create repeatable test cases for Average Invoice Costing, order to make a refactor we need to:

- \* stabilize current code
- \* guarantee that refactoring don't break actual results

Brief description of the tests:

- \* login into GardenAdmin
- \* ensure that tenant is configured with avg inv costing
- \* create a product (random name/value)
- \* create a PO to purchase 50 products with value 10
- \* create a material receipt
- \* create a matchPO
- \* post the matchPO
- \* create an invoice
- \* post the invoice
- \* verify that avg cost=10 and qty=50
- \* create a POS sales order for 3 products
- \* post the shipment
- \* verify that shipment was posted with cost=10
- \* verify that avg cost=10 and qty=47
- \* create a PO to purchase 10 products with value 11
- \* create a material receipt based on the PO
- \* check the matchPO created
- \* post the matchPO
- \* verify that avg cost=10.175438596 and qty=57
- \* create an internal use inventory for 2 products
- \* post internal use
- \* verify that internal use used avg cost=10.175438596
- \* verify that avg cost=10.175438596 and qty=55

► Included page: [CommonTests.LoginGardenAdmin](#) [Expand All](#) | [Collapse](#)

# ZkSuite [add child]

## Contents:

- [MissingFieldTranslationTab](#)
- [ProductionTest](#)
- [ProductionTestEmptyTable](#)
- [SearchOnEmptyTable](#)
- [ZkGardenAdminLogin](#)
- [ZkSystemAdminLogin](#)

[Front Page](#) | [User Guide](#)  
[root](#) (for global !path's, etc.)

## See Also

Still open how to handle thes links inside the wiki or to outside in docusaurus:

- `[NF001_Fitnesse_Integration](NF001_Fitnesse_Integration "wikilink")`
- `[Free PDF book describing in more detail](http://red1.org/iDempiereFitNesse.pdf)`

`[en:Fitnesse.HowTo](en:Fitnesse.HowTo "wikilink")`

`[es:Fitnesse.HowTo](es:Fitnesse.HowTo "wikilink")`

`[U](Category:Fitnesse "wikilink") [Category:Functional`

`Testing](Category:Functional_Testing "wikilink")`

`[Category:Development](Category:Development "wikilink")`


`[Category:Developer`

`documentation](Category:Developer_documentation "wikilink")`

`[Category:HowTo-Technical](Category:HowTo-Technical "wikilink")`

*End of Document*

**Tags:** [docusaurus](#) [test](#) [idempiere](#)

 [Edit this page](#)