

Zabbix - Gerenciando relatórios personalizados com JasperReports

Através da [API do Zabbix](#)

Por Thiago Murilo Diniz ([LinkedIn](#))
fev/2018

Índice

- [Introdução](#) (slide 3)
- [Criando o projeto no Jaspersoft Studio](#) (slide 6)
- [Criando o relatório no Jaspersoft Studio](#) (slide 10)
- [Scriptlets](#) (slide 27)
- [Publicando o relatório no JasperReports Server](#) (slide 44)
- [Subreports](#) (slide 63)
- [Conclusão](#) (slide 85)

Introdução

O presente documento tem como objetivo descrever uma maneira de criar e gerenciar relatórios personalizados para o Zabbix, acessando as informações através da API JSON-RPC disponibilizada pelo mesmo.

O Jaspersoft Studio é utilizado para criação dos relatórios. Uma vez prontos, os relatórios podem ser publicados e gerenciados no JasperReports Server.

Versões utilizadas na criação deste documento

Zabbix 3.4

Download: <https://www.zabbix.com/download>

JasperReports Server 6.4.2

Download: <https://community.jaspersoft.com/project/jasperreports-server/releases>

Manual de instalação disponível na área de downloads.

Jaspersoft Studio 6.4.3

Download: <https://community.jaspersoft.com/project/jaspersoft-studio/releases>

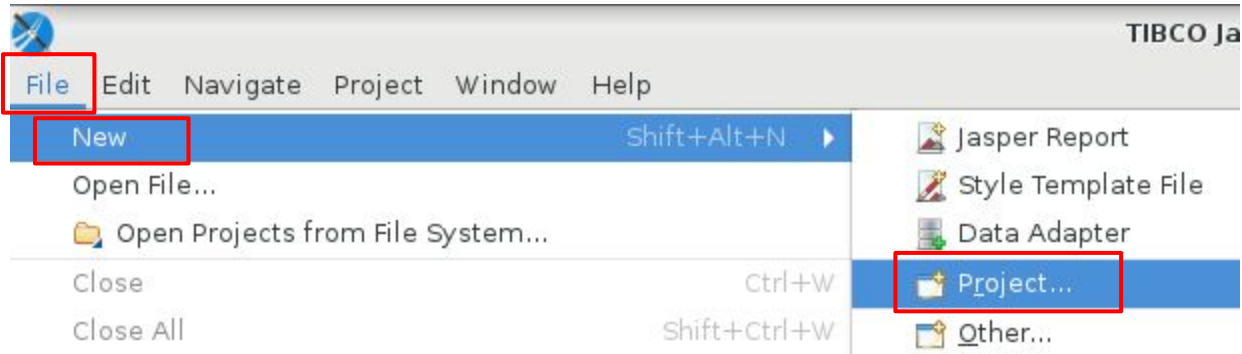
Importante!

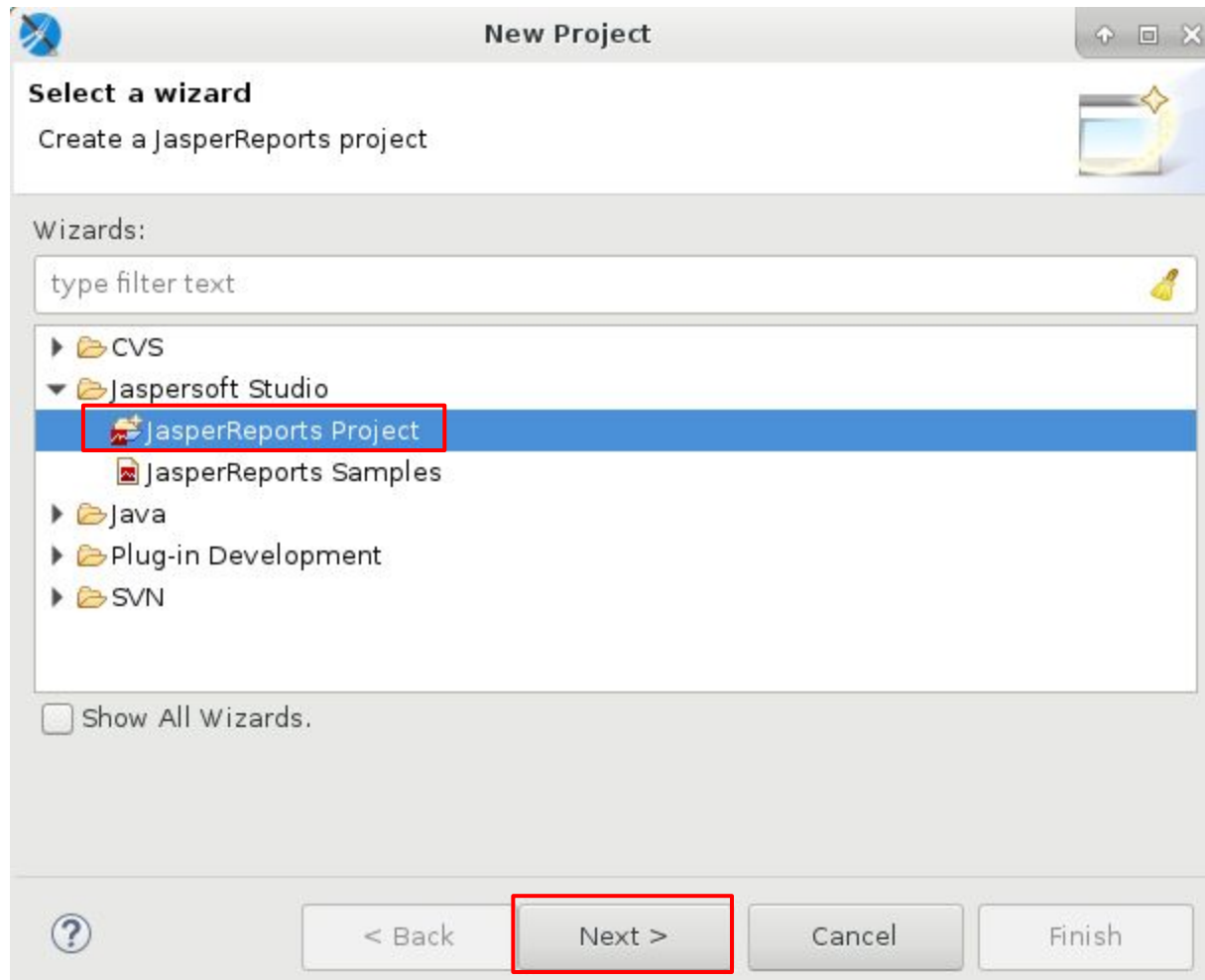
Para que o Jaspersoft Studio e o JasperReports Server possam consumir os dados do Zabbix através da API JSON-RPC é necessário habilitar o “[Web Service Data Source](#)”:


Download: <https://community.jaspersoft.com/project/web-service-data-source/releases>

Instalação: <https://community.jaspersoft.com/wiki/installing-webservice-datasource>

Criando o projeto no Jaspersoft Studio






 **New Jasper Project Wizard** ↑ □ ×

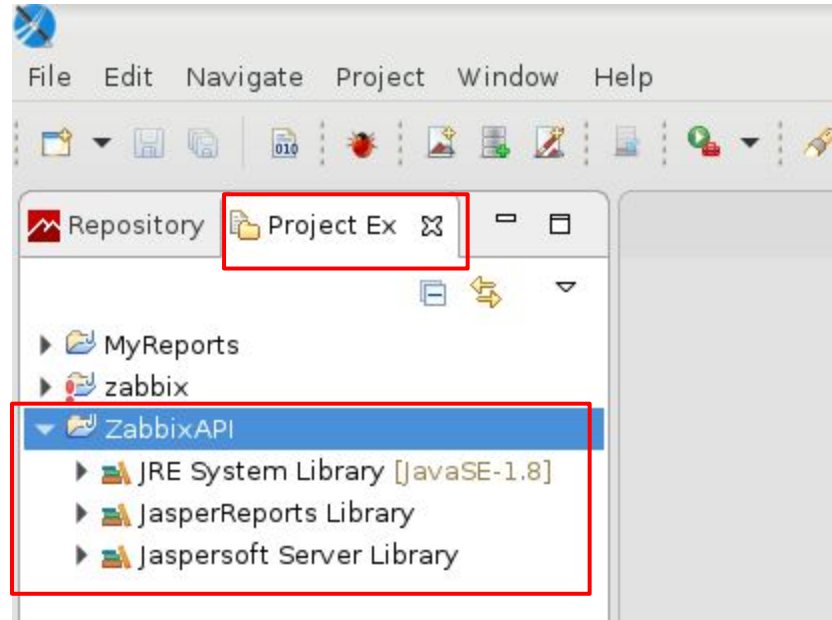
New JasperReports Project

Create a new JasperReports project

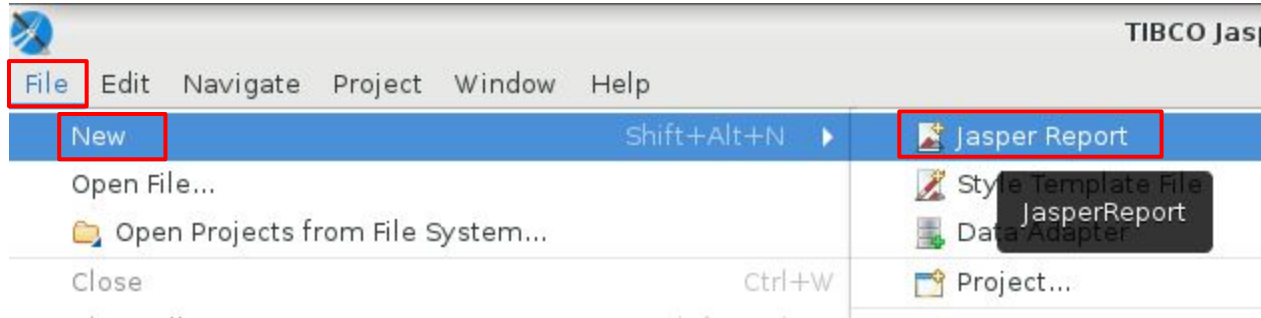
Name

 < Back Next > Cancel Finish

Projeto criado!



Criando o relatório no Jaspersoft Studio





Report Templates

Please select a report from a template

Zoom:

Categories

All

Print Ready Reports

Web Interactive Rep

Format A4

Format Landscape

Report Books



Blank A4



Blank A4 Landscape



Blank Letter



Blank Let...Landscape



Cherry



Cherry Landscape





< Back

Next >

Cancel




Finish




New Report Wizard

Report file


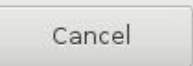

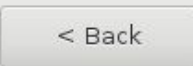

Please select your reports file name with .jrxml extension.

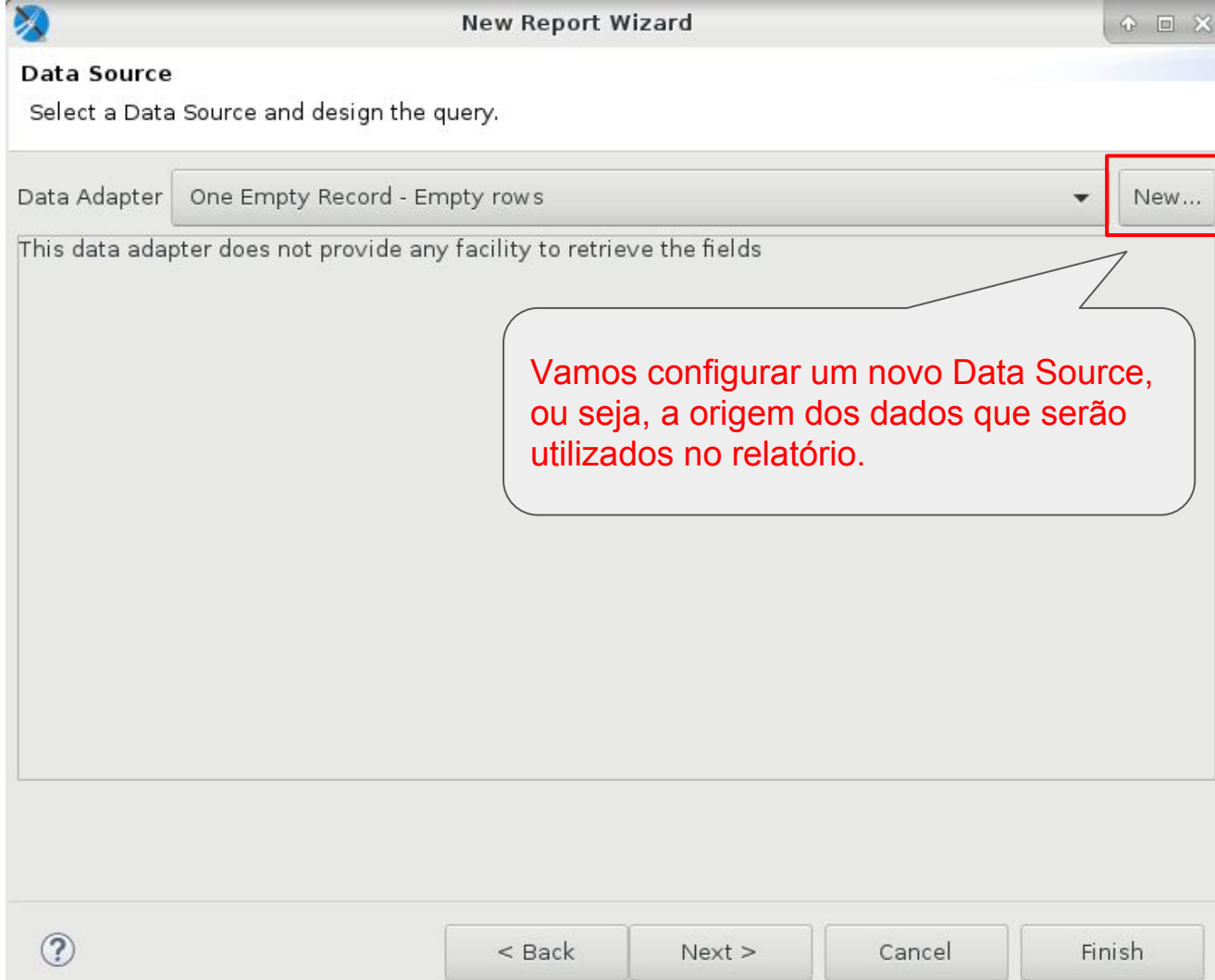
Enter or select the parent folder:

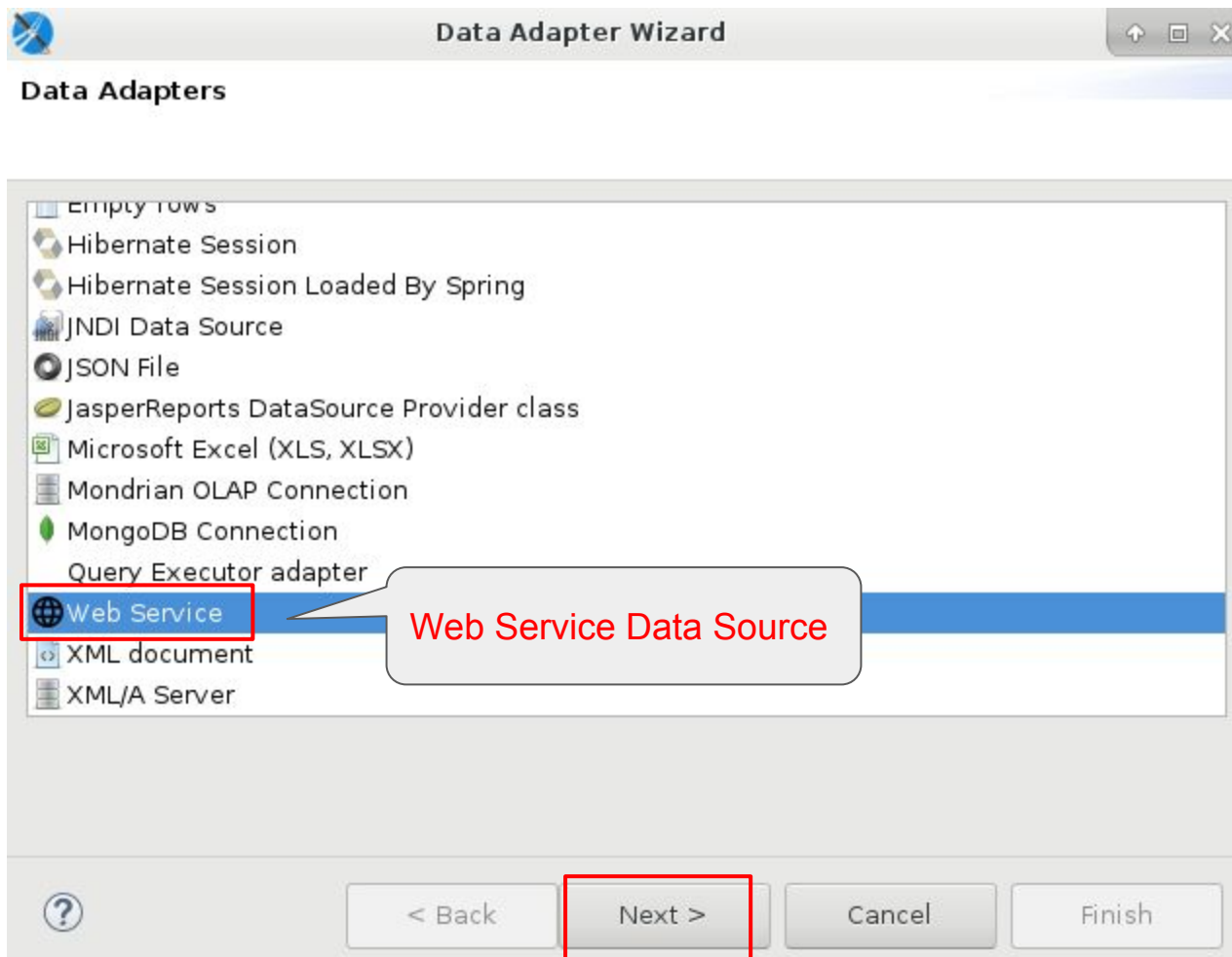



- ▶  MyReports
- ▶  ZabbixAPI
- ▶  zabbix

File name:







 **Data Adapter Wizard** ↑ □ ×

Data Adapter
Web Service


Name:

Server URI Endereço do Zabbix

Connection Type

Language

Authentication ...



New Report Wizard

Data Source

Select a Data Source and design the query.

Data Adapter: **ZabbixAPI - Web Service** New...

Base URL: **api_jsonrpc.php**

Content Type: **JSON**

Request Fields Source

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": [
      "hostid",
      "host"
    ],
    "selectInterfaces": [
      "interfaceid",
      "ip"
    ]
  },
  "id": 2,
  "auth": "355867044fd5af6f47d9d93189925a0f"
}
```

Requisição JSON que será feita à API Zabbix.

Uso do [método host.get](#) da API Zabbix.

Como solicitar à API Zabbix um token de autenticação via CURL:

```
curl -i -X POST -H 'Content-type:application/json' -d
'{"jsonrpc":"2.0","method":"user.login","params":{"user":"Admin","password":"zabbix"},"auth":null,"id":0}'
http://192.168.100.1/api_jsonrpc.php
```

Uso do [método user.login](#).

< Back Next > Cancel Finish

New Report Wizard

Data Source

Select a Data Source and design the query.

Data Adapter: ZabbixAPI - Web Service

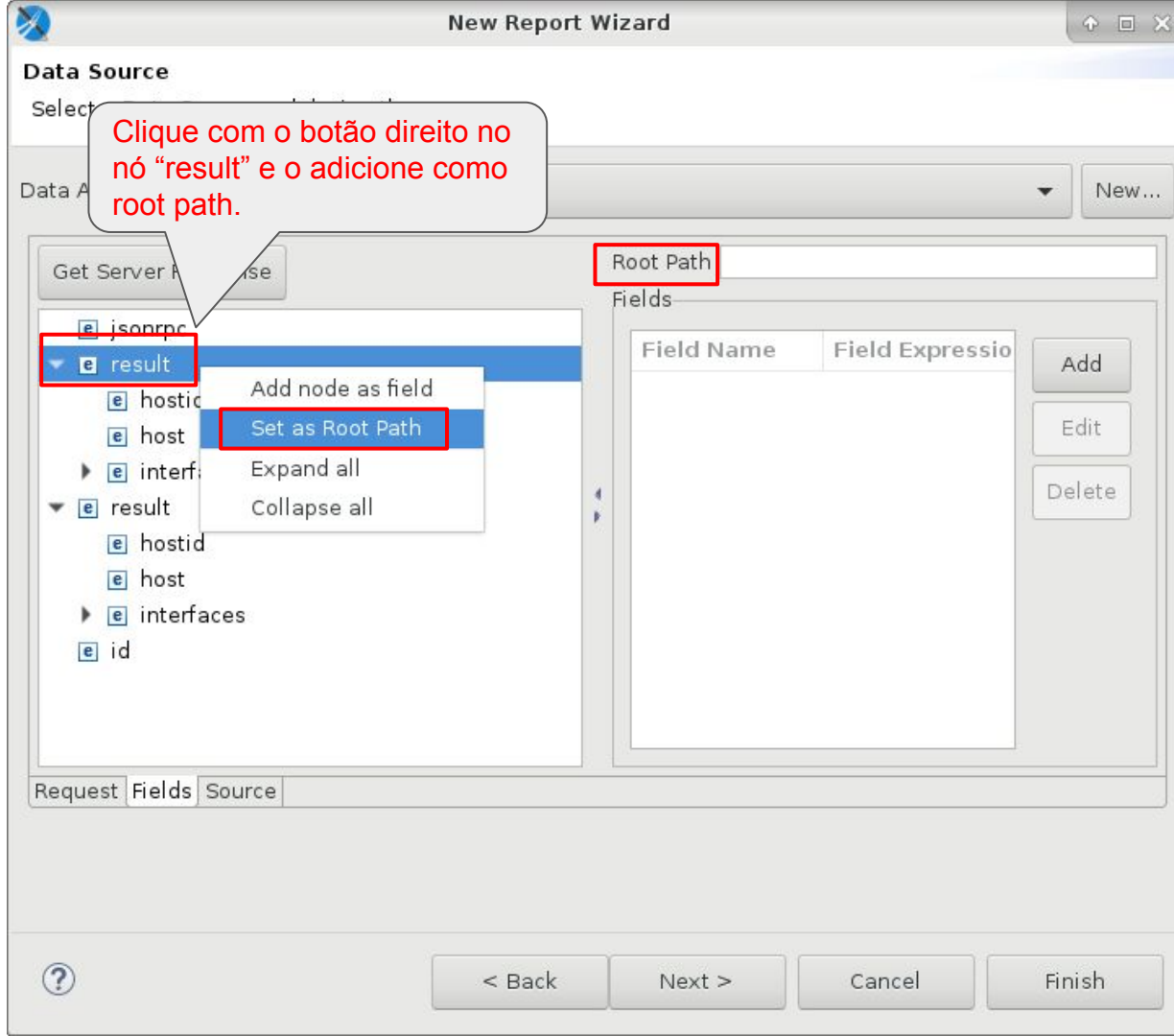
Get Server Response

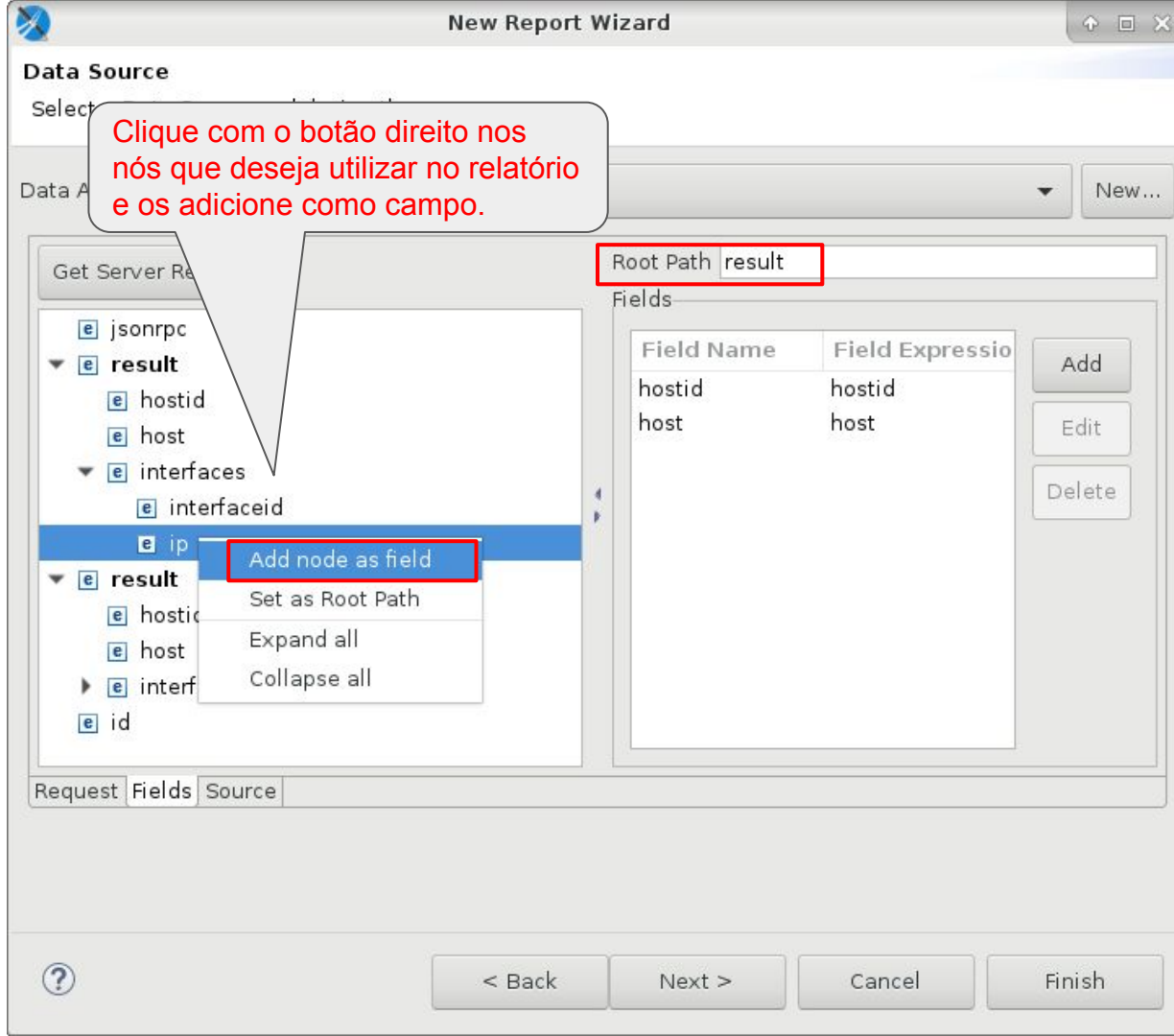
jsonrpc
result
 hostid
 host
 interfaces
result
 hostid
 host
 interfaces
id

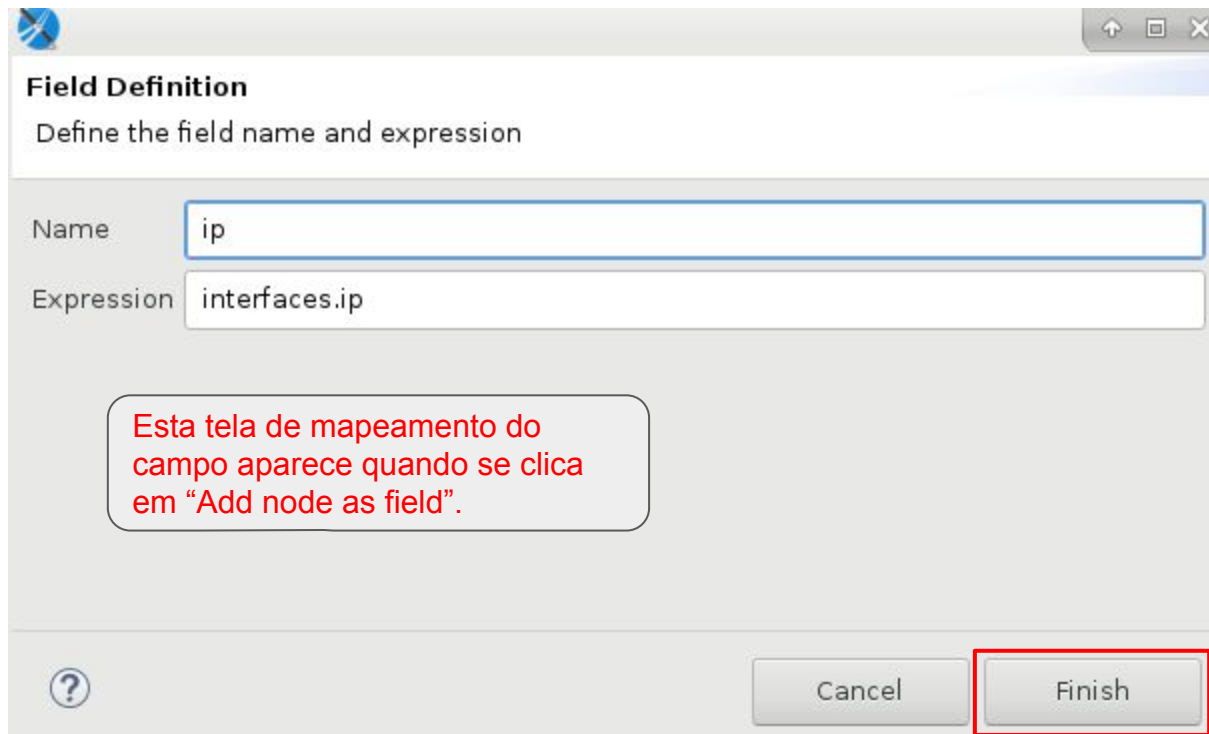
Request Fields Source

Clique neste botão para que o Jaspersoft Studio receba o retorno da API do Zabbix, permitindo assim o mapeamento dos campos disponíveis para uso no relatório.

< Back Next > Cancel Finish








The image shows a 'Field Definition' dialog box with a title bar containing a question mark icon and standard window controls. The main area has a subtitle 'Define the field name and expression'. Below this are two text input fields: 'Name' with the value 'ip' and 'Expression' with the value 'interfaces.ip'. A red-bordered callout box contains text in Portuguese. At the bottom, there is a help icon (question mark), a 'Cancel' button, and a 'Finish' button which is highlighted with a red rectangle.

Field Definition
Define the field name and expression

Name

Expression

Esta tela de mapeamento do campo aparece quando se clica em "Add node as field".



New Report Wizard

↑ □ ×

Data Source

Select a Data Source and design the query.

Data AdapterZabbixAPI - Web ServiceNew...

Get Server Response

jsonrpc

result

hostid

host

interfaces

interfaceid

ip

result

hostid

host

interfaces

id

Root Pathresult

Fields

Field Name	Field Expression
hostid	hostid
host	host
ip	interfaces.ip

AddEditDelete

Request

Fields

Source

?

< Back

Next >

Cancel

Finish

New Report Wizard

Fields

Please select dataset fields

Dataset Fields

- # ip

>

>>

<

<<

Fields

- # hostid
- # host

Up

Down

Arraste para a caixa da direita os campos que deverão estar disponíveis para uso no relatório.

?

< Back

Next >

Cancel

Finish

File Edit View Navigate Project Window Help

The screenshot displays the TIBCO JasperSoft Studio interface. The top menu bar includes File, Edit, View, Navigate, Project, Window, and Help. Below the menu is a toolbar with various icons for file operations and report design. The left sidebar contains several panels: Repository, Project Explorer, Outline, and Report State. The Repository panel shows a tree structure with 'MyReports', 'zabbix', and 'ZabbixAPI' folders. Under 'ZabbixAPI', there are 'JRE System Library [JavaSE-1.8]', 'JasperReports Library', 'Jaspersoft Server Library', and 'main.jrxml'. The 'main.jrxml' file is highlighted with a red box. The Project Explorer panel shows the 'main' project with sub-items: Styles, Parameters, Fields, Sort Fields, Variables, and Scriptlets. The 'Fields' item is highlighted with a red box, and a list of fields (hostid, host, ip) is shown below it. The Outline panel shows the report structure with sections: Title, Page Header, Column Header, and Detail 1. The Report State panel shows tabs for Console, Errors, and Statistics. The main design area shows a grid with these sections. A speech bubble points to the 'Fields' list in the Outline panel, containing the text: 'Relatório criado e com os campos disponíveis para uso!'.

Repository Project Ex

main.jrxml

Main Report

100%

0 1 2 3 4 5 6 7 8

0 1

Title

Page Header

Column Header

Detail 1

Design Source Preview

Report State

Console Errors Statistics

Outline

main

Styles

Parameters

Fields

hostid

host

ip

Sort Fields

Variables

Scriptlets

Relatório criado e com os campos disponíveis para uso!

Adição do título no relatório

The screenshot displays the Jaspersoft Studio environment with a report titled "Main Report" open. The design view shows a grid with sections labeled "Page Header", "Detail 1", and "Page Footer". The text "Zabbix API" is placed in the "Page Header" section. A red arrow points from the "Static Text" element in the "Basic Elements" palette to the "Zabbix API" text on the grid. A speech bubble contains the instruction: "Arraste um elemento 'Static Text' para a seção 'Title' do relatório." The "Properties" panel on the right shows the "Text" property set to "Zabbix API".

Arraste um elemento "Static Text" para a seção "Title" do relatório.

StaticT...bix API

Text Zabbix API

Text Alignment

Rotation

File Edit View Navigate Project Window Help



Repository

Project Ex



MyReports

zabbix

ZabbixAPI

JRE System Library [JavaSE-1.8]

JasperReports Library

Jaspersoft Server Library

main.jrxml

Outline



main

Styles

Parameters

Fields

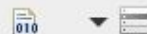
hostid

host

ip

*main.jrxml

Main Report



100%

Se

0 1 2 3 4 5 6 7 8

0 1 2 3 4 5 6 7 8

Zabbix API

Page Header

host

ip

\$F{host}

\$F{ip}

Column Header

Detail 1

Column Footer

Design Source Preview

Report State

Console Errors Statistics

Arraste os campos para a seção "Detail".
Automaticamente será criado o cabeçalho da
coluna para o respectivo campo.
A seção "Detail" faz o loop nos resultados
retornados pela API do Zabbix, exibindo uma
linha por Host encontrado.



Repository Project Ex

- MyReports
- zabbix
- ZabbixAPI**
 - JRE System Library [javaSE-1.8]
 - JasperReports Library
 - Jaspersoft Server Library
 - main.jrxml

Outline

main.jrxml

ZabbixAPI Java Page 1 of 1

Zabbix API	
host	ip
Zabbix server	["127.0.0.1"]
192.168.25.1	["192.168.25.1"]

Design Source **Preview**

Clicando em "Preview" o relatório será executado e exibido. No caso foram listados os dois hosts existentes na minha instalação do Zabbix.

Report State

Console Errors (0) Statistics

Compilation Time	0,167 sec
Filling Time	0,418 sec
Report Execution Time	0,729 sec
Export Time	0 sec
Total Pages	1 pages
Processed Records Count	-records

Melhorando as coisas...

Durante a configuração do Data Source ([slide 16](#)), para fazer a requisição JSON à API do Zabbix, o token de autenticação foi adquirido utilizando o comando CURL. Mas seria muito chato ter que autenticar manualmente na API do Zabbix toda vez que for executar o relatório.

Para resolver esta questão e fazer com que o relatório faça automaticamente o login e o logout na API do Zabbix foi utilizado o recurso “Scriptlets” do Jaspersoft Studio. Os Scriptlets são basicamente uma interface entre o relatório e um código Java, provendo muita flexibilidade por permitir o uso de funções criadas pelo usuário.

Sobre Scriptlets:

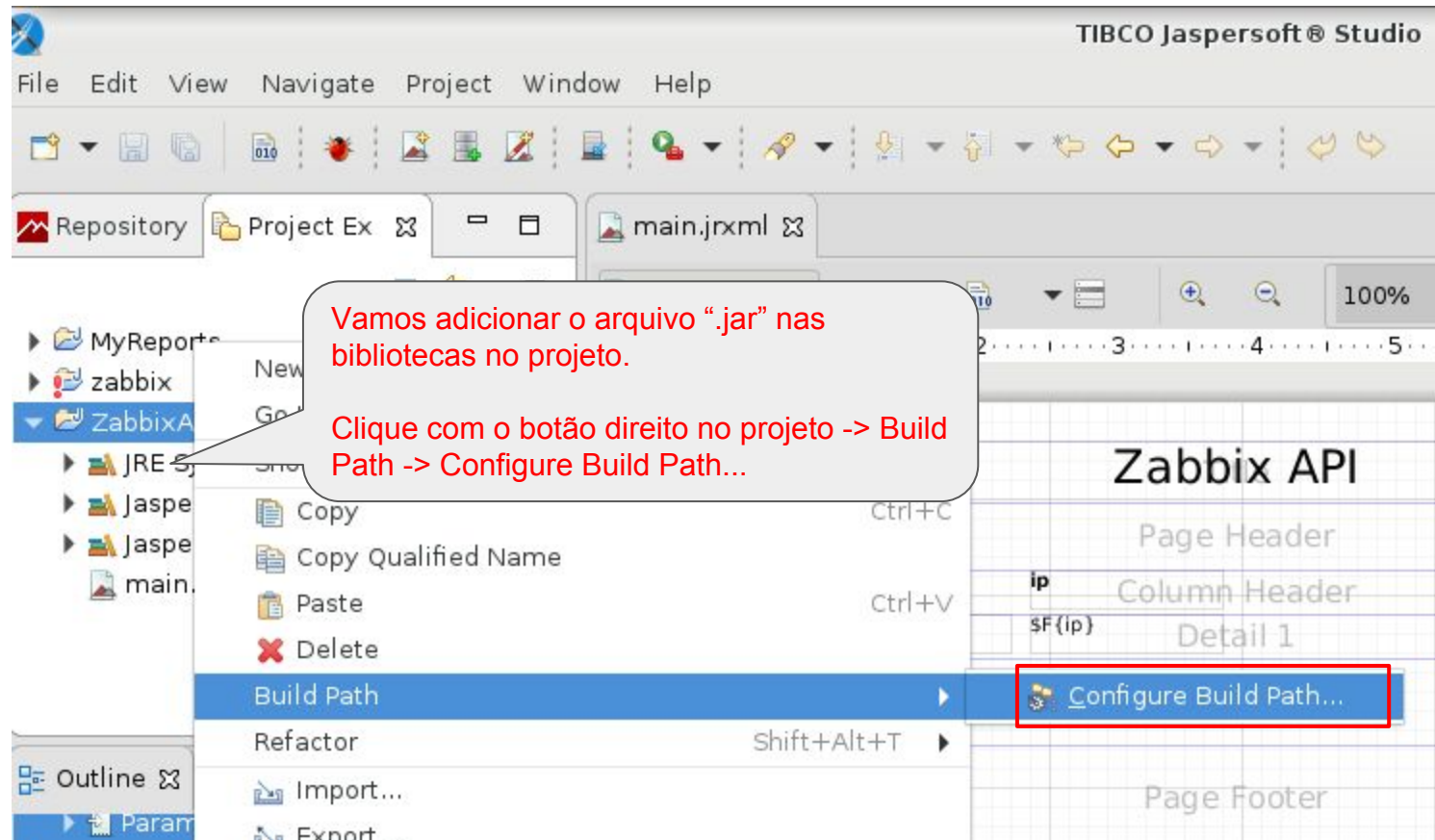
<https://community.jaspersoft.com/blog/all-you-want-know-about-scriptlets>

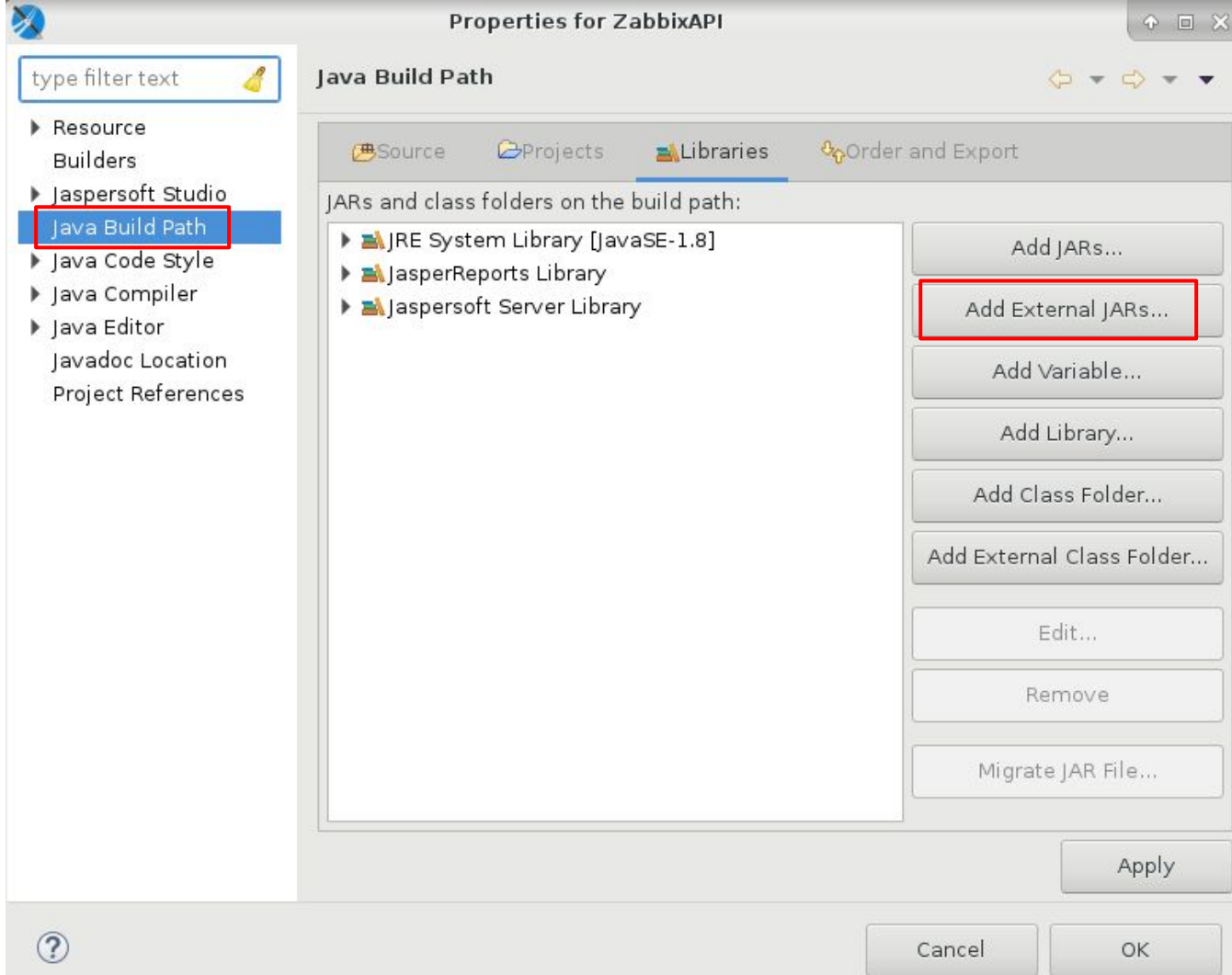
<https://community.jaspersoft.com/blog/scriptlet-example-jaspersoft-6x-begineers-hello-world-sample-example>

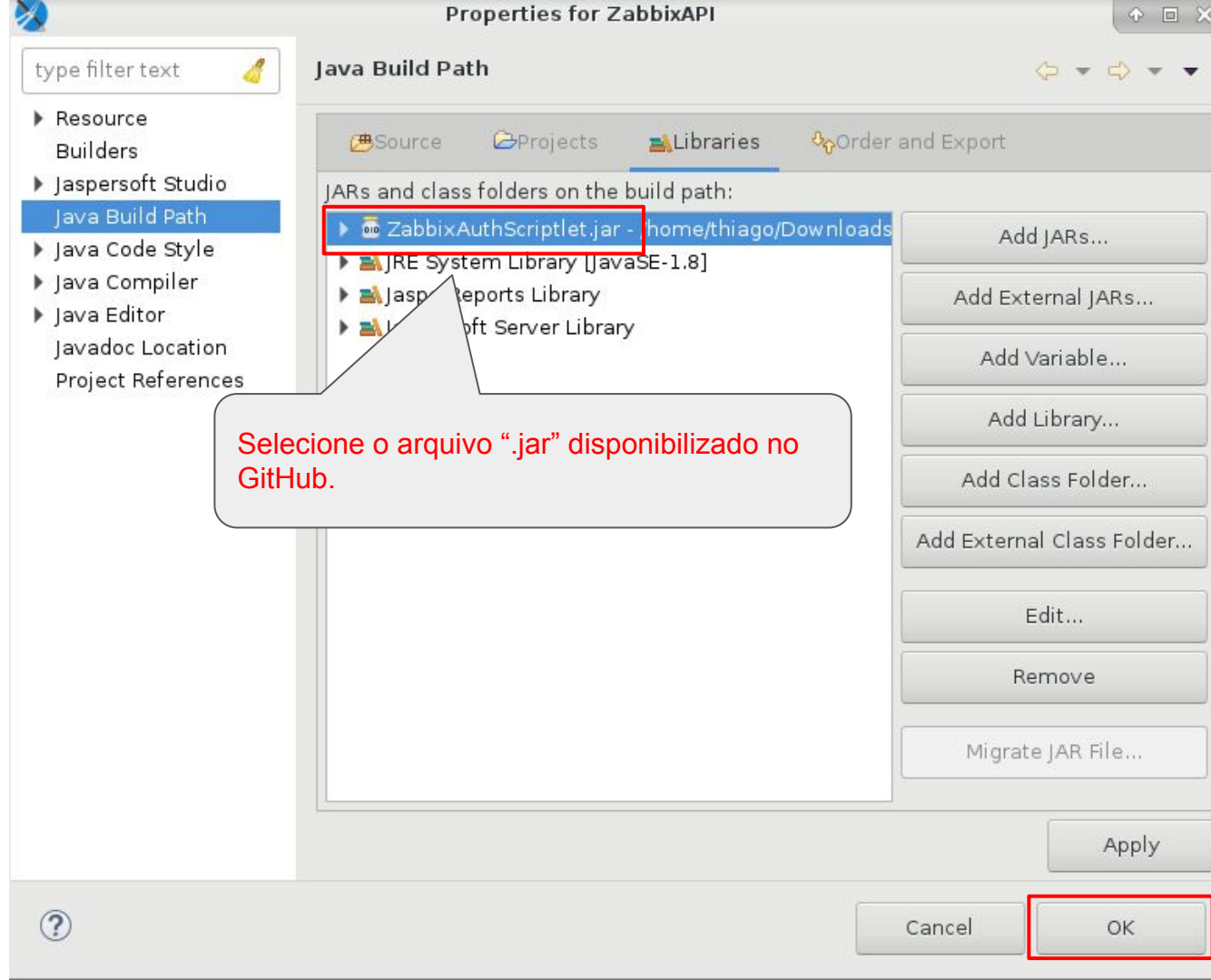
Scriptlets

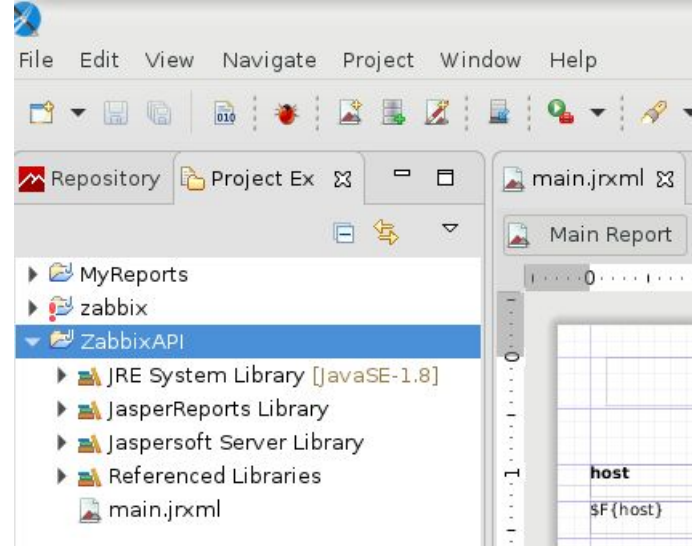
Criei um código Java para ser utilizado como Scriptlet e que permite fazer login e logout na API do Zabbix. Tanto o arquivo “.jar” quanto o código fonte estão no GitHub:

<https://github.com/thiagomdiniz/zabbix>

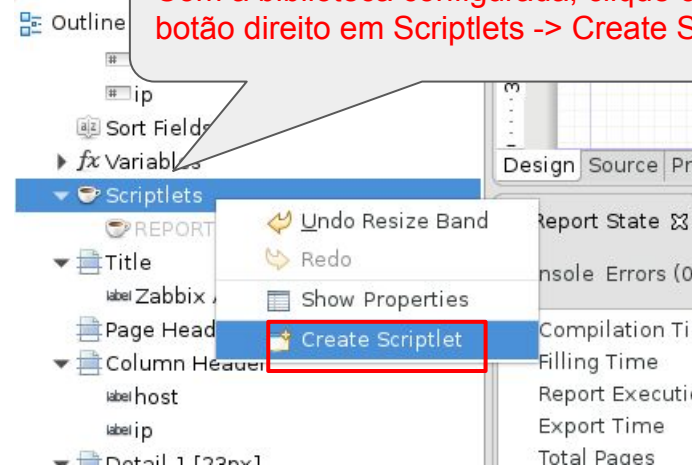


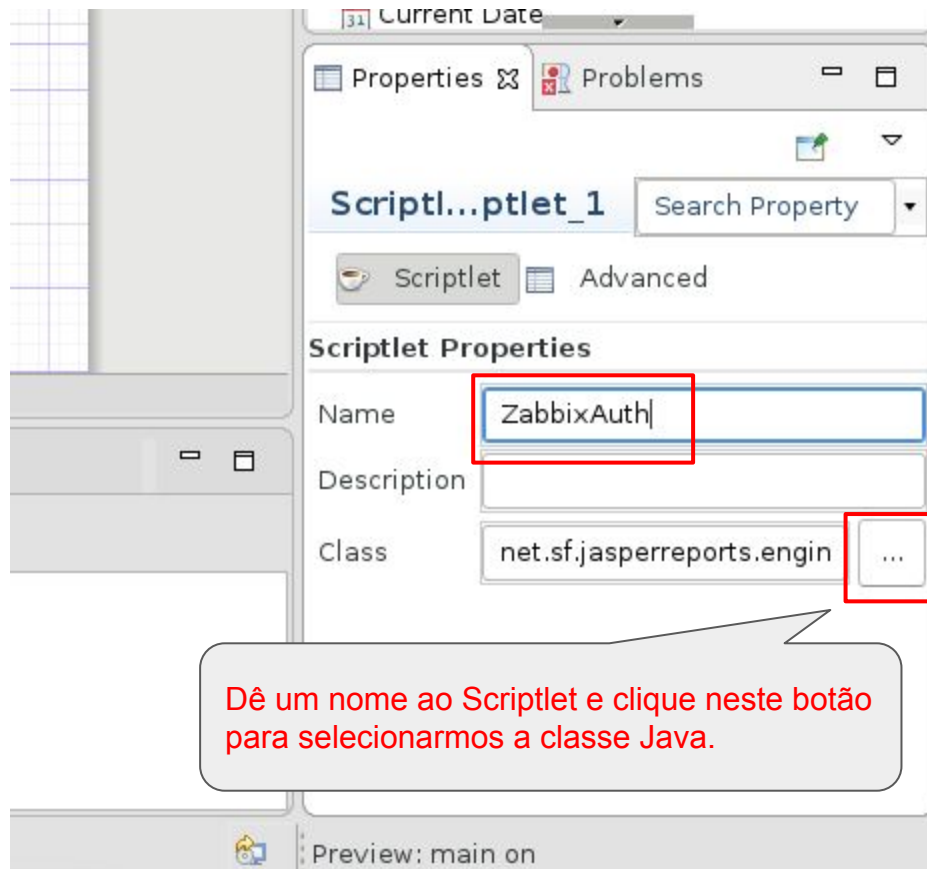


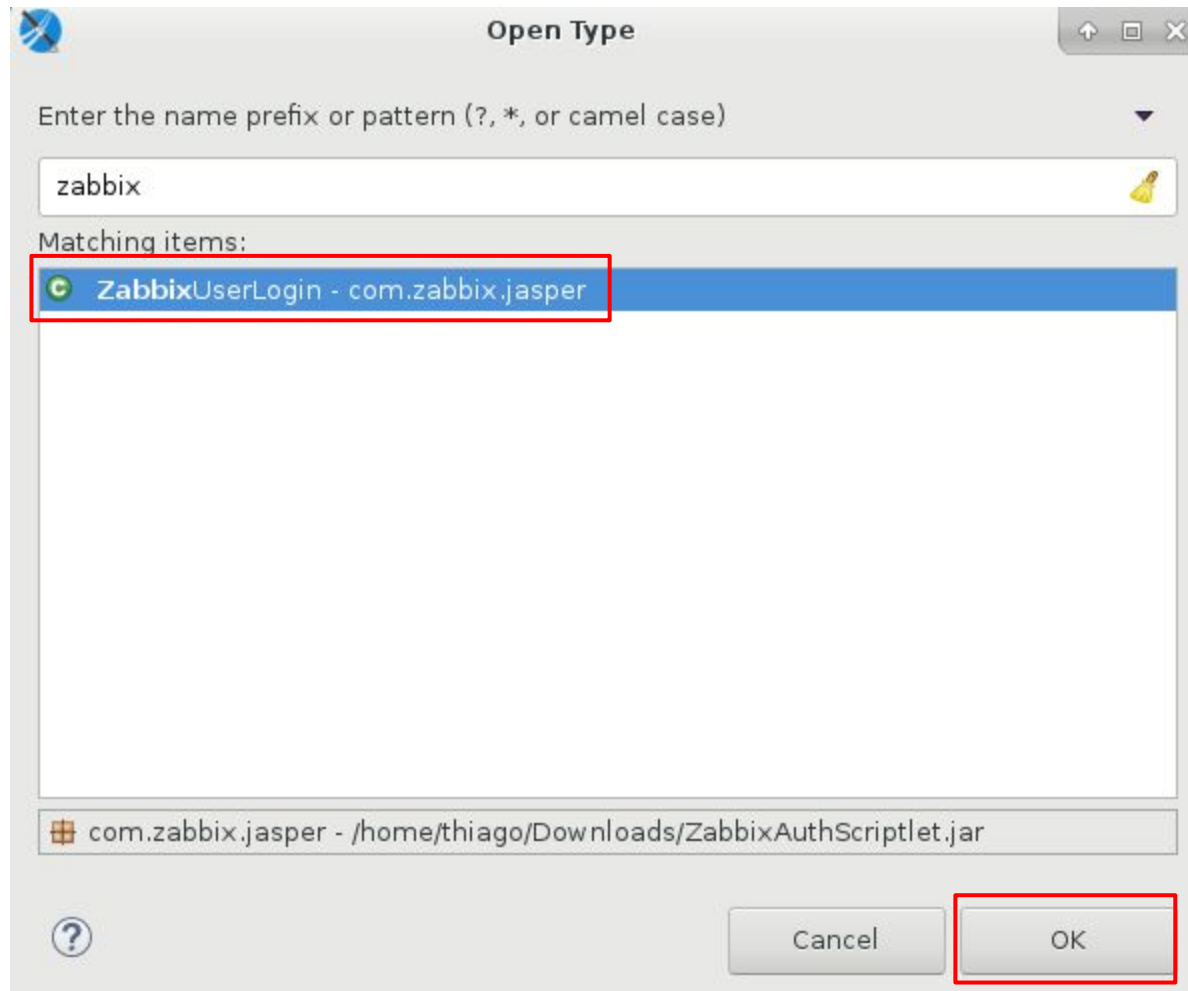




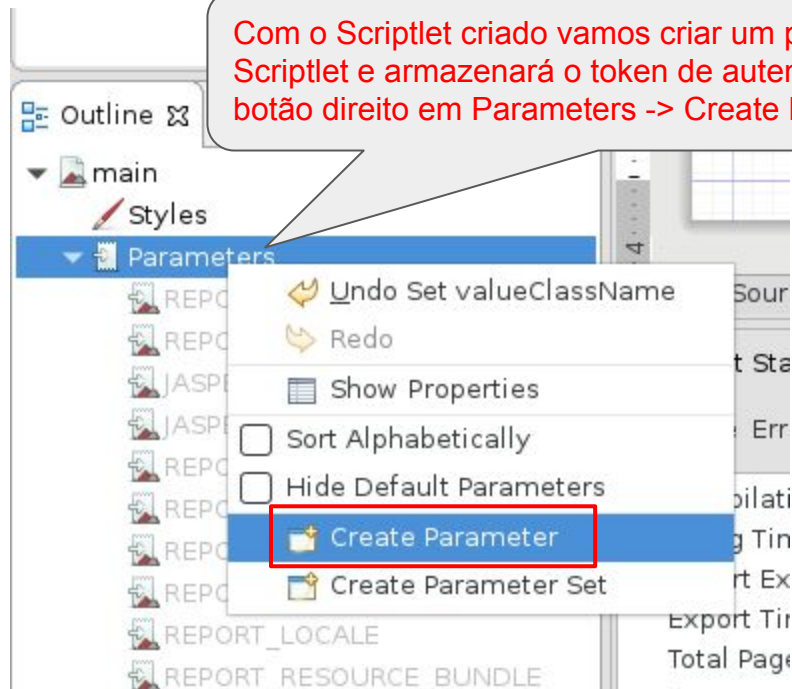
Com a biblioteca configurada, clique com o botão direito em Scriptlets -> Create Scriptlet.

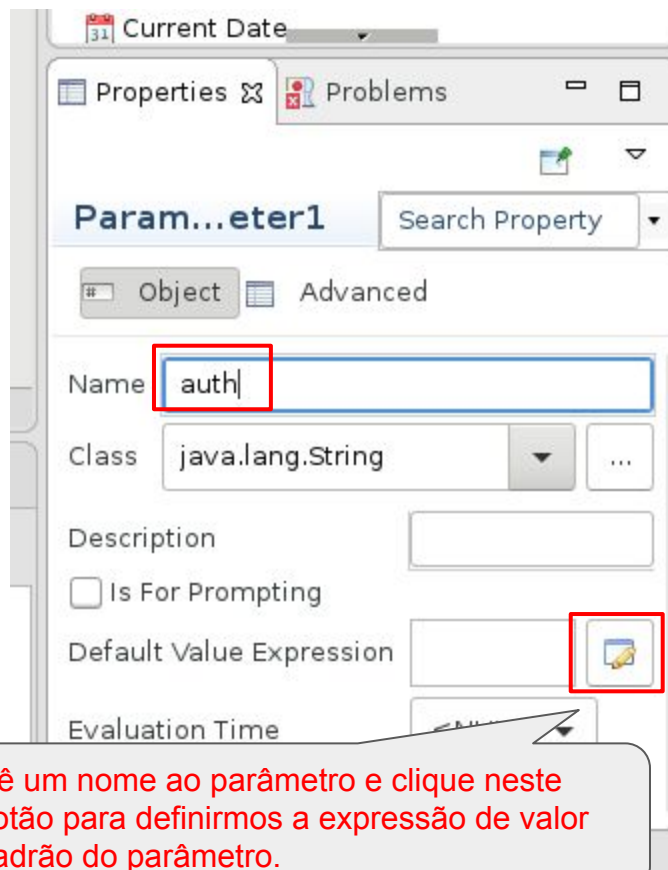


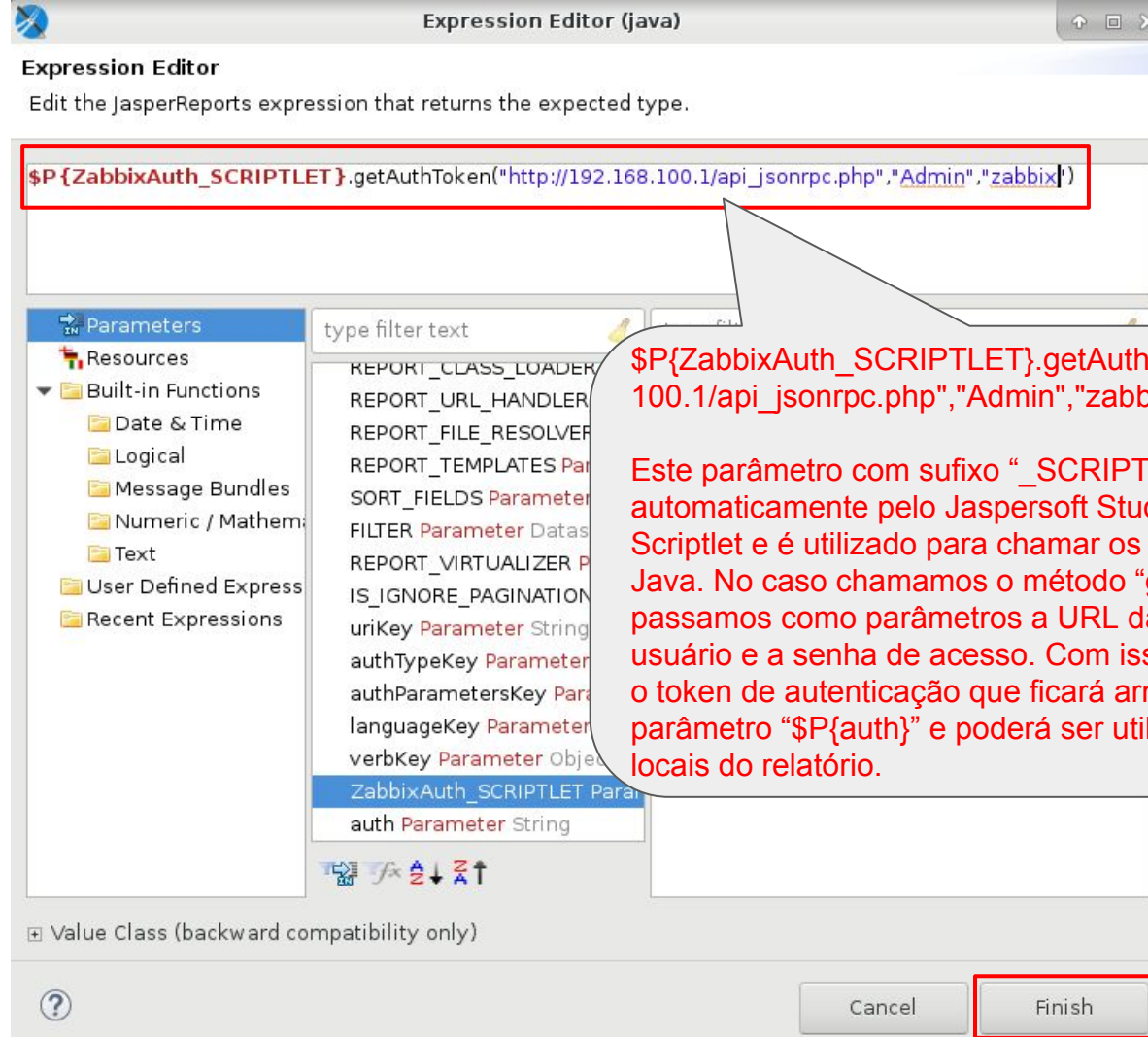


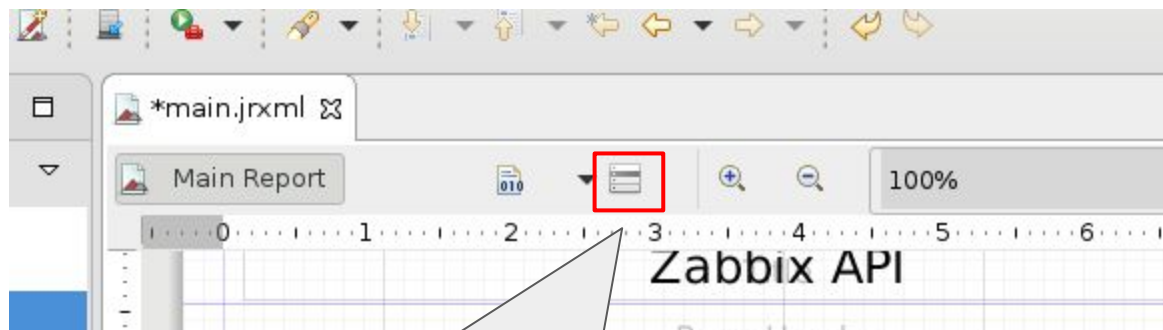


Com o Scriptlet criado vamos criar um parâmetro que usará o Scriptlet e armazenará o token de autenticação. Clique com o botão direito em Parameters -> Create Parameter.









Clique neste botão para editar o DataSet do relatório.

Dataset and Query Dialog

ZabbixAPI

Query **Java Bean**

Language **WebServiceQuery** Read Fields

Base URL **api_jsonrpc.php**

Content Type **JSON**

```
"params": {  
  "output": [  
    "hostid",  
    "host"  
  ],  
  "selectInterfaces": [  
    "interfaceid",  
    "ip"  
  ]  
},  
"id": 2  
"auth": "$P{auth}"  
}
```

Header Name Header Value Add Edit Delete

Request **Fields** **Force**

Field Name	Class Type	Description	Properties
hostid	java.lang.Integer		

Add Edit Delete Up Down

? Cancel OK

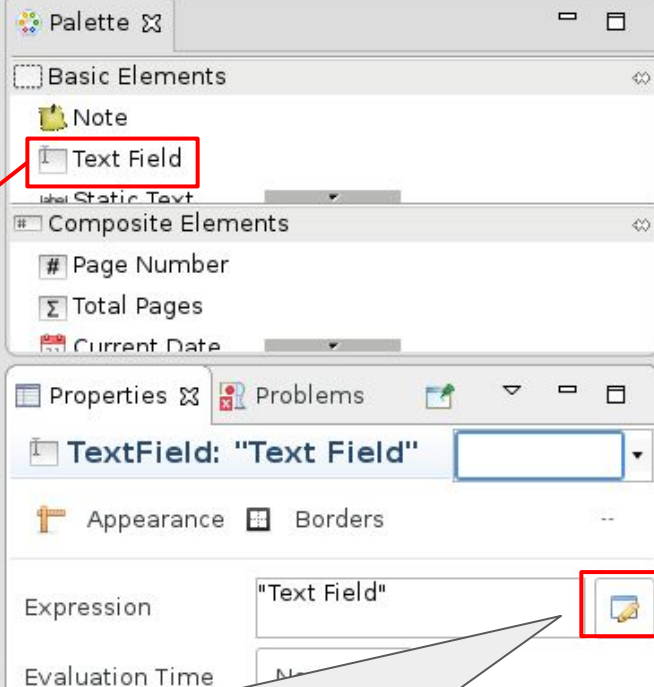
Agora, no lugar do token que havíamos colocado na requisição JSON, vamos colocar o parâmetro “\$P{auth}”. No momento em que o relatório for executado este parâmetro será substituído pelo token de autenticação pego através da classe Java (Scriptlet).

Mas...

Conforme recomendado na documentação da API Zabbix, também é necessário realizar o logout para prevenir a geração de um grande número de sessões de usuário abertas.

Mediante esta informação vamos configurar também o logout.

A seção "Summary" é a última a ser executada no relatório. Por isso colocaremos a ação de logout nela. Arraste um campo de texto para a seção "Summary".



Palette

- Basic Elements
 - Note
 - Text Field**
 - Static Text
- Composite Elements
 - # Page Number
 - Σ Total Pages
 - Current Date

Properties

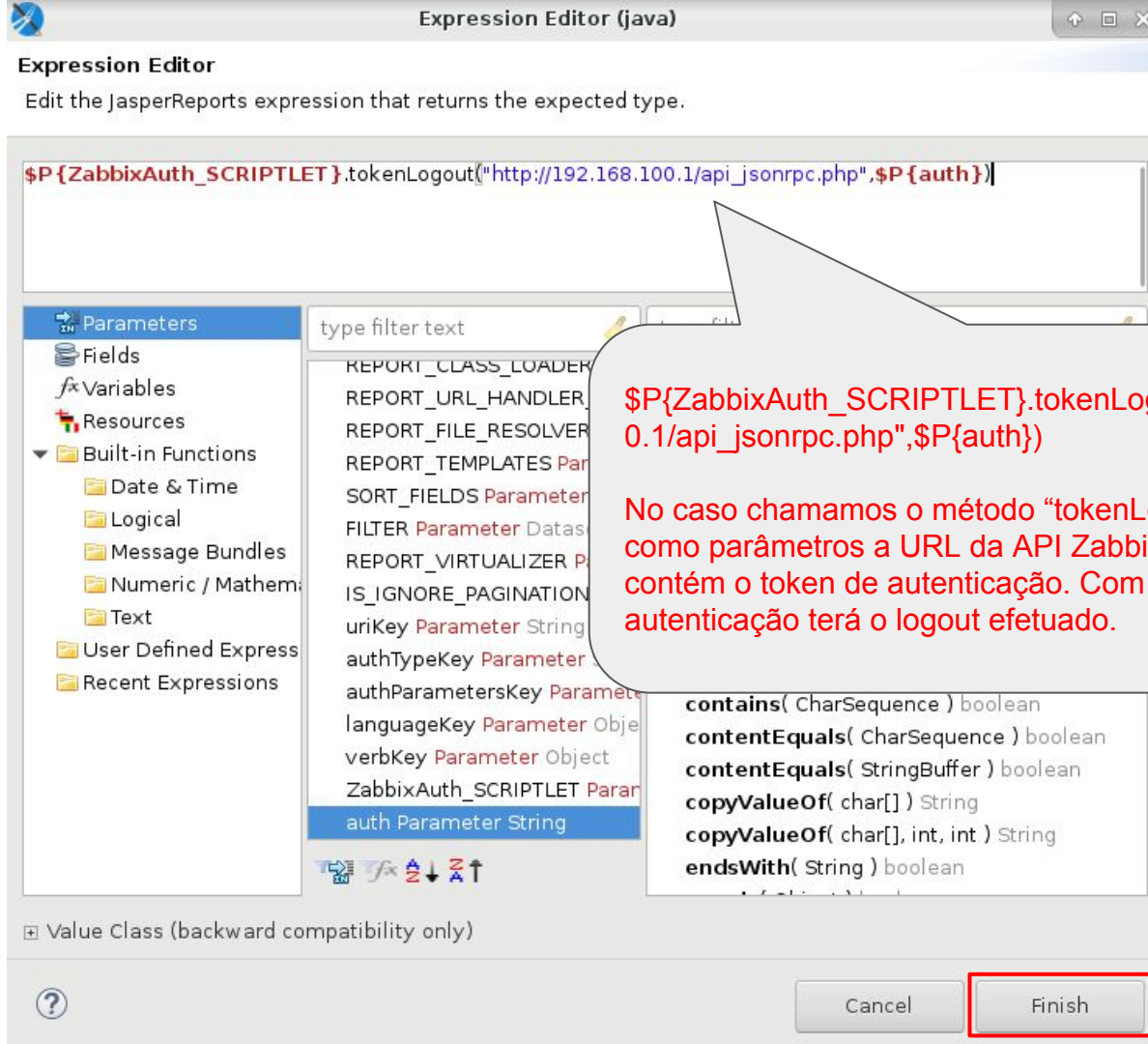
TextField: "Text Field"

Appearance Borders

Expression "Text Field"

Evaluation Time

Clique neste botão para editarmos a expressão que executará o logout do token de autenticação.



`$P{ZabbixAuth_SCRIPTLET}.tokenLogout("http://192.168.100.1/api_jsonrpc.php",$P{auth})`

No caso chamamos o método “tokenLogout” e passamos como parâmetros a URL da API Zabbix e o parâmetro que contém o token de autenticação. Com isso o token de autenticação terá o logout efetuado.

main.jrxml

Main Report

100%

Settings

0 1 2 3 4 5 6 7 8

host	ip
\$F{host}	\$F{ip}
Column Header	
Detail 1	
Column Footer	
Page Footer	
Summary	

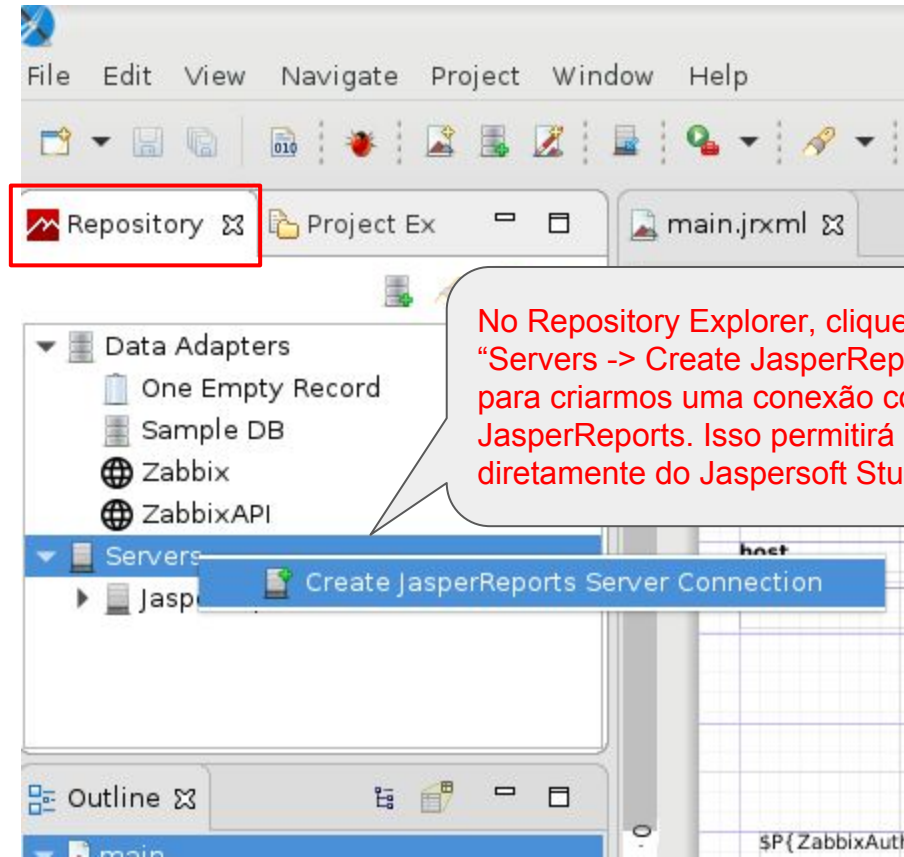
\$P{ZabbixAuth_SCRIPTLET}.tokenLogout("http://192.168.100.1/api_jsonrpc.php",\$P{auth})


Seção "Summary" com a ação de logout configurada.

Design Source Preview

Report State

Publicando o relatório no JasperReports Server



 **Server Profile Wizard** ↑ □ ×

JasperReports Server Access Configuration

Configure server connection

Name

URL


Account

Organization

User

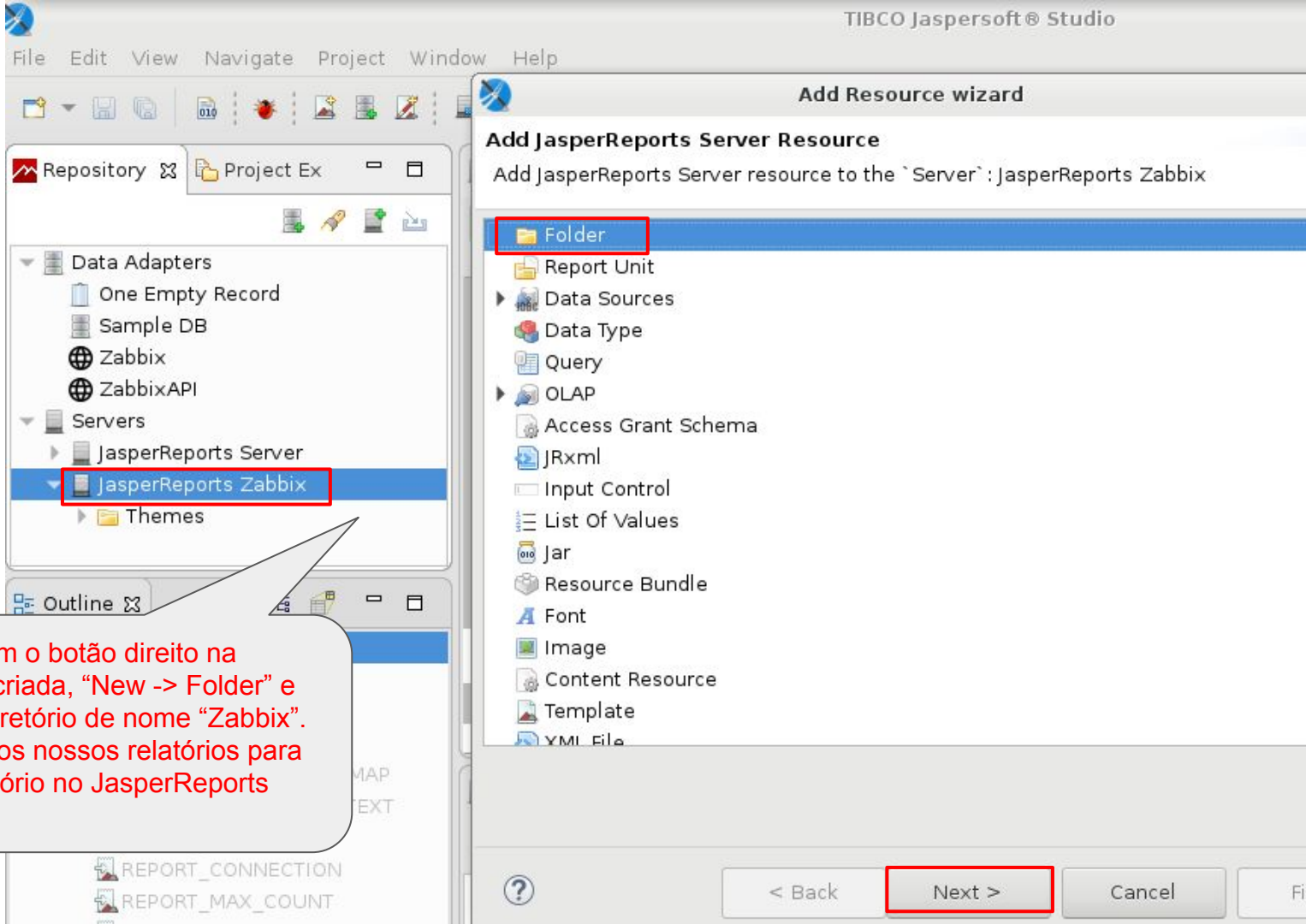
Password

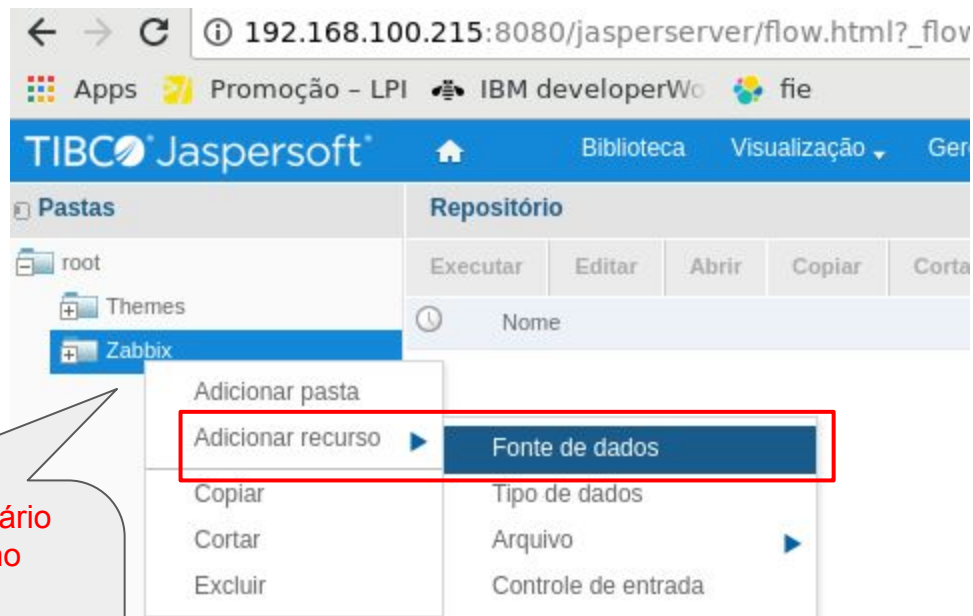
☒ **Advanced Settings**



Endereço do JasperReports Server.

Credenciais de acesso ao JasperReports Server.





Antes de prosseguir é necessário criar o Data Source também no JasperReports Server. Acesse o JasperReports Server, clique com o botão direito no diretório Zabbix (criado anteriormente) -> Adicionar recurso -> Fonte de dados.

Definir tipo e propriedades de fonte de dados

Selecione primeiro o tipo de fonte de dados que deseja adicionar e digite os valores de propriedades necessários.

Tipo: Web Service Data Source

Auth Type:

none

Language:

json

Request Verb:

post

Web Service URI:

http://192.168.100.1/

Password:

Username:

Preencha com os mesmos dados utilizados na criação do Data Source no Jaspersoft Studio.

Salvar

Cancelar

Salvar

Nome da fonte de dados (necessário):

ZabbixAPI

ID do recurso (necessário):

ZabbixAPI

Descrição

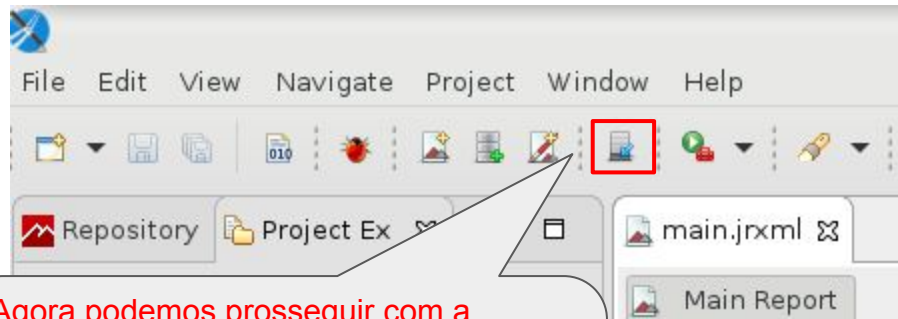
root

Themes

Zabbix

Salvar

Cancelar



Agora podemos prosseguir com a publicação do relatório.
Voltando ao Jaspersoft Studio, clique neste botão para iniciar o processo de publicação do relatório no JasperReports Server.

Report Publishing Wizard

Publish To JasperReports Server

Select JasperReports Server and location where report will be published

- ▼ JasperReports Zabbix
 - Themes
 - ▼ Zabbix
 - ZabbixAPI
 - JasperReports Server

☒ Create Report Unit

Report Unit Label

Report Unit Name (ID)

Report Unit Description

Selecione o diretório no qual o relatório será publicado no JasperReports Server.

Configure the data source

✖ There is a problem with selected Datasource which is not valid

Datasource **Query**

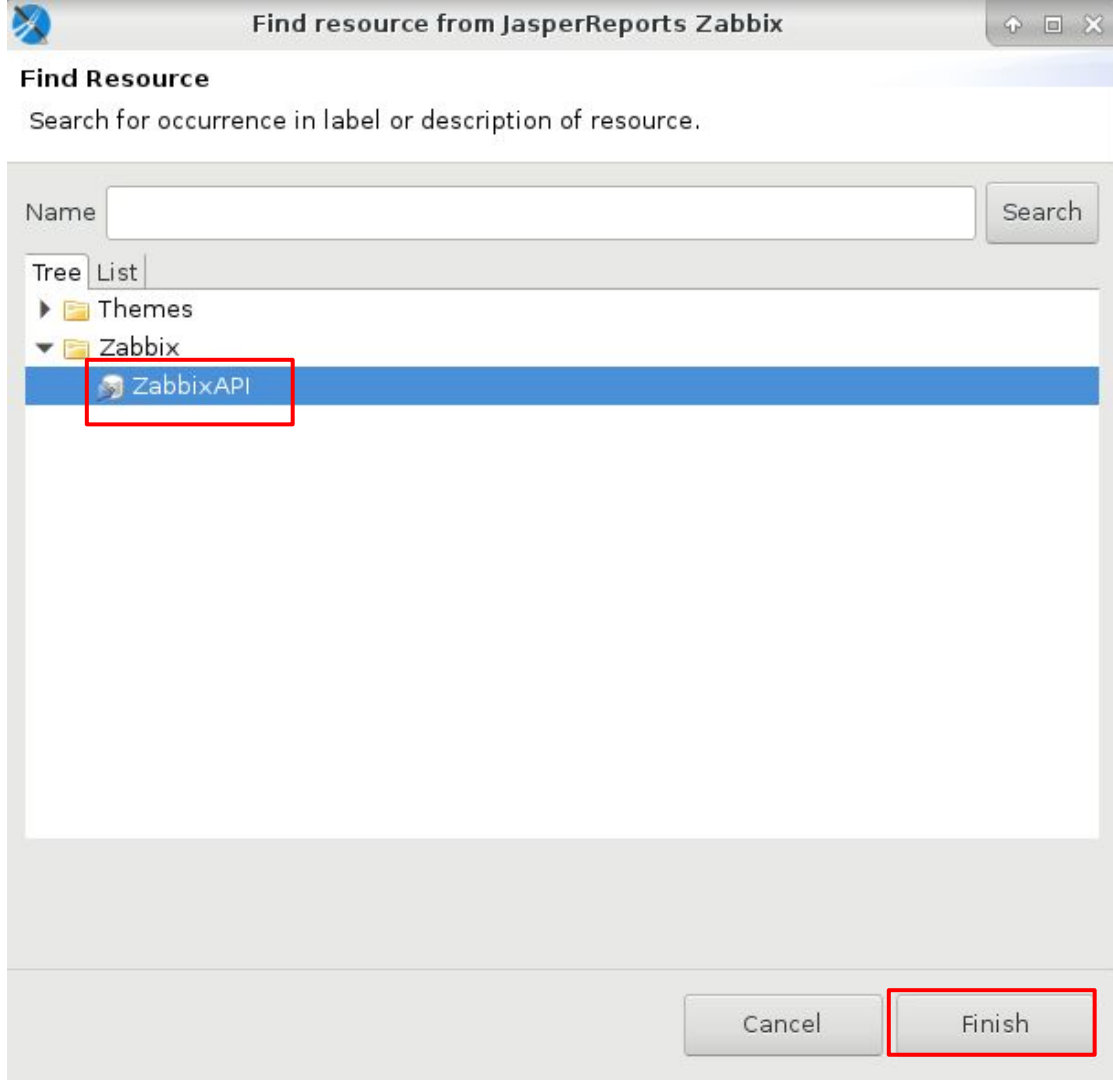
☒ Data Source from Repository

☐ Local Data Source

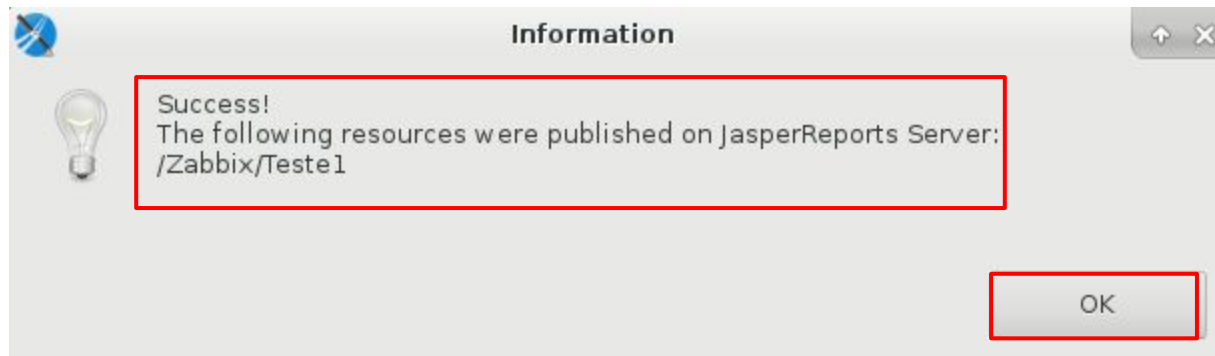
☐ Don't use any Data Source

Vamos selecionar o Data Source criado no JasperReports Server...

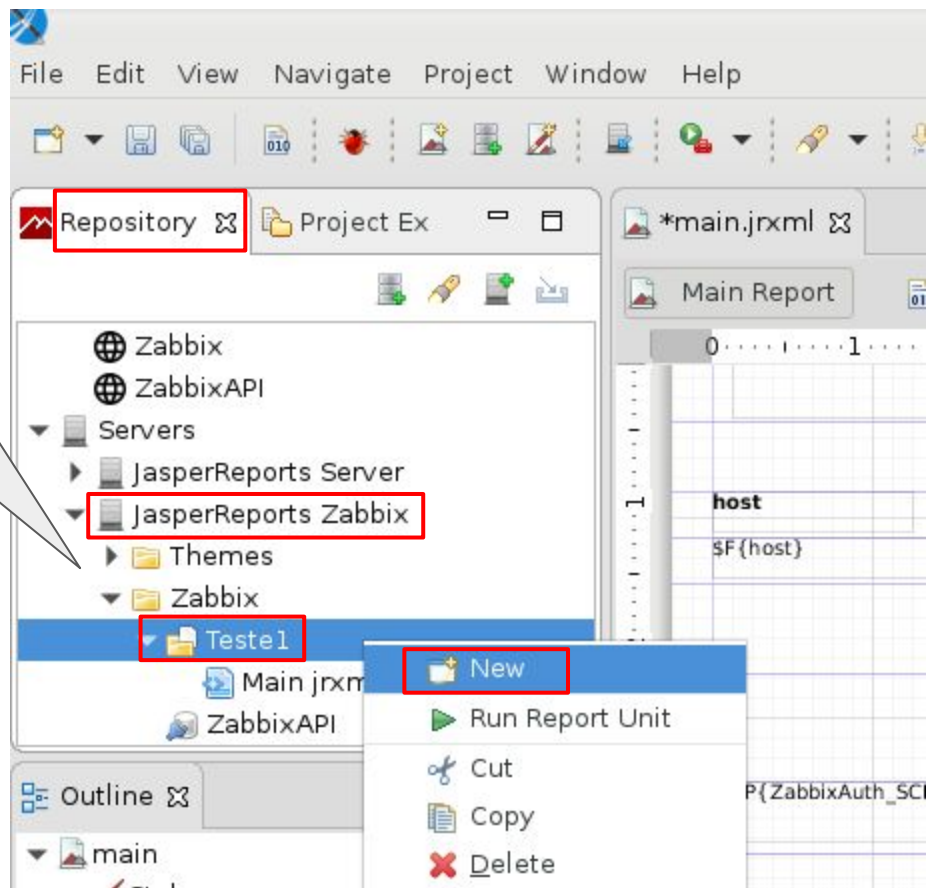
< Back **Next >** **Cancel** **Finish**

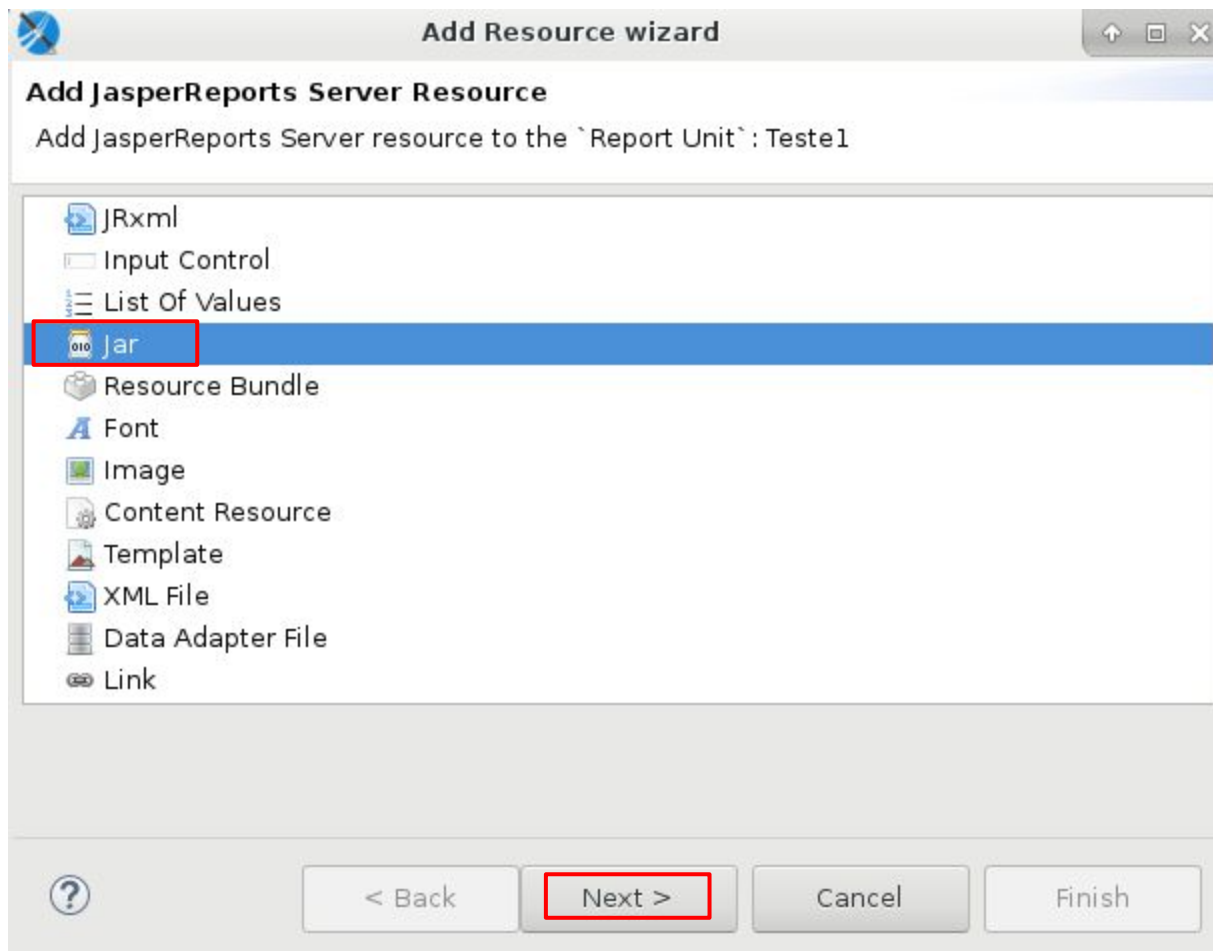




Após selecionar o Data Source, clique em Finish para concluir a publicação:



Para concluir resta apenas enviar o arquivo “.jar”, utilizado pelo Scriptlet de login/logout, como recurso para o relatório publicado.
Para isso clique com o botão direito no relatório -> New.





Add Resource wizard

Resource Editor
Resource Editor

Parent Folder

/Zabbix/Teste1_files

Type

jar

☐ is Reference

Permissions

Creation Date:

Update Date


Name

ZabbixAuthScriptlet.jar

ID

ZabbixAuthScriptlet.jar

Description

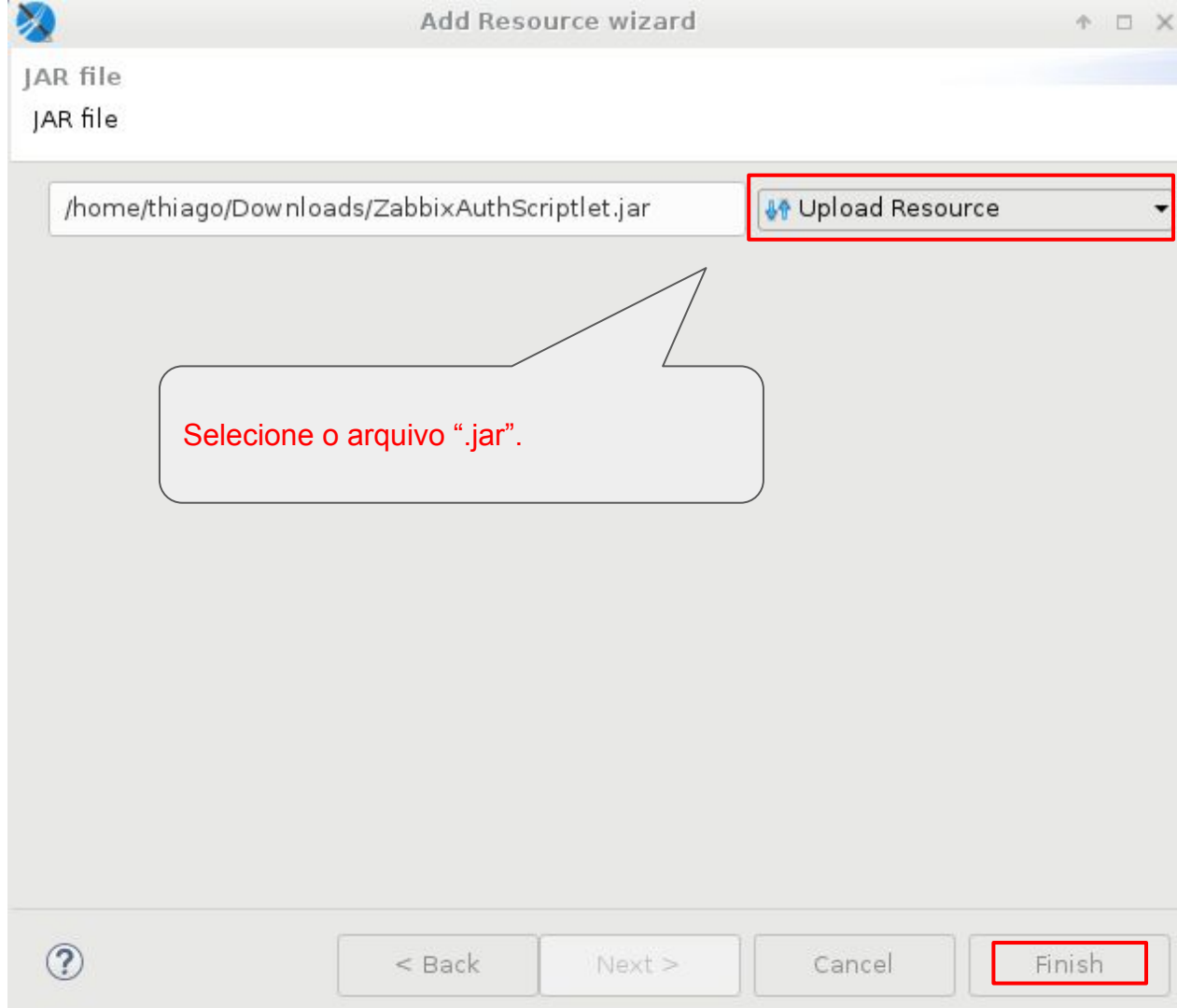


< Back

Next >

Cancel

Finish



← → ↻ ⓘ 192.168.100.215:8080/jasperserver/flow.html?_flowId=searchFlow

Apps Promoção - LPI IBM developerWo fie

TIBCO Jaspersoft 🏠 Biblioteca Visualização ▾ Gerenciar ▾

Pastas

- root
 - Themes
 - Zabbix**

Repositório

Executar	Editar	Abrir	Copiar	Cortar	Colar	Excluir	
🕒	Nome	Descrição					
	Teste1						
	ZabbixAPI						

Agora no JasperReports Server podemos abrir/executar o relatório publicado...

← → ↻ 192.168.100.215:8080/jasperserver/flow.html?_flowId=viewReportFlow&_flowId=viewReportFlow&ParentFolderUri=%2FZabbix&reportUnit=%2FZabbix%2FTeste1&... ☆

Apps Promoção - LPI IBM developerWo fie

TIBCO Jaspersoft Biblioteca Visualização Gerenciar jasperadmin User SAIR

Teste1 Dados atualizados 2018-02-04 em 19:56:01 ↺

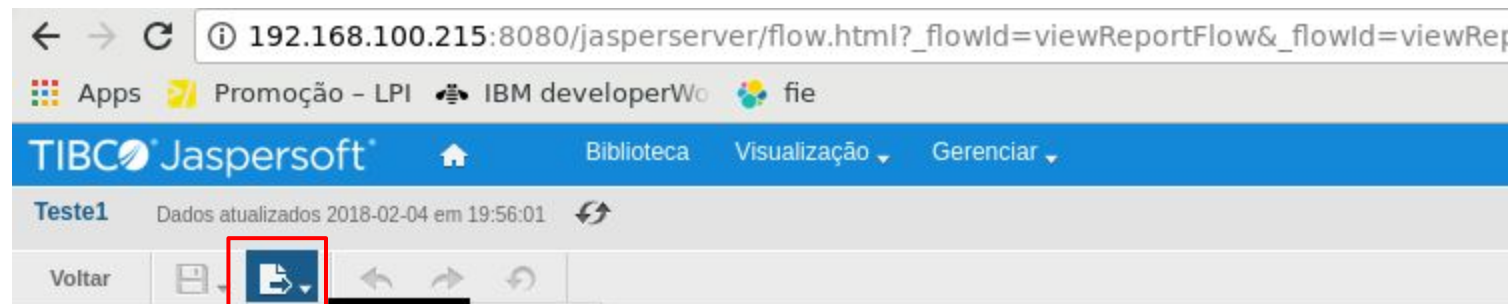
Voltar 100% pesquisar relatório

Zabbix API

host	ip
Zabbix server	["127.0.0.1"]
192.168.25.1	["192.168.25.1"]

...e o relatório será executado!

Assim os relatórios criados ficam acessíveis através de uma interface web com vários recursos de controle/gerenciamento à disposição!



O JasperReports Server permite exportar o relatório para diversos formatos.

Exportar

- Como Excel (Paginado)
- Como Excel
- Como CSV
- Como DOCX
- Como RTF
- Como ODT
- Como ODS
- Como XLSX (Paginado)
- Como XLSX
- Como PPTX

Zabbix API

host

Zabbix server
192.168.25.1

ip

["127.0.0.1"]
["192.168.25.1"]

The screenshot shows the TIBCO JasperSoft web interface. The address bar displays the URL: 192.168.100.215:8080/jasperserver/flow.html?_flowId=searchFlow. The browser's address bar also shows 'Apps', 'Promoção - LPI', 'IBM developerWo', and 'fie'. The interface has a blue header with the TIBCO JasperSoft logo and navigation links: 'Biblioteca', 'Visualização', and 'Gerenciar'. Below the header, there are two main sections: 'Pastas' (Folders) on the left and 'Repositório' (Repository) on the right. The 'Pastas' section shows a tree structure with 'root', 'Themes', and 'Zabbix'. The 'Repositório' section shows a table with columns 'Nome' and 'Descrição'. A table entry 'Teste1' is selected, and a right-click context menu is open over it. The menu options are: 'Executar', 'Executar em nova guia', 'Executar em segundo plano...', 'Agendar...' (highlighted with a red box), 'Editar', 'Copiar', 'Cortar', 'Excluir', 'Permissões...', and 'Propriedades...'. A speech bubble points to the 'Agendar...' option.

Além disso também é possível agendar a execução do relatório (clique com botão direito). O relatório executado poderá ser salvo em local determinado ou até enviado por e-mail.

Sobre envio de relatórios por e-mail:

https://community.jaspersoft.com/wiki/configuring-report-scheduler#Setting_Up_the_Connection_to_your_Outbound_Email_Server

<https://community.jaspersoft.com/wiki/how-configure-jasperreports-server-send-e-mails-gmail-mail-account>

Indo mais além - Subreports

Para aumentar as possibilidades e criar relatórios mais complexos podemos utilizar subreports (sub-relatórios).

Criamos até aqui um relatório que lista os hosts existentes em nosso Zabbix. Agora demonstrarei como adicionar um subreport que listará os itens monitorados para cada host listado no relatório principal.

The screenshot shows a report designer interface for a file named `*main.jrxml`. The main workspace displays a report template for "Zabbix API". The report structure is as follows:

- Page Header:** Contains the text "Zabbix API".
- Column Header:** Contains two columns: "host" with the value `$F{host}` and "ip" with the value `$F{ip}`.
- Detail 1:** An empty section intended for data details.
- Column Footer:** A section at the bottom of the table.


On the right side, there is a **Palette** with two categories:

- Basic Elements:** Includes "Frame" and "Subreport". The "Subreport" element is highlighted with a red box.
- Composite Elements:** Includes "Page Number", "Total Pages", and "Current Date".

A red arrow points from the "Subreport" element in the palette to the "Detail 1" section of the report. A speech bubble points to this area with the following text:

Aumente o tamanho da seção "Detail" e arraste um elemento "Subreport" para dentro da seção.


At the bottom of the interface, there are tabs for "Design", "Source", and "Preview".

Subreport

Please specify a report to be used as subreport, or just press finish to create an empty subreport element.


☒ Create a new report

☐ Select an existing report



Select a report file...

☐ Just create the subreport element



< Back

Next >

Cancel

Finish



Report file

Please select your reports file name with .jrxml extension.

Enter or select the parent folder:

ZabbixAPI



MyReports

ZabbixAPI

.settings

bin

zabbix

File name

sub.jrxml



< Back

Next >

Cancel

Finish

Project Window Help

Subreport criado!

*main.jrxml *sub.jrxml

Main Report 100% Settings

Page Header

Column Header

host ip

\$F{host} \$F{ip}

Detail 1

Design Source Preview

The screenshot shows the JReport Designer application. The top menu bar includes 'Project', 'Window', and 'Help'. Below it is a toolbar with various icons. The 'Project' window on the left shows a file tree with 'avaSE-1.8]' and 'ary'. The main workspace displays a report design on a grid. The design is divided into sections: 'Page Header', 'Column Header', and 'Detail 1'. The 'Column Header' section contains two columns labeled 'host' and 'ip', with data fields '\$F{host}' and '\$F{ip}' respectively. The 'Detail 1' section is highlighted with a red rectangle, indicating where a subreport is being created. A speech bubble points to the '*sub.jrxml' file tab in the 'Project' window, with the text 'Subreport criado!' (Subreport created!). The bottom status bar shows 'Design', 'Source', and 'Preview' tabs.

The screenshot shows the Jaspersoft Studio IDE interface. On the left, the 'Repository' pane shows a project named 'zabbix' with a subreport named 'ZabbixAPI'. Below it, the 'Outline' pane lists report components: Sort Fields, Variables, Scriptlets, Title, Page Header, Column Header, Detail 1 [125px], Column Footer, Page Footer, Last Page Footer, Summary, No Data, and Background. The main workspace displays a report design with a 'Main Report' section and a 'Detail 1' section. A red box highlights a button in the toolbar, and a callout bubble points to it with the following text:

No subreport clique neste botão para configurarmos o DataSet, pois a requisição do relatório principal busca por hosts e a requisição do subreport buscará por itens.

The bottom of the IDE shows the 'Report State' pane with tabs for Console, Errors, and Statistics.

Dataset and Query Dialog

ZabbixAPI

Query Java Bean

Language WebServiceQuery

Read Fields

Base URL api_jsonrpc.php

Content Type JSON

```
{
  "jsonrpc": "2.0",
  "method": "item.get",
  "params": {
    "output": "extend",
    "hostids": "10084",
    "sortfield": "name"
  },
  "auth": "a127b64a9ad3d78110534ecb58dab26a",
  "id": 1
}
```

Header Name Header Value

Add

Edit

Delete

Request Fields Source

Field Name	Class Type	Description
------------	------------	-------------

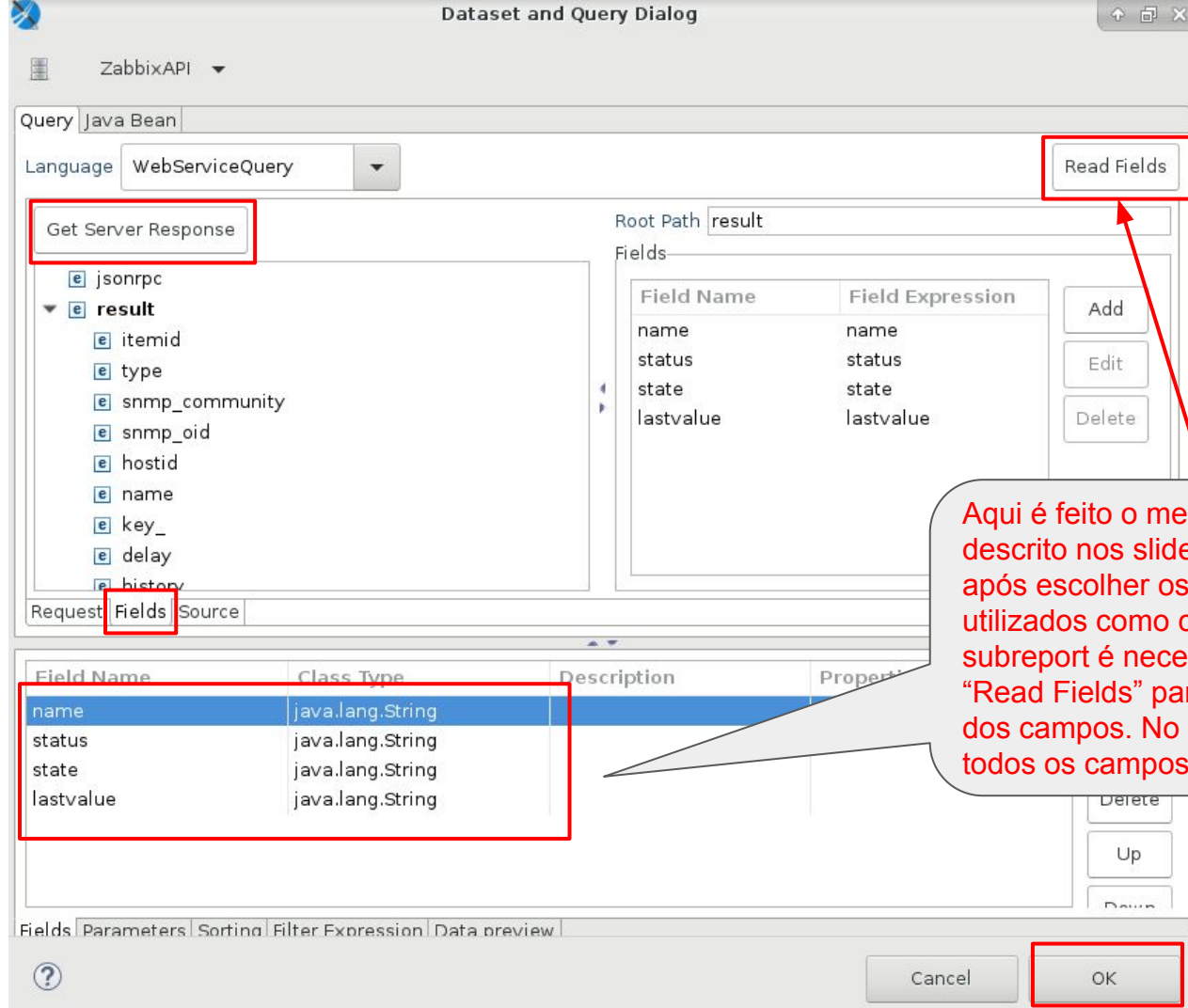
Fields Parameters Sorting Filter Expression Data preview

Cancel OK

Requisição JSON que será feita à API Zabbix.

Uso do [método item.get](#) da API Zabbix.

Para a configuração e validação inicial do DataSet fiz igual ao processo do [slide 16](#), informando um token de autenticação e desta vez informando também o ID do host "Zabbix server" para consulta dos itens.



Aqui é feito o mesmo processo descrito nos slides 17 ao 21, mas após escolher os nós a serem utilizados como campos no subreport é necessário clicar em "Read Fields" para mapeamento dos campos. No meu caso deixei todos os campos como String.

Repository Project

MyReports
zabbix
ZabbixAPI

- JRE System Library [javaSE-1.8]
- JasperReports Library
- Jaspersoft Server Library
- Referenced Libraries
- main.jasper
- main.jrxml
- sub.jrxml

Outline

sub

- Styles
- Parameters
- Fields
 - name
 - status
 - state
 - lastvalue
- Sort Fields
- Variables
- Scriptlets

Main Report

100%

name	status	state	lastvalue
\$F{name}	\$F{status}	\$F{state}	\$F{lastvalue}

Column Header
Detail 1

No subreport deixei ativas apenas as seções “Column Header” e “Detail” e arrastei os campos para a seção Detail.

Design Source Preview

Report State

Console Errors (0) Statistics

Compilation Time 0,008 sec

Project

*main.jrxml *sub.jrxml

ZabbixAPI Java Page 1 of 5

name	status	state	lastvalue
Agent ping	0	0	1
Available memory	0	0	3029147648
Checksum of \$1	0	0	208177056
Context switches per	0	0	8086
CPU \$2 time	0	0	84.5649
CPU \$2 time	0	0	0.3515
CPU \$2 time	0	0	1.8078
CPU \$2 time	0	0	0.0126
CPU \$2 time	0	0	
CPU \$2 time	0	0	
CPU \$2 time	0	0	
CPU \$2 time	0	0	
Free disk space on \$1	0	0	
Free disk space on \$1	0	0	3284344832

Design Source Preview

Report State

Ao clicar em "Preview" o subreport é executado e deverá listar os itens monitorados do host informado no DataSet.

Comunicação entre relatório principal e subreport

Como o subreport foi adicionado na seção “Detail” do relatório principal, ele será executado no loop para cada host listado.

Agora iremos informar ao subreport o token de autenticação gerado pelo Scriptlet, para que também possa consumir a API do Zabbix, e o ID do host no qual os itens deverão ser listados.

main.jrxml sub.jrxml

Main Report

100%

Page Header

Column Header

host ip

\$F{host} \$F{ip}

Detail 1

Design Source Preview

Report State

Console Errors (0) Statistics

Compilation Time	0,013 sec
Filling Time	0,283 sec
Report Execution Time	0,337 sec
Export Time	0 sec

No relatório principal, selecione o subreport e clique em "Edit Parameters".

Palette

- Basic Elements
 - Generic
 - Frame
 - Subreport
- Composite Elements
 - Page Number
 - Total Pages
 - Current Date

Properties Problems

Subrepo...breport Search Property

Appearance Subreport Advanced

Overflow type Stretch

Using Cache Default

Expression "sub.jasper"

Parameters Map Expression

Connection Expression \$P{REPORT_CONNEC

Source Expression

Edit Return Values Edit Parameters

Properties

Subreport Parameters

Subreport Parameters Map.

Name	Expression
auth	\$P{auth}
hostid	\$F{hostid}

Adicione dois parâmetros: um será o token de autenticação e o outro será o campo hostid de cada linha do relatório principal.

Buttons: Add, Edit, Delete, Up, Down, Copy From Master

Buttons: Cancel, Finish

File Edit View Navigate Project Window Help

Repository Project

main.jrxml sub.jrxml

Main Report

name	status
\$F{name}	\$F{status}

Outline

sub

Styles

Parameters

- Undo Set defaultValueExpression
- Redo
- Show Properties
- ☐ Sort Alphabetically
- ☐ Hide Default Parameters
- Create Parameter**
- Create Parameter Set

Os parâmetros repassados pelo relatório principal ao subreport devem existir no subreport para que possamos utilizá-los. No subreport, clique com o botão direito em Parameters -> Create Parameter.

TIBCO Jaspersoft® Studio

File Edit View Navigate Project Window Help

Repository Project

- zabbix
- ZabbixAPI
 - JRE System Library [javaSE-1.8]
 - JasperReports Library
 - Jaspersoft Server Library
 - Referenced Libraries
 - main.jrxml
 - sub.jasper
 - sub.jrxml
 - ZabbixAPI.xml

Outline

- uriKey
- authTypeKey
- authParametersKey
- languageKey
- verbKey
- auth
- hostid

Fields

- Sort Fields
- Variables
- Scriptlets

main.jrxml sub.jrxml

Main Report

name	status	state	lastvalue
SF(name)	SF(status)	SF(state)	SF(lastvalue)

Column Header
Detail 1

Parâmetros criados no subreport.

Report State

Console Errors (0) Statistics

Compilation Time	0,142 sec
Filling Time	0,244 sec

Palette

Basic Elements

- Note
- Text Field
- Static Text

Composite Elements

- Page Number
- Total Pages
- Current Date

Properties Problems

Parameter: hostid Search Property

Object Advanced

Name hostid

Class java.lang.String

Description

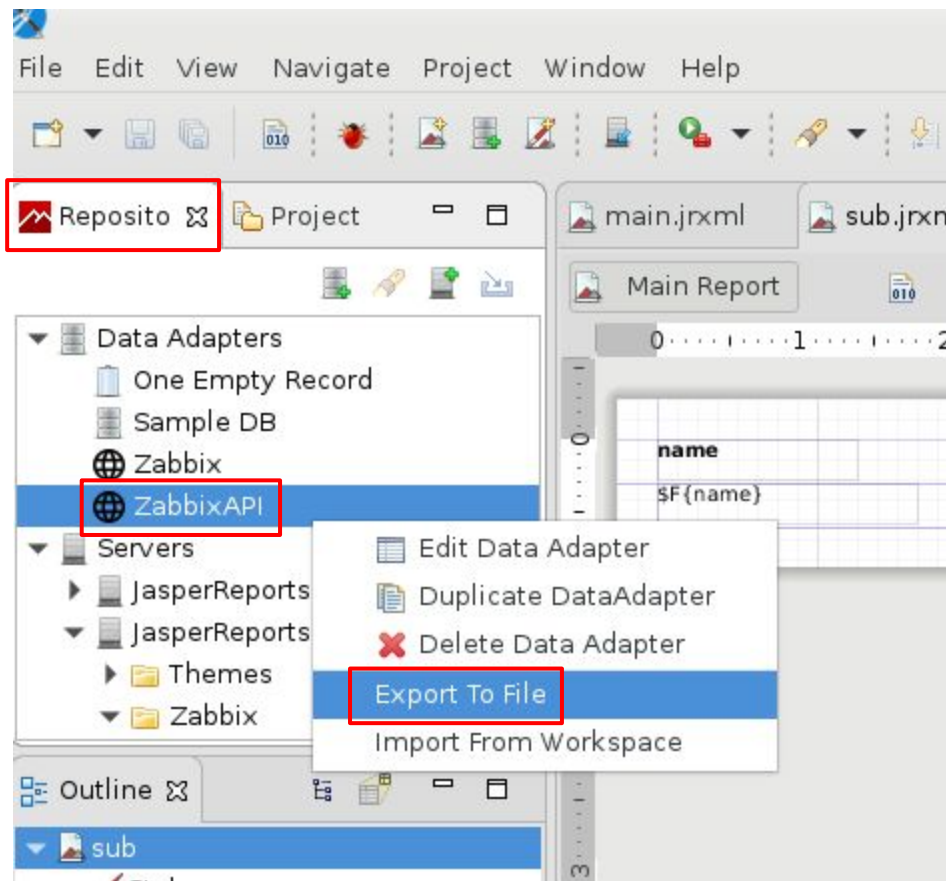
Is For Prompting

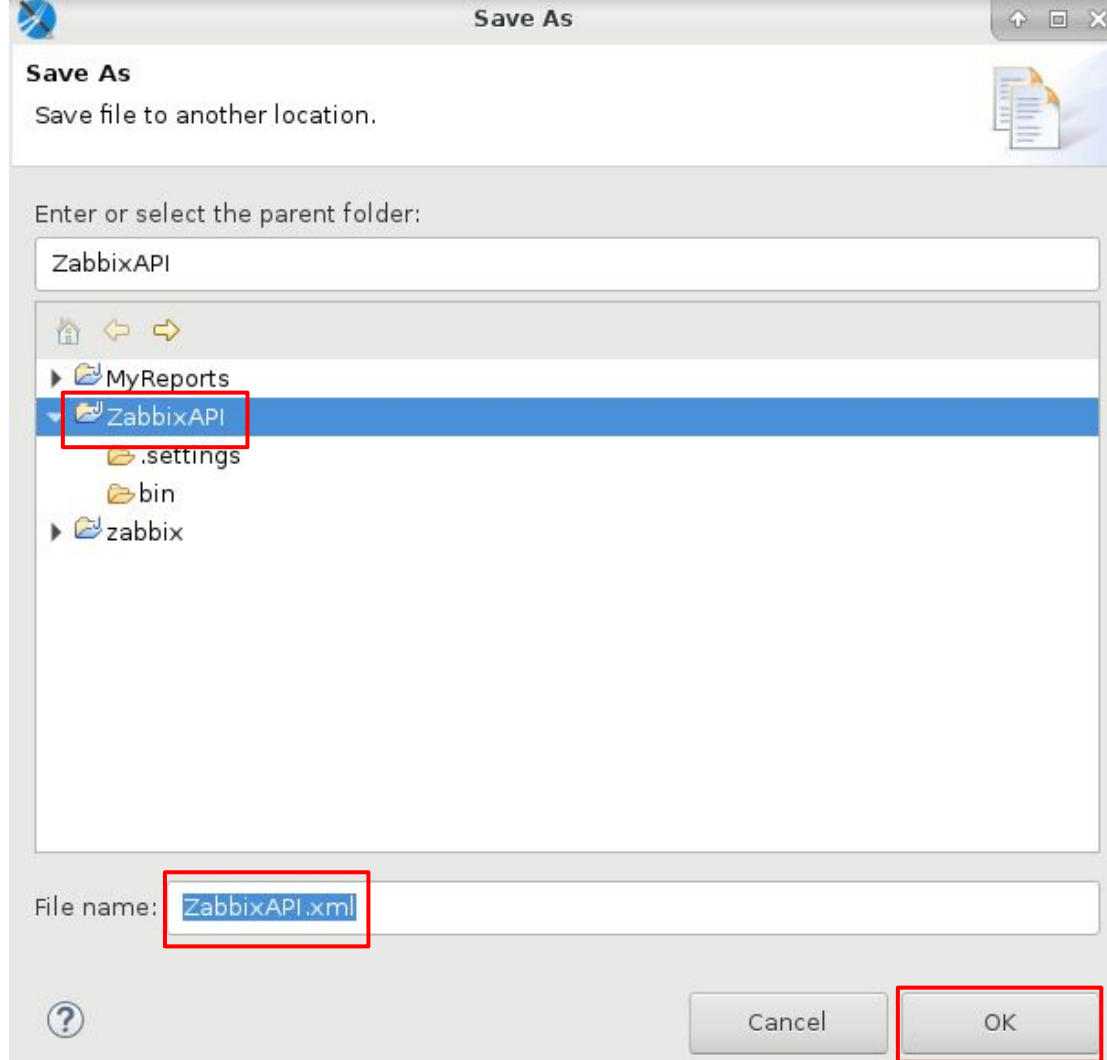
Default Value Expression

Evaluation Time <NULL>

No Jasper Reports o único tipo de Data Adapter que permite compartilhar a conexão de um relatório principal com outros subreports é o JDBC. Como no nosso caso o Data Adapter utilizado é do tipo Web Service, o subreport fará uma nova requisição à API do Zabbix e não terá como reutilizar a requisição feita no relatório principal (mais detalhes [aqui](#)).

Por isso, para não gerar erros no relatório com subreport, exportaremos o Data Adapter criado (ZabbixAPI) para um arquivo...





Repository Project

MyReports
zabbix
ZabbixAPI
JRE System Library [JavaSE-1.8]
JasperReports Library
Jaspersoft Server Library
Referenced Libraries
ZabbixAPI.xml
main.jrxml
sub.jasper

Outline

sub
Styles
Parameters
REPORT_CONTEXT
REPORT_PARAMETERS_MAP
JASPER_REPORTS_CONTEXT
JASPER_REPORT
REPORT_CONNECTION
REPORT_MAX_COUNT
REPORT_DATA_SOURCE
REPORT_SCRIPTLET

main.jrxml *sub.jrxml

Main Report

name	status	state	lastvalue
SF(name)	SF(status)	SF(state)	SF(lastvalue)

Column Header
Detail 1

Design Source Preview

Report State

Console Errors (0) Statistics

Compilation Time 0,007 sec

Properties Problems

Report: sub Search Property

Report Advanced

Dataset

When Resource Missing Type Null

Scriptlet Class

Resource Bundle

Default Data Adapter ZabbixAPI.xml

Edit query, filter and sort options

Nas propriedades do subreport defina como "Default Data Adapter" o arquivo exportado.

Dataset and Query Dialog

ZabbixAPI

QueryJava Bean

LanguageWebServiceQueryRead Fields

Base URLapi_jsonrpc.php

Content TypeJSON

Header NameHeader Value

Add

Edit

```
{
  "jsonrpc": "2.0",
  "method": "item.get",
  "params": {
    "output": "extend",
    "hostids": "$P{hostid}",
    "sortfield": "name"
  },
  "auth": "$P{auth}",
  "id": 1
}
```

RequestFieldsSource

Field Name	Class Type	Description	Properties
name	java.lang.String		
status	java.lang.String		
state	java.lang.String		
lastvalue	java.lang.String		

Add

Edit

Delete

Up

Down

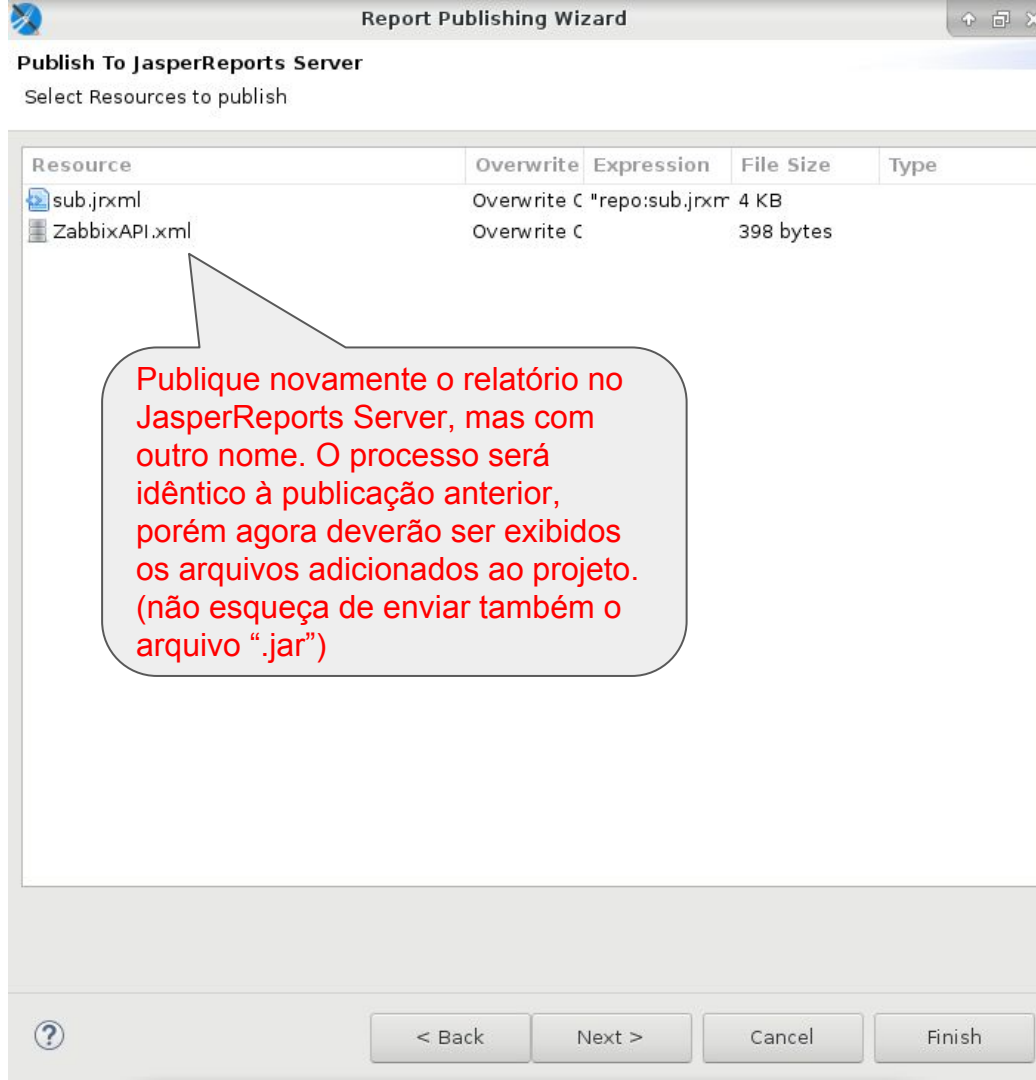
FieldsParametersSortingFilter ExpressionData preview

Cancel

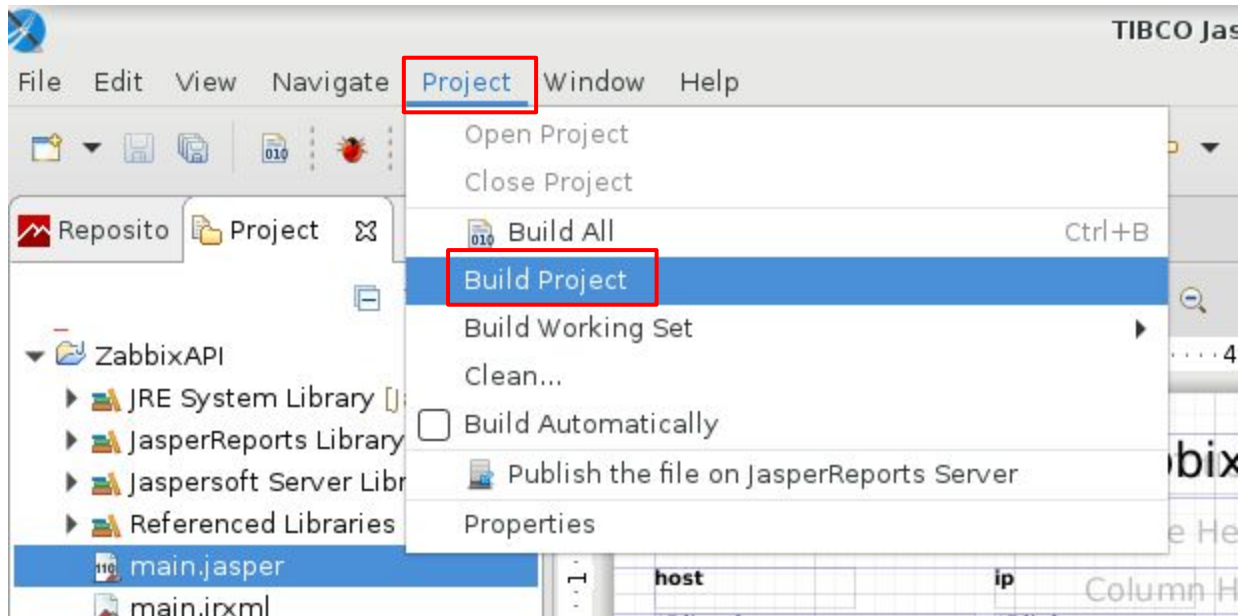
OK

Edite o DataSet do subreport e substitua o host ID e o token de autenticação pelos parâmetros que serão repassados pelo relatório principal.

81



Dica: caso algum comportamento estranho ou inesperado ocorra com o relatório, tente fazer o build do projeto para que os fontes sejam compilados.



Executando o relatório com subreport no JasperReports Server...

Zabbix API

host	ip
Zabbix server	["127.0.0.1"]

Host...

name	status	state	lastvalue
Agent ping	0	0	1
Available memory	0	0	3894771712
Checksum of \$1	0	0	208177056
Context switches per	0	0	9300
CPU \$2 time	0	0	80.4119
CPU \$2 time	0	0	0.3501
CPU \$2 time	0	0	2.1753
CPU \$2 time	0	0	0.0000
CPU \$2 time	0	0	0.1855
CPU \$2 time	0	0	0.0000
CPU \$2 time	0	0	2.1270
CPU \$2 time	0	0	12.9458
Free disk space on \$1	0	0	9881575424
Free disk space on \$1	0	0	3276693504
Free disk space on \$1	0	0	3276693504
Free disk space on \$1	0	0	3276693504
Free disk space on \$1	0	0	3276693504

Itens do host (subreport).

Conclusão

A maioria dos casos de uso que encontrei para criação de relatórios do Zabbix com Jasper Reports consultavam os dados do Zabbix diretamente do banco de dados. A vantagem de consumir os dados através da API do Zabbix é que se abstrai a modelagem do banco de dados, utilizando o recurso ideal para integração entre aplicações (API).

O Jasper Reports possui muitos recursos não citados neste documento, como [gráficos](#) e [autenticação de usuários em base externa \(LDAP\)](#)

Espero que o exemplo demonstrado tenha agregado algum conhecimento ao leitor e que possa ser adaptado para atender possíveis necessidades.

