

# JIRA 入门手册

## 目录

什么是 JIRA? .....	2
JIRA 功能概述: .....	2
• 可捕捉任何地方的问题: .....	2
• 可让你在所希望的方式下工作: .....	2
• 高效、强大的规划: .....	3
• 容易协作: .....	3
• 快速的可见性: .....	3
• 连接发展现实的策略.....	4
快速入门指南.....	5
在 Windows 系统上安装 JIRA.....	5
在 Linux 系统上安装 JIRA .....	8
使用文件包安装 JIRA.....	12
连接数据库.....	15
连接 PostgreSQL 数据库 .....	15
连接 MySQL 数据库 .....	18
连接 Oracle 数据库 .....	23
连接 SQL Server 数据库 .....	27
创建项目.....	32
编辑项目键.....	37
创建软件开发项目 .....	39
创建简单问题跟踪项目 .....	40
查看项目.....	42
创建问题.....	43
搜索问题.....	44
基础搜索.....	46
快捷搜索.....	49
高级搜索.....	52
高级搜索字段参考.....	54
共享搜索结果.....	86
更新概览.....	87
v6.4 版本更新.....	87
案例赏析.....	88
配置项目权限.....	88
JIRA Dashboards 入门演示.....	95
创建、共享和定制仪表板案例演示.....	95

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式:

电话: 023-66090381|400-700-1020

邮箱: [sales@evget.com](mailto:sales@evget.com)

网址: <http://www.evget.com/product/3516>

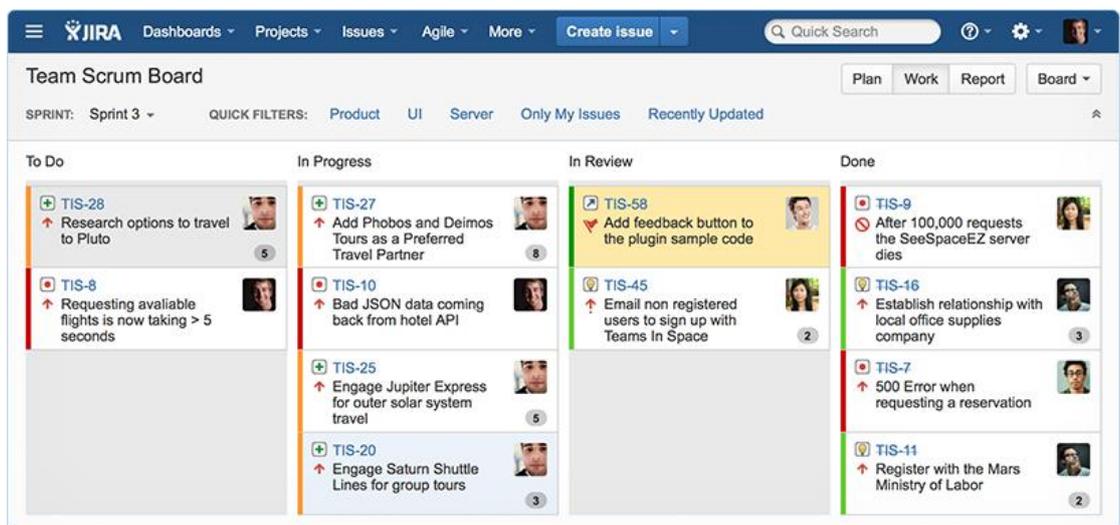
## 什么是 JIRA?

JIRA 是用于团队项目跟踪和构建伟大产品的企业软件。成千上万的团队选择 JIRA 捕获和组织问题，分配工作以及进行团队活动的召集。在你的办公桌面或者跟随你的新型移动设备，JIRA 协助你的团队完成工作。

## JIRA 功能概述:

- 可捕捉任何地方的问题:

人们的任务、想法和请求通常都会被隐藏在文档、大脑和收件箱里。使用 JIRA 可以捕获和组织你的团队成员的各种问题，优先考虑并实施重要的工作并保持记录最新的工作进度。



JIRA 使所有事情都变得容易，您的团队可以花更少的时间来管理工作并拥有更多的时间来构建伟大的产品。

- 可让你在所希望的方式下工作:

你的流程是你所做一切工作的核心。不要将你的团队工作硬塞进项目跟踪软件所定义的默认工作流程里。

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式:

电话: 023-66090381|400-700-1020

邮箱: [sales@evget.com](mailto:sales@evget.com)

网址: <http://www.evget.com/product/3516>

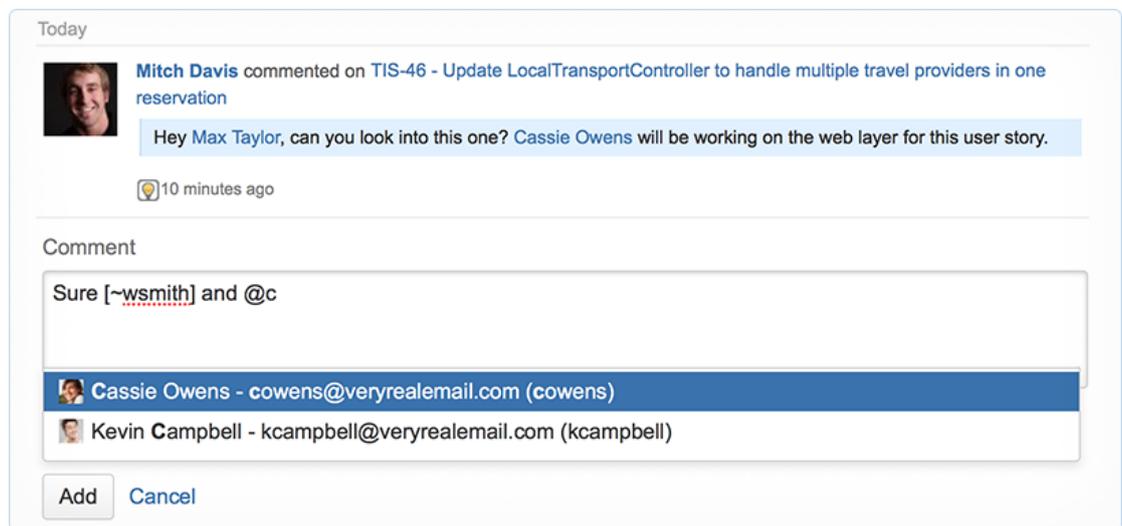
JIRA 工作流会自动匹配你现有的流程，你可以随着团队的发展轻易改变。你需要让工具来匹配你的团队工作，而不是让团队工作来适应工具。

- **高效、强大的规划：**

JIRA 可以协助团队捕获、分配和安排工作，使用他们变得更加高效。确保你的团队中的每个人都确切地知道需要做什么，然后再考虑如何将它们做得完美。JIRA 的 Agile 插件可以帮助你更好的完成计划冲刺与信息共享。

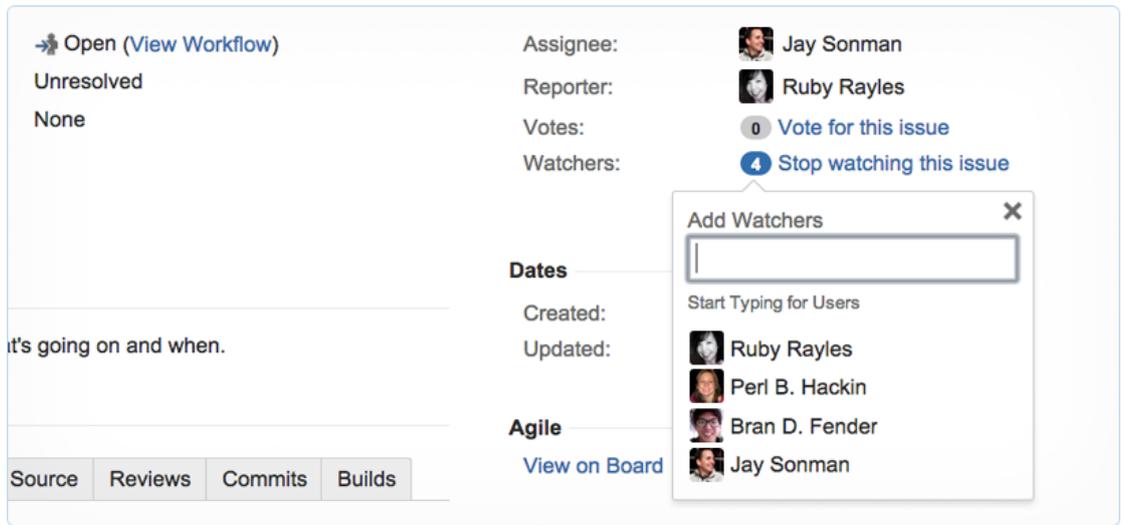
- **容易协作：**

在任何团队，成员之间的信息共享和能其从中获取帮助是非常重要的。JIRA 简单，直观的界面可以让你与队友协作并更有效地完成工作。



- **快速的可见性：**

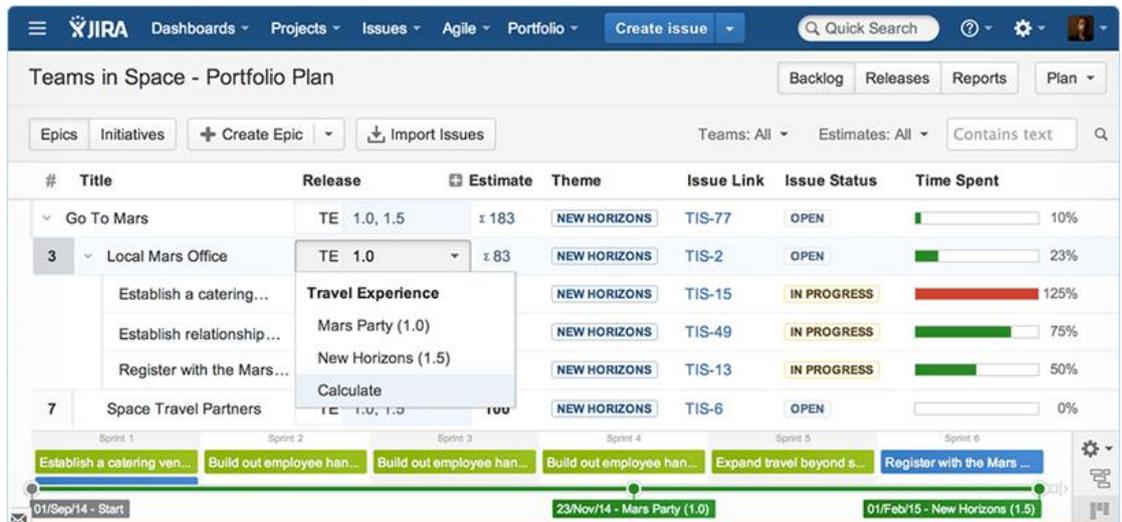
在 JIRA 不缺乏保持循环的方法，无论你是通过电子邮件接受更新、聊天还是用移动设备检查，JIRA 都可以满足你。



观看对你来说最重要的问题，监控活动流并使用项目仪表盘等分享信息，使用 JIRA 你可以立刻知道是怎么一回事。

- **连接发展现实的策略**

自动调度、灵活计划以及团队组织工作中的单独集会，JIRA Portfolio 是管理跨越多个团队和项目的最简单方法。



随着敏捷方法成为许多团队的标准开发实践，在其方法在扩展到整个业务领域的同时挑战也随之而来。JIRA Portfolio 是你的团队需要确保跨项目计划总是与时间表一致的理想工具。

## 【[在线免费试用申请](#)】

# 快速入门指南

根据具体的操作系统安装说明安装 JIRA。

- [在 Windows 系统上安装 JIRA](#)
- [在 Linux 系统上安装 JIRA](#)
- [使用文件包安装 JIRA](#)

## 在 Windows 系统上安装 JIRA

**⚠ 注意：**一些反病毒或其他互联网安全工具可能会干扰 JIRA 的安装过程，并阻止安装过程成功完成。为了防止你遇到这样的问题，请在进行 JIRA 安装前禁用此工具。

两种使用 Windows Installer 安装 JIRA 的方法：

- 使用安装向导
- 执行无值守安装

### 使用安装向导

如果你是在服务器上首次安装 JIRA 或希望指定安装选项，那么使用安装向导吧。如果已经通过安装向导安装过 JIRA，并希望再次使用相同的安装选项重新安装 JIRA，那么你可以采用“无值守模式”重新安装 JIRA（详见下文），并且无需任何用户输入。

#### 1. 下载并运行 JIRA “Windows Installer”

✔ 要安装 JIRA 作为服务器，Windows Installer 必须以 Windows 管理员身份运行否则你的安装选项将受限。

1.1. 下载 JIRA “Windows Installer” 文件（.exe）。

1.2. 运行 “.EXE” 文件，启动安装向导。

✔ 如果 Windows7（或 Vista）的“User Account Control”对话框提示是否允许对计算机进行更改，点击“Yes”。否则安装向导访问到您的操作系统以及任何后续安装选项将会受限。

1.3. 在“Upgrading JIRA?”这一步，选择“Express Install”或“Custom Install”：

**Express Install** - 如果选择此选项，JIRA 将按照安装向导下一步默认设置安装。如果你想手动选择这些选项，点击“Back”按钮，选择“Custom Install”。

**Custom Install** - 如果选择此项，JIRA 将提示你选择以下选项（在安装向导的后续步

---

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

骤都预先填充使用默认值):

- .在“Destination Directory”中安装 JIRA
- .每一个 JIRA 安装的 JIRA Home Directory 必须唯一
- .Windows 的“Start”菜单文件夹选项
- .JIRA 将贯穿所有 TCP 端口（HTTP 和一个控制端口）
- .如果以管理员身份运行安装程序，系统将提示“Install JIRA as a service”(推荐)。

你也可以稍后手动执行此操作，详见 [JIRA 作为 Service 运行指南](#)。

- ✔ 如果你安装 JIRA 作为一个服务，在 Windows 命令提示符运行 start-jira.bat 无法启动 JIRA，你必须通过 Windows “开始”菜单启动 JIRA。

#### 1.4.当 JIRA 安装完成后会自动启动 JIRA。

并且如果你选择了启动到浏览器窗口，JIRA 也将自动启动到浏览器窗口。

#### ❗ 注意：

.如果你选择安装 JIRA 作为一个服务， JIRA 将以 Windows'SYSTEM'用户身份运行。要想改变用户身份，请参阅[更改 JIRA service 用户](#)。

.如果你没有安装 JIRA 作为一个服务，那么一旦开始，JIRA 将以安装 JIRA 的 Windows 用户身份运行。

.如果你是為了输出而在 Windows Server 运行 JIRA，我们强烈建议您创建一个专用的用户帐户（例如，使用用户名“JIRA”）来运行 JIRA。

.有关创建一个专用的用户帐户和编写权限设置，参阅 [guidelines](#) 详细信息。

- ✔ 如果你是在 Microsoft Active Directory 下操作你的 Windows Server，让 Active Directory 的管理员创建一个专用的用户帐户（新权限）来运行 JIRA。

.如果 JIRA 是作为 service 安装，运行时不要忘了更改 JIRA 服务运行的专用用户帐户。

## 2.启动 JIRA

如果 JIRA 尚未启动，您可以使用相应的 Windows “Start”菜单快捷方式或命令提示符选项启动 JIRA。一旦 JIRA 开始，你可以从相应的 Windows “Start”菜单的快捷方式或通过任何计算机上的浏览器访问你的 JIRA 服务器。

### 2.1 Windows “Start”菜单的快捷方式

程序安装将创建下列 Windows “Start”菜单的快捷方式：

.Access JIRA—打开一个网页浏览器窗口来访问你的 JIRA 应用程序。

- ❗ 你的 JIRA server 必须已启动该快捷方式。

.Start JIRA Server —启动运行在 JIRA 的安装的 Apache Tomcat 应用服务器，这样就可以通过 Web 浏览器访问 JIRA。

.Stop JIRA Server —关闭运行 JIRA 安装的 Apache Tomcat 应用服务器。在你选择此快捷方式后，将无法通过 Web 浏览器来访问 JIRA。

.Uninstall JIRA —从 Windows 操作系统卸载 JIRA。

### 2.2 通过命令提示符启动和关闭 JIRA

---

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

输入 JIRA 安装目录的 bin 子目录，并运行相应的文件：

.start-jira.bat (启动 JIRA)

.stop-jira.bat (关闭 JIRA)

**i** 如果你按照我们的指南以一个专用的用户帐户来运行 JIRA，然后以用户帐户（例如“jira”）使用 runas 命令来执行启动 jira.bat 运行 JIRA。例如：

```
>runas /env /user:<DOMAIN>\jira start-jira.bat
```

(<DOMAIN>是你的 Windows 域名或计算机名)

### 2.3 从浏览器访问 JIRA

你可以从任何一个支持的 Web 浏览器的计算机访问 JIRA 以及访问此网址：

http://计算机上安装 JIRA 的名称或 IP 地址:安装 JIRA 指定的 HTTP 端口号

**i** 如果 JIRA 没有出现在你的网页浏览器，你可能需要[更改 JIRA 运行的端口](#)。

## 3. 运行 Setup Wizard

## 4. Next Steps

4.1 见 [JIRA 101](#) 开始创建项目，创建用户，并自定义你的 JIRA 实例。

4.2 如果你没有安装 JIRA 作为一个服务器，每次重启电脑时都要手动启动 JIRA。要将 JIRA 安装修改成作为服务器运行，请参阅 [Run JIRA as a Service](#)。

4.3 为了使 JIRA 效果最优，请参阅 [Optimizing Performance](#)。

## 执行无值守安装

如果你已经使用安装向导（如上）安装了 JIRA，你可以使用此 JIRA 安装配置文件（response.varfile）重新“无人值守模式”安装而无需任何用户输入。

如果你以前安装的 JIRA 用于测试目的，并且你需要在基于相同配置的多个服务器计算机上安装 JIRA，那么无值守模式下安装 JIRA 可以节省你的时间。

**⚠ 注意：**

·该 response.varfile 文件包含以前的 JIRA 安装时的指定选项。因此，不要卸载以前安装的 JIRA。

·如果你要修改 response.varfile 文件，请确保指定的所有目录路径是绝对的，例如，  
sys.installationDir= C:\: \\ Program Files\\ Atlassian\\JIRA

·如果在该文件中指定任何相对目录路径那么无值守安装将会失败。

### 在无值守模式下下载并运行 JIRA 'Windows Installer'

1. 从 [JIRA 下载中心](#) 将 JIRA 'Windows Installer' 文件（.exe）下载到一个合适的位置。
2. 打开 Windows 命令提示符，然后执行命令提示符中的其余步骤。
3. 将位于以前的 JIRA 安装目录.install4j 的子目录里的 response.varfile 文件复制到和下载的“Windows 安装程序”文件相同的路径。

---

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- i** 你可以在此步骤之后卸载以前安装的 JIRA。如果需要在多台计算机上安装 JIRA 那么请保存 response.varfile。
4. 更改目录到“Windows Installer”文件的位置并且执行以下命令：  
atlassian-jira-X.Y.exe -q -varfileresponse.varfile
    - 。 X.Y 一指的是你将要安装 JIRA 的版本。
    - 。 -q 一指示在无人参与模式下运行安装程序 ('quietly')。
    - 。 -varfileresponse.varfile 一指安装程序使用的包含的配置选项的配置文件。配置文件的位置和名称应在 -varfileoption 之后指定。
  5. 当安装完成后 JIRA 将自动启动如上第 2 步启动 JIRA。

## 在 Linux 系统上安装 JIRA

**!** 注意：一些反病毒或其他互联网安全工具可能会干扰 JIRA 的安装过程，并阻止安装过程成功完成。为了避免遇到这样的情况，请在进行 JIRA 安装前关闭此类工具。

两种使用 Linux Installer 安装 JIRA 的方法：

- 使用控制台向导
- 执行无值守安装

### 使用控制台向导

如果你是在服务器上首次安装 JIRA 或希望指定安装选项，那么使用安装向导吧。如果已经通过安装向导安装过 JIRA，并希望再次使用相同的安装选项重新安装 JIRA，那么你可以采用“无值守模式”重新安装 JIRA（详见下文），并且无需任何用户输入。

#### 1. 下载安装 JIRA “Linux Installer”

如果你用 'root' 用户权限执行 Linux 安装程序，安装程序会创建一个 JIRA 专用用户帐户。你也可以不用 'root' 用户权限执行 Linux 安装程序，但安装选项将受限，并且安装程序将无法创建一个专用的用户帐户运行 JIRA。要运行 JIRA 作为服务，Linux 的安装程序必须以 'root' 用户权限执行。

1.1 从 JIRA 下载页面下载相应的 JIRA ‘Linux 64 位/32 位安装文件’ (.bin 文件)。

**i** 注意：

- 要访问 32 位安装包，你需要点击 'JIRA Download' 页面上的 'Show all' 链接来访问其它安装包。

- 64 位/32 位的 .bin 安装差异涉及到运行 JIRA 的 Java 平台。在运行相似水平的性能时，64 位与 32 位相比需要更多的内存。这是因为 64 位的 Java 平台引用对象的大小是 32 位的两倍。

1.2 打开 Linux 控制台，并更改目录到 “.BIN” 文件所在的目录。

⚠️ 如果下载的 “.BIN” 文件是不可执行的，那么使其可执行，例如：

```
chmod+xatlassian-jira-X.Y.bin
```

（其中 X.Y 指 JIRA 的版本）

1.3 执行 “.BIN” 文件，启动控制台向导。

1.4 当提示 ‘Express Install’，‘Custom Install’ 或 ‘Upgrade an existing JIRA installation’ 时，选择 ‘Express Install’ 或 ‘Custom Install’ 选项：

**Express Install**–如果选择此选项，JIRA 将按照下一步的默认设置安装。

📘 注意：

- 如果以 ‘root’ 用户权限安装程序，JIRA 将被安装为服务。
- 如果你想手动选择这些选项：
  - i. 输入 ‘e’ 退出控制台。
  - ii. 再次执行控制台向导（上面的步骤 1.3）。
  - iii. 选择 ‘Custom Install’ 选项。

**Custom Install**–如果选择此选项，JIRA 将提示选择以下选项（在控制向导的后续步骤都预先填充使用默认值）：

- 在 “Destination Directory” 中安装 JIRA
- 每个 JIRA 安装的 JIRA Home Directory 必须唯一
- JIRA 将贯穿所有 TCP 端口（HTTP 和一个控制端口）
- 如果以 ‘root’ 用户权限执行 Linux 安装程序，系统推荐安装 JIRA 作为服务器。你也可以稍后手动执行，详见[在 Linux 系统上启动 JIRA](#)

1.5 安装结束后会自动启动 JIRA

📘 注意：

- 如果以 ‘root’ 用户权限执行的 Linux 安装程序，Linux 安装程序将创建一个专门用来运行 JIRA 的 Linux 账户 ‘jira’，密码为空。此帐户的权限有：
  - i. 对 [JIRA Home Directory](#) 完整的写入权限。
  - ii. 对 [JIRA Installation Directory](#) 部分写入权限。
- ‘root’ 用户捆绑安装具有默认的掩码（0022 或 002），如果没有设置可能会导致如下安装问题：

📌 JIRA-32435 - JIRA Linux Installer does not set files with correct permissions when a non-default umask is used **VERIFIED**

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

---

请确保在安装前已设置。

- 如果不是以 'root' 用户权限执行的 Linux 安装程序, JIRA 仍然可以 'root' 权限运行。但是为了操作系统的安全性, 不建议这样做。
- 你可能需要根据 JIRA 实例的大小来增加可用的操作系统文件的最大数量。详见 [Loss of Functionality due to Too Many Open Files Error](#)。

## 2. 启动 JIRA

如果 JIRA 尚未启动, 你可以在 Linux 控制台使用适当的命令来启动 JIRA。JIRA 一旦启动, 可以通过任何计算机上的浏览器进行访问。

### 2.1 手动启动和关闭 JIRA

如果以 'root' 权限的用户在 Linux 控制台安装的 JIRA, 这些命令将需要 'root' 权限的用户执行:

- `service jira start` (启动 JIRA)
- `service jira stop` (关闭 JIRA)

如果不是以 'root' 权限的用户在 Linux 控制台安装 JIRA, 输入 JIRA 安装目录的 bin 子目录, 并执行相应的文件:

- `start-jira.sh` (启动 JIRA)
- `stop-jira.sh` (关闭 JIRA)

当下列信息出现在应用程序的日志文件时, JIRA 将可以通过浏览器访问:

```
*****
... You can now access JIRA through your web browser.
*****
```

### 2.2 从浏览器访问 JIRA

你可以从任何一个支持的 Web 浏览器的计算机访问 JIRA 以及访问此网址:

`http://计算机上安装 JIRA 的名称或 IP 地址:安装 JIRA 指定的 HTTP 端口号`

 注意:

- 如果你的网页浏览器连接不到 JIRA, 你可能需要更改 JIRA 运行的端口。

- 应用服务器日志（如 Apache Tomcat）将被写入到 JIRA 安装目录中的 logs/catalina-YYYY-MM-DD.log 文件中。

### 3. 运行 Setup Wizard

### 4. Next Steps

1. 见 [JIRA 101](#) 开始创建项目，创建用户，并自定义你的 JIRA 实例。
2. 如果你没有安装 JIRA 作为一个服务器，每次重启电脑时都要手动启动 JIRA。要将 JIRA 安装修改成作为服务器运行，请参阅[在 Linux 上启动 JIRA](#)。
3. 为了使 JIRA 效果最优，请参阅 [Optimizing Performance](#)。

## 执行无值守安装

如果你已经使用控制台向导（如上）安装过 JIRA，你可以使用 JIRA 的配置文件（response.varfile）通过“无人值守模式”无需任何输入重新安装 JIRA。如果你以前安装的 JIRA 用于测试目的，并且你需要在基于相同配置的多个服务器计算机上安装 JIRA，那么无值守模式下安装 JIRA 可以节省你的时间。

### 注意：

- 该 response.varfile 文件包含以前的 JIRA 安装时的指定选项。因此，不要卸载以前安装的 JIRA。
- 如果你要修改 response.varfile 文件，请确保指定的所有目录路径是绝对的，例如：  
sys.installationDir=/opt/atlassian/jira  
如果在该文件中指定任何相对目录路径那么无值守安装将会失败。

## 在无值守模式下下载和运行 JIRA 'Linux Installer'

1. 从 [JIRA 下载中心](#) 将 JIRA 'Linux Installer' 文件 (.bin) 下载到一个合适的位置。
2. 打开 Linux 控制台。
3. 将位于以前的 JIRA 安装目录里的 .install4j/response.varfile 文件复制到和下载的 'Linux Installer' 文件相同的路径。  
 你可以在此步骤之后[卸载以前安装的 JIRA](#)。如果需要在多台计算机上安装 JIRA 那么请保存 response.varfile。
4. 更改目录 (cd) 到 'Linux Installer' 文件的位置，并执行以下命令：  
atlassian-jira-X.Y.bin -q -varfileresponse.varfile
  - X.Y 一指的是你将要安装 JIRA 的版本
  - -q 一指示在无人参与模式下运行安装程序 ('quietly')
  - -varfileresponse.varfile 一指安装程序使用的包含的配置选项的配置文件。配置文件的位置和名称应在 -varfileoption 之后指定。

5. 当安装完成后 JIRA 将自动启动如上第 2 步启动 JIRA。

## 使用文件包安装 JIRA

### 首先

在安装前确保你已经[安装 Java 并设置了 JAVA\\_HOME](#)。另请参阅[Supported Platforms](#)了解其他支持 JIRA 的平台详细信息。

 Linux 发行版通常会安装一个名为 Gcj 的 JIRA 平台。不要使用此平台—这是不完整的，将使 JIRA 无法成功运行。

### 1. 下载并解压缩 JIRA 存档文件

1.1 下载符合你操作系统的相应的文件夹 (Windows 是 'zip', Linux/Solaris 是 'tar.gz')。

1.2 解压下载的文件夹

 Windows 建议使用 7-Zip 解压工具。

 Solaris 使用 GNU tar 处理长名文件比系统默认的 tar 效果更好。

### 2. 在 JIRA 中设置 JIRA Home Directory

方法如下：

- 编辑 jira-application.properties 文件，将“jira.home”的属性值设置为 [JIRA Home Directory](#) 的路径（这个路径应不同于 application directory，否则后面可能会遇到问题）。如果要在 Windows 上指定这个路径，子目录之间需使用双回斜线（“\\”）。例如：

X:\\path\\to\\JIRA\\Home

 如果是在 Windows 系统上定义 UNC 路径，注意开始的双斜回线。例如：

\\\\\\machinename\\path\\to\\JIRA\\home

 要想找到这个文件的具体位置详见 [JIRA Installation Directory](#)。

- 在你的操作系统设置名为 JIRA\_HOME 的环境变量，其值是你的 JIRA Home Directory 路径。步骤如下：

- **Windows**

- 通过 Windows 用户界面配置此环境变量（通常是通过“我的电脑”或“计算机”）
- 在命令提示符下，在运行 JIRA 之前，输入以下命令（用你自己的 JIRA Home 路径）：

```
set JIRA_HOME=X:\path\to\JIRA\Home
```

 设置 JIRA\_HOME 环境变量值的格式如下：

- X 是 JIRA Home Directory 放置的驱动盘
- ' ' 旁边无空格

- 指定上面的命令来启动 JIRA。
- **Linux/Solaris**
  - 运行 JIRA 前在 shell/console 控制台输入以下命令：  
export JIRA\_HOME=/path/to/jira/home
  - 指定上面的脚本命令来启动 JIRA。

你可以在硬盘为你的 JIRA home directory 指定任何位置，一定要是一个绝对路径。

**i** 注意，不能在相同的 JIRA home directory 里有多个实例。建议将你的 JIRA Home Directory 放在完全独立于 JIRA Installation Directory 的路径（非嵌套），这将最大限度地减少大操作时的信息丢失（如备份和恢复）。

### 3. 在操作系统上创建一个运行 JIRA 的专用帐户

**i** 如果你正在评估 JIRA，这一步是可选的，但将 JIRA 作为 production 安装时是强制性的。

建立一个运行 JIRA 的专用帐户并且以此帐户运行 JIRA，否则 JIRA 可能会被滥用。方法如下：

- 如果你的操作系统是\*nix-based（如 Linux 或 Solaris），在控制台输入：  
\$ sudo /usr/sbin/useradd --create-home --comment "Account for running JIRA"  
--shell /bin/bash jira
- 如果你的操作系统是 Windows
  1. 通过下面任一种方式创建一个专用帐户：
    - 在 Windows 命令行键入以下内容：  
> net user jiramypassword /add /comment:"Account for running JIRA"  
（这将创建一个用户名'jira'，密码'mypassword'的帐户，且密码可修改）
    - 打开 Windows 的“计算机管理”控制台增加用户名为'jira'，密码自定的帐户。
  2. （可选）使用 Windows 的“计算机管理”控制台删除'jira'用户所有 Windows 组里不必要的成员关系，如默认的'Users'组。
    - ✓**如果是在 Microsoft Active Directory 操作 Windows，让 Active Directory 管理员给你创建一个名为'jira'的帐户。

确保只有以下目录可以被写入到这个专用帐户（'jira'）：

- JIRA 安装时推荐使用的 JIRA Installation Directory 子目录：
  - logs
  - temp
  - work

- 你的 JIRA Home Directory.

⚠ 不要让此专用帐户对 JIRA Installation Directory 有写入权限。另参阅 [Tomcat security best practices](#)。

## 4. 启动 JIRA

输入 JIRA 安装目录的 bin 子目录，并执行相应的文件开始运行 JIRA:

- start-jira.sh (Linux/Solaris)
- start-jira.bat (Windows)

📘 以上面创建的专用帐户（'jira'）运行 JIRA

- **Windows:** 用 runas 命令运行 start-jira.bat. 例如:  
runas /env /user:\jira start-jira.bat  
(是你的 Windows 域名或计算机名)
- **Linux:** 运行 start-jira.sh 前用 su 命令切换到 'jira' 帐户 (或以 'jira' 帐户用 su 命令直接运行 start-jira.sh)

当下列信息出现在应用程序的日志文件时，JIRA 将可以通过浏览器访问:

```
*****  
... You can now access JIRA through your web browser.  
*****
```

你可以从任何一个支持的 Web 浏览器的计算机访问 JIRA 以及访问此网址:

- http://计算机上安装 JIRA 的名称或 IP 地址:安装 JIRA 指定的 HTTP 端口号

📘 如果你的网页浏览器连接不到 JIRA，你可能需要更改 JIRA 运行的端口。

📘 日志文件将被写到 logs/catalina.out.

## 5. 运行 Setup Wizard

详见[运行 Setup Wizard](#)。

## 6. Next Steps

1. 见 [JIRA 101](#) 开始创建项目，创建用户，并自定义你的 JIRA 实例。

2. 如果希望每次重启电脑时自动启动 JIRA。详见[在 Linux 上自动启动 JIRA](#) 或 [将 JIRA 作为 Service 运行](#) (Windows 系统)。
3. 默认情况下，通过文件包安装的 JIRA 使用标准的 Tomcat 端口（即 8080）。如需更改端口请参阅[更改 JIRA 的 TCP 端口](#)。
4. 为了使 JIRA 效果最优，请详见 [Optimizing Performance](#)。

## 连接数据库

JIRA 需要一个相关联的数据库来存储问题数据。

如果是设置完全新安装的 JIRA 的话，JIRA Setup Wizard 提供数据库的噢连接配置，无论是 JIRA 内部的 HSQL 数据库还是外部的数据库。

**i** JIRA 的内部的 HSQL 数据库适用于评估目的，但 HSQL 容易造成数据污染。因此强烈建议连接 JIRA 到其它支持的数据库类型，可以从分利用数据库系统本身的备份和恢复功能。

下面是配置 JIRA 数据库连接的详细说明：

- [连接 PostgreSQL 数据库](#)
- [连接 MySQL 数据库](#)
- [连接 Oracle 数据库](#)
- [连接 SQL Server 数据库](#)

## 连接 PostgreSQL 数据库

### 首先

- 检查是否支持你的 PostgreSQL 版本。请参见[支持的平台](#)。
- 如果是转移 JIRA 到另一台服务器，先导出数据为 XML 备份，然后将旧数据库中的数据传传输到新的数据库。
- 在开始前关闭 JIRA，除非你正在运行 Setup Wizard。

### 1. 创建并配置 PostgreSQL 数据库

- 创建与 JIRA 连接的数据库用户（登录角色），如：**jiradbuser**。
- 使用 Unicode 排序规则创建一个存储 JIRA 问题的数据库，如：**jiradb**。  
`CREATE DATABASE jiradb WITH ENCODING 'UNICODE' LC_COLLATE 'C' LC_CTYPE 'C' TEMPLATE template0;`  
或通过命令行：  
`$ createdb -E UNICODE -l C -T template0 jiradb`
- 确保该用户拥有连接数据库以及创建和编辑表的权限。

## 2. 将 PostgreSQL JDBC 驱动程序复制到你的应用服务器(仅 JIRA WAR)

**⚠️** 如果安装了包含 PostgreSQL JDBC 驱动的 JIRA 'Recommended' 分布可跳过此步。JIRA WAR 分布不包含该驱动。

1. 从 <http://jdbc.postgresql.org/download.html> 下载 PostgreSQL JDBC 驱动。
2. 添加 PostgreSQL JDBC 驱动程序 jar 到应用服务器的 lib/目录下。

## 3. 配置 JIRA 服务器来连接到 PostgreSQL 数据库

三种配置 JIRA 服务器链接到 PostgreSQL 数据库的方法:

- **使用 JIRA 安装向导** (不适用于 JIRA WAR) - 如果你已经安装 JIRA 并且是第一次设置, 你的设置将被保存到 JIRA 主目录的 dbconfig.xml 文件里。
- **使用 JIRA 配置工具** (不适用于 JIRA WAR) - 如果有已存在的 JIRA 实例, 你的设置将被保存到 JIRA 主目录的 dbconfig.xml 文件里。
- **手动配置** - 只有当你有 JIRA WAR 实例或控制台时使用这种方法链接到 JIRA 服务器。你需要手动更新 JIRA 主目录里的 dbconfig.xml 文件。

每种配置方法的说明:

<p><b>JIRA setup wizard</b></p>	<p>当你首次在浏览器上访问 JIRA 时, JIRA setup wizard 才会显示。</p> <ol style="list-style-type: none"> <li>1. 在最开始页面的 'Configure Language and Database' 设置 <b>Database Connection</b> 为 <b>My own database</b>。</li> <li>2. 设置 <b>Database Type</b> 为 <b>PostgreSQL</b>。</li> <li>3. 按照下面介绍的数据库连接字段部分描述填写字段。</li> <li>4. 测试连接并保存。</li> </ol>
<p><b>JIRA configuration tool</b></p>	<ol style="list-style-type: none"> <li>1. 按如下指示运行 JIRA 配置工具:  <b>Windows:</b> 打开命令提示符, 运行 JIRA 安装目录的 bin 子目录文件 config.bat。  <b>Linux/Unix:</b> 打开控制台, 运行 JIRA 安装目录的 bin 子目录文件 config.sh。  <b>i</b> 可能会出现失败的错误, 参照此 <a href="#">文章</a> 的解决方法。</li> <li>2. 导航到 <b>Database</b> 选项卡, 设置 <b>Database type</b> 为 <b>PostgreSQL</b>。</li> <li>3. 按下面描述的数据库连接字段填写字段。</li> <li>4. 测试连接并保存。</li> <li>5. 重启 JIRA。</li> </ol>
<p><b>Manually</b></p>	<ol style="list-style-type: none"> <li>1. 找到 JIRA 主目录根目录下的 dbconfig.xml 文件。如果该文件不存在, 创建该文件, 将 XML 示例 (如下) 复制粘贴到该文件。</li> <li>2. 根据下面数据库连接字段描述更新该文件。通过在每一个末端添加 'amp;' 转义所有 '&amp;' 字符。  <b>i</b> 注意: &lt;database-type/&gt;元素必须指定数据库类型。</li> </ol>

	<p>例：如果你正在使用 PostgreSQL 9.2。如果你忘记指定数据库类型就启动 JIRA，那么数据库表将不会被正确创建。若遇到类似问题，参见<a href="#">指定不正确的数据库类型</a>进行解决。</p> <p>3. 保存文件并重启 JIRA。</p>
--	--

#### 数据库连接字段

Setupwizard/ configuration tool	dbconfig.xml	描述
Hostname	位于<url>标记（例如下面的粗体文字） <url>jdbc:postgresql:// <b>dbserver</b> :5432/jiradb</url>	安装 PostgreSQL 服务器的机器名称或 IP 地址。
Port	位于<url>标记（例如下面的粗体字） <url>jdbc:postgresql://dbserver: <b>5432</b> /jiradb</url>	PostgreSQL 服务器正在监听的 TCP/IP 端口。也可以留空来使用默认端口。
Database	位于<url>标记（例如下面的粗体字） <url>jdbc:postgresql://dbserver:5432/ <b>jiradb</b> </url>	PostgreSQL 数据库（JIRA 数据将保存的数据库）的名称。应该在上面第一步时创建。
Username	位于<username>标记（例如下面的粗体字） <username> <b>jiradbuser</b> </username>	JIRA 用于连接 PostgreSQL 服务器的用户。应在上面第一步时创建。
Password	位于<password>标记（例如下面的粗体字） <password> <b>jiradbuser</b> </password>	用于与 PostgreSQL 服务器身份验证的用户密码。
Schema	位于<schema-name>标记（例如下面的粗体字） <schema-name> <b>public</b> </schema-name>	PostgreSQL 数据库使用的模式名称。 PostgreSQL 7.2 及以后的版本要求在 <schema-name/>中指定模式。如果你的 PostgreSQL 使用默认 'public' 模式，这需要在 <schema-name/>元素中指定。 确保你的数据库模式名

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

		称是小写，因为 JIRA 不支持 PostgreSQL 数据库中的模式名称包含大写字母。
--	--	--

dbconfig.xml 文件示例

```
<?xml version="1.0" encoding="UTF-8"?>

<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>postgres72</database-type>
  <schema-name>public</schema-name>
  <jdbc-datasource>
    <url>jdbc:postgresql://dbserver:5432/jiradb</url>
    <driver-class>org.postgresql.Driver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <validation-query>select version();</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

    <pool-test-on-borrow>false</pool-test-on-borrow>
    <pool-test-while-idle>true</pool-test-while-idle>

  </jdbc-datasource>
</jira-database-config>
```

## 4.启动 JIRA

现在 JIRA 连接到 PostgreSQL 数据库就配置好了。下一步就是启动 JIRA。

- 如果正在使用 JIRA 的'recommended'分配，启动它，并观察日志中是否存在错误。
- 如果使用的是 JIRA WAR 分配，重建并重新部署在应用服务器的 Web 应用程序。

## 连接 MySQL 数据库

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

## 首先

- 查看 MySQL 的版本是否支持，查阅[支持的平台](#)。
- 如果是转移 JIRA 到另一台服务器，先导出数据为 XML 备份，然后将旧数据库中的数据传输到新的数据库。
- 如果你打算用同一个 MySQL 服务器安装 Confluence 和 JIRA，请阅读[Confluence MySQL 安装指南](#)，配置你的 MySQL 服务器来适应 Confluence 以及 JIRA。注意 Confluence 要求比 JIRA 要严格，因此应该配置 MySQL 来适应 Confluence，这种配置同样也适用于 JIRA。
- 在开始前关闭 JIRA，除非你正在运行 Setup Wizard。

## 1. 创建并配置 MySQL 数据库

1. 创建与 JIRA 连接的数据库用户（登录角色），如：**jiradbuser**。  

```
mysql> CREATE USER '<USERNAME>'@'<JIRA_SERVER_HOSTNAME>' IDENTIFIED BY '<PASSWORD>';
```
2. 创建一个数据库（如 **jiradb**）来存储 JIRA 的问题，数据库必须有一个 UTF8 的字符集。在 MySQL 命令客户端输入以下命令：  

```
CREATE DATABASE jiradb CHARACTER SET utf8 COLLATE utf8_bin;
```
3. 确保该用户拥有连接数据库以及创建和写入表的权限。授权命令如下：  

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER,INDEX on <JIRADB>.* TO '<USERNAME>'@'<JIRA_SERVER_HOSTNAME>' IDENTIFIED BY '<PASSWORD>';
```

```
flush privileges;
```

### Tip:

为了确认权限是否被授予成功，用 JIRA DB 用户登录到 DB 服务器并运行一下命令：  

```
SHOW GRANTS FOR <USERNAME>@<JIRA_SERVER_HOSTNAME>;
```

## 2. 将 MySQL JDBC 驱动程序复制到应用服务器

如果是升级 JIRA 并且使用的是推荐的 MySQL 驱动器(Connector/J JDBC driver v5.1)，可跳过此步。JIRA 更新会自动复制已存在的驱动到升级后的安装。

### 将 MySQL JDBC 驱动程序复制到你的应用服务器

1. 获得 MySQL 驱动程序：
  - 1.1 如果是**安装 JIRA**，下载推荐的 MySQL 驱动器 Connector/J JDBC driver v5.1。可通过选择'Platform Independent'选项下载.tar.gz 或.zip 文件。从存档中为驱动提取 jar（例如：mysql-connector-java-5.x.x-bin.jar）。
  - 1.2 如果是**升级 JIRA** 并且没有使用推荐的 MySQL 驱动(JDBC Connector/J 5.1)，在更新前先从 JIRA 备份驱动程序。驱动程序将存储在<JIRA 安装目录>/lib/ 目录。

2. 复制 MySQL JDBC 驱动程序到<JIRA 安装目录>/lib/ 目录来更新 JIRA。如果使用 Windows installer 安装 JIRA，你需要在运行 Windows installer 后，但在运行 Setup Wizard 之前做这一步。
3. 重启 JIRA/JIRA 服务器。
4. 如果正在安装 JIRA，跳过该页面的其他指示，通过浏览器连接 JIRA 来运行 Setup Wizard。

**注意：**

上面推荐的 MySQL Connector/J 驱动，有用户报告遇到 Resin JDBC driver for MySQL 的问题。

### 3.配置 JIRA 服务器来连接 MySQL 数据库

三种配置 JIRA 服务器连接 MySQL 数据库的方法：

- **使用 JIRA 安装向导**（不适用于 JIRA WAR）-如果你已经安装 JIRA 并且是第一次设置，你的设置将被保存到 JIRA 主目录的 dbconfig.xml 文件里。
- **使用 JIRA 配置工具**（不适用于 JIRA WAR）-如果有已存在的 JIRA 实例，你的设置将被保存到 JIRA 主目录的 dbconfig.xml 文件里。
- **手动配置**-只有当你有 JIRA WAR 实例或控制台时使用这种方法链接到 JIRA 服务器。你需要手动更新 JIRA 主目录里的 dbconfig.xml 文件。

每种配置方法的说明：

<b>JIRA setup wizard</b>	<p>当首次在浏览器连接 JIRA 时，JIRA 设置向导就会出现。</p> <ol style="list-style-type: none"> <li>1. 在最开始页面的'Configure Language and Database'设置 <b>Database Connection</b> 为 <b>My own database</b>。</li> <li>2. 设置 <b>Database Type</b> 为 <b>MySQL</b>。</li> <li>3. 按照下面描述的数据库连接字段填写字段。</li> <li>4. 测试连接并保存。</li> </ol>
<b>JIRA configuration tool</b>	<ol style="list-style-type: none"> <li>1. 根据下面运行 JIRA 配置工具：  <b>Windows:</b>打开命令提示符，运行 JIRA 安装目录的 bin 子目录文件 config.bat。  <b>Linux/Unix:</b>打开控制台，运行 JIRA 安装目录的 bin 子目录文件 config.sh。  <span style="color: blue;">i</span>可能会出现失败的错误，参照此<a href="#">文章</a>的解决方法。</li> <li>2. 导航到 <b>Database</b> 选项卡设置 <b>Database type</b> 为 <b>MySQL</b>。</li> <li>3. 按照下面描述的数据库连接字段填写字段。</li> <li>4. 测试连接并保存。</li> <li>5. 重启 JIRA。</li> </ol>
<b>Manually</b>	<ol style="list-style-type: none"> <li>1. 找到 JIRA 主目录根目录下的 dbconfig.xml 文件。如果该文件不存在，创建该文件，将 XML 示例（如下）复制粘贴到该文件。</li> <li>2. 根据下面数据库连接字段描述更新该文件。通过在每一个末端添加'amp;'转义所有'&amp;'字符。</li> </ol>

	<p><b>i</b>注意：&lt;database-type/&gt;元素必须指定数据库类型，如：<b>mysql</b>。如果你忘记指定数据库类型就启动 JIRA，那么数据库表将不会被正确创建。若遇到类似问题，参见指定不正确的数据库类型进行解决。</p> <p>3. 保存文件并重启 JIRA。</p>
--	---

数据库连接字段：

setupwizard/ configuration tool	dbconfig.xml	描述
<b>Hostname</b>	位于<url>标记（例如下面的粗体文字）： <url>jdbc:mysql:// <b>dbserver:3306</b> /jiradb?useUnicode=true&characterEncoding=UTF8&sessionVariables=storage_engine=InnoDB</url>	安装 MySQL 服务器的机器名称或 IP 地址。
<b>Port</b>	位于<url>标记（例如下面的粗体文字）： <url>jdbc:mysql://dbserver: <b>3306</b> /jiradb?useUnicode=true&characterEncoding=UTF8&sessionVariables=storage_engine=InnoDB</url>	MySQL 服务器正在监听的 TCP/IP 端口。也可以留空来使用默认端口。
<b>Database</b>	位于<url>标记（例如下面的粗体文字）： <url>jdbc:mysql://dbserver:3306/ <b>jiradb</b> ?useUnicode=true&characterEncoding=UTF8&sessionVariables=storage_engine=InnoDB</url>	MySQL 数据库（JIRA 数据将保存的数据库）的名称。应该在上面第一步时创建。
<b>Username</b>	位于<username>标记（例如下面的粗体字）： <username> <b>jiradbuser</b> </username>	JIRA 用于连接 MySQL 服务器的用户。应在上面第一步时创建。
<b>Password</b>	位于<password>标记（例如下面的粗体字）： <password> <b>jiradbuser</b> </password>	用于与 MySQL 服务器身份验证的用户密码。

dbconfig.xml 文件示例

```
<?xml version="1.0" encoding="UTF-8"?>

<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>mysql</database-type>
  <jdbc-datasource>
    <url>jdbc:mysql://dbserver:3306/jiradb?useUnicode=true&characterEncoding=UTF8&sessionVariables=storage_engine=InnoDB</url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <validation-query>select 1</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between- eviction-runs-millis>300000</time-between- eviction-runs-millis>

    <pool-test-on-borrow>false</pool-test-on-borrow>
    <pool-test-while-idle>true</pool-test-while-idle>
    <validation-query-timeout>3</validation-query-timeout>
  </jdbc-datasource>
</jira-database-config>
```

## 4.启动 JIRA

现在 JIRA 连接到 MySQL 数据库就配置好了。下一步就是启动 JIRA。

- 如果正在使用 JIRA 的'recommended'分配，启动它，并观察日志中是否存在错误。
- 如果使用的是 JIRA WAR 分配，重建并重新部署在应用服务器的 Web 应用程序。

### 已知的问题和解决方案：

- **权限中的主机名称相当于字符串**-如果在 MySQL 中授权给主机名为 localhost，那么从 JIRA 连接数据库时必须使用相同的字符串。因此使用 127.0.0.1 就不能工作，尽管它们指向同一个地址。这将产生一个找不到表的错误，这是因为当启动 JIRA 时 JDBC 连接没有创建表的权限。
- **连接关闭**-如果使用下面任何一个来使用 MySQL 数据库，可能会遇到连接关闭的问题(详见 JIRA-15731)。阅读 [Surviving Connection Closures](#) 来解决问题。
  - JIRA 3.13 or above,
  - version 5.5.25 or higher of Tomcat 5,
  - version 6.0.13 or higher of Tomcat 6,
- **数据库密码特殊字符**-JIRA 不能释义数据库密码中的特殊字符。
- **使用 InnoDB 存储引擎**- 先前 MySQL Server 版本到 5.5 使用的是默认存储引擎 MyISAM。因此 JIRA 数据库在默认配置的 MySQL Server 5.5 以前的版本运行时，可能导致 JIRA 中的数据损坏。强烈建议在数据库指定 sessionVariables=storage\_engine=InnoDB 参数。这样做确保写入到 JIRA 的 MySQL 数据库的表将使用 InnoDB 存储引擎，它支持 JIRA 需要的'database transactions'。

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- **二进制日志**-注意 JIRA 的 MySQL 使用'READ-COMMITTED'事务隔离级别，目前只支持基于行的二进制日志。若需要 MySQL 的二进制日志功能，则必须配置 MySQL 的二进制日志格式为'row-based'。否则，在 JIRA 中创建问题时可能会出错。参见[当使用 MySQL 的二进制日志时不能创建问题](#)进行解决。
- **4 字节字符**-注意 JIRA 不支持 MySQL 使用 4 字节字符。

## 连接 Oracle 数据库

### 首先

- 检查 Oracle 的版本是否支持，详见[支持的平台](#)。
- 如果是转移 JIRA 到另一台服务器，先导出数据为 XML 备份，然后将旧数据库中的数据传输到新的数据库。
- 在开始前关闭 JIRA，除非你正在运行 Setup Wizard。

### 1.配置 Oracle

1. 确保有适合 JIRA 的数据库实例（新建或使用已存在的）。
2. 在这个数据库实例里创建一个将与 JIRA 连接的用户，如：**jiradbuser**。  
`create user <user> identified by <user_pass> default  
tablespace<tablespace_name> quota unlimited on <tablespace_name>;`  
注意：
  - 。当你在 Oracle 中创建一个用户时，Oracle 会自动创建一个'schema'。
  - 。当创建用户时，表对象的空间必须被指定。
  - 。当创建用户时，只能用 Oracle 支持的不带引号的字符。至少已有一个已知的使用不支持的字符时出现的问题（[Cannot Create XML Backup Due to "Could not get unique fields for table"](#)）。
3. 确保创建的用户有下列权限：
  - 。grant connect to <user>;
  - 。grant create table to <user>;
  - 。grant create sequence to <user>;
  - 。grant create trigger to <user>;

⚠如果权限分配不正确，JIRA 实例可能不会正常工作，参考[JIRA XML Backup and Restore fails with error: Could not find column <column name> in previously parsed query](#) 里的描述。所以只分配上面提到的权限。
4. 确保数据库与 JIRA 配置为使用相同的字符编码，推荐使用 AL32UTF8（Oracle 等效的为 Unicode UTF-8）。

### 2.将 Oracle JDBC 驱动程序复制到应用服务器（仅 JIRA WAR）

⚠️ 如果安装的 JIRA 'Recommended' 分布可跳过此步，其中包含 Oracle JDBC 驱动。而 JIRA WAR 分布不包含该驱动。

1. 下载 Oracle JDBC 驱动（从 [Oracle 网站](#)）。
2. 添加适当的 Oracle JDBC 驱动 jar（**ojdbc6.jar** for JDK 1.6）到 lib/ 目录。  
 ⓘ 注意一些 Oracle JDBC 驱动版本不能与 JIRA 一起使用或内在不稳定。已知的 Oracle 驱动有关的问题如下：

。建议使用 11.2.x 版本的 Oracle 驱动（是向后兼容的）。许多其他版本的驱动已出现以下问题：

- 。10.2.0.3.0 版本发布的 2JDBC 驱动产生 ORA-01461 的错误，详见 [Oracle Support](#) 网站。
- 。10.2.0.1.0 版本发布的 2JDBC 驱动出现有一些数据库挂断的问题。
- 。10.1.0.4 版本发布的 1JDBC 驱动虽不挂断，但会抛出 `ArrayIndexOutOfBoundsException` 异常。
- 。注意：JDK 1.6 不支持 JIRA 6.0 及以后的版本。但可以使用 Oracle JDBC 驱动 JDK 1.7 替代 JDK 1.6。

### 3. 配置 JIRA 服务器来连接 Oracle 数据库

三种连接 Oracle 数据库的方法：

- **使用 JIRA 安装向导**（不适用于 JIRA WAR）-如果你已经安装 JIRA 并且是第一次设置，你的设置将被保存到 JIRA 主目录的 `dbconfig.xml` 文件里。
- **使用 JIRA 配置工具**（不适用于 JIRA WAR）-如果有已存在的 JIRA 实例，你的设置将被保存到 JIRA 主目录的 `dbconfig.xml` 文件里。
- **手动配置**-只有当你有 JIRA WAR 实例或控制台时使用这种方法链接到 JIRA 服务器。你需要手动更新 JIRA 主目录里的 `dbconfig.xml` 文件。

每种配置方法的说明

<b>JIRA setup wizard</b>	<p>当首次在浏览器连接 JIRA 时，JIRA 设置向导就会出现。</p> <ol style="list-style-type: none"> <li>1. 在最开始页面的 'Configure Language and Database' 设置 <b>Database Connection</b> 为 <b>My own database</b>。</li> <li>2. 设置 <b>Database Type</b> 为 <b>Oracle</b>。</li> <li>3. 按照下面描述的数据库连接字段填写字段。</li> <li>4. 测试连接并保存。</li> </ol>
<b>JIRA configuration tool</b>	<ol style="list-style-type: none"> <li>1. 根据下面运行 JIRA 配置工具：  <b>Windows:</b> 打开命令提示符，运行 JIRA 安装目录的 <code>bin</code> 子目录文件 <code>config.bat</code>。  <b>Linux/Unix:</b> 打开控制台，运行 JIRA 安装目录的 <code>bin</code> 子目录文件 <code>config.sh</code>。            可能会出现失败的错误，参照此 <a href="#">文章</a> 的解决方法。</li> <li>2. 导航到 <b>Database</b> 选项卡设置 <b>Database type</b> 为 <b>Oracle</b>。</li> <li>3. 按照下面描述的数据库连接字段填写字段。</li> <li>4. 测试连接并保存。当手动在 JIRA 配置 Oracle 时任何自</li> </ol>

	<p>定义设置（如 adding the &lt;connection-properties&gt;SetBigStringTryClob=true&lt;/connection-properties&gt;）都会被删掉，因此需要手动恢复它们。</p> <p>5. 重启 JIRA。</p>
Manually	<ol style="list-style-type: none"> <li>找到 JIRA 主目录根目录下的 dbconfig.xml 文件。如果该文件不存在，创建该文件，将 XML 示例代码（如下）复制粘贴到该文件。</li> <li>根据下面数据库连接字段描述更新该文件。通过在每一个末端添加'amp;'转义所有'&amp;'字符。  <b>i</b>注意： &lt;database-type/&gt;元素必须指定数据库类型，如：oracle10g。如果你忘记指定数据库类型就启动 JIRA，那么数据库表将不会被正确创建。若遇到类似问题，参见<a href="#">指定不正确的数据库类型</a>进行解决。</li> <li>保存文件并重启 JIRA。</li> </ol>

#### 数据库连接字段

Setupwizard/ configuration tool	dbconfig.xml	描述
Hostname	位于<url>标记（例如下面的粗体文字）： <url>jdbc:oracle:thin:@ <b>dbserver</b> :1521:ORCL</url>	安装 Oracle 服务器的机器名称或 IP 地址。
Port	位于<url>标记（例如下面的粗体文字）： <url>jdbc:oracle:thin:@dbserver: <b>1521</b> :ORCL</url>	MySQL 服务器正在监听的 TCP/IP 端口。默认端口号是'1521'。
SID	位于<url>标记（例如下面的粗体文字）： <url>jdbc:oracle:thin:@dbserver:1521: <b>ORCL</b> </url>	Oracle "System Identifier"。大多数 Oracle 服务器默认值是'ORCL'。如果使用的是 Oracle Express Edition，默认的将会是'XE'。
Username	位于<username>标记（例如下面的粗体文字）： <username> <b>jiradbuser</b> </username>	JIRA 用于连接 Oracle 服务器的用户。应在上面第一步时创建。
Password	位于<password>标记（例如下面的粗体文字）： <password> <b>jiradbuser</b> </password>	用于与 Oracle 服务器身份验证的用户密码。

## dbconfig.xml 文件示例

```
<?xml version="1.0" encoding="UTF-8"?>

<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>oracle10g</database-type>
  <jdbc-datasource>
    <url>jdbc:oracle:thin:@dbserver:1521:ORCL</url>
    <driver-class>oracle.jdbc.OracleDriver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <validation-query>select 1 from dual</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

    <pool-test-while-idle>true</pool-test-while-idle>
    <pool-test-on-borrow>false</pool-test-on-borrow>

  </jdbc-datasource>
</jira-database-config>
```

## 4. 启动 JIRA

现在 JIRA 连接到 Oracle 数据库就配置好了。下一步就是启动 JIRA。

- 如果正在使用 JIRA 的'recommended'分配，启动它，并观察日志中是否存在错误。
- 如果使用的是 JIRA WAR 分配，重建并重新部署在应用服务器的 Web 应用程序。

### 已知的问题和解决方案：

- 当处理自定义 workflow 或拥有长描述、评论、或自定义字段值的问题时，如果遇到这个问题，试着在 dbconfig.xml 文件里添加 `<connection-properties>SetBigStringTryClob=true</connection-properties>` 作为 `</jdbc-datasource>` 元素的子元素。添加此连接属性就能解决这一问题。注意你需要重启来使这一设置产生作用。

## 连接 SQL Server 数据库

### 首先

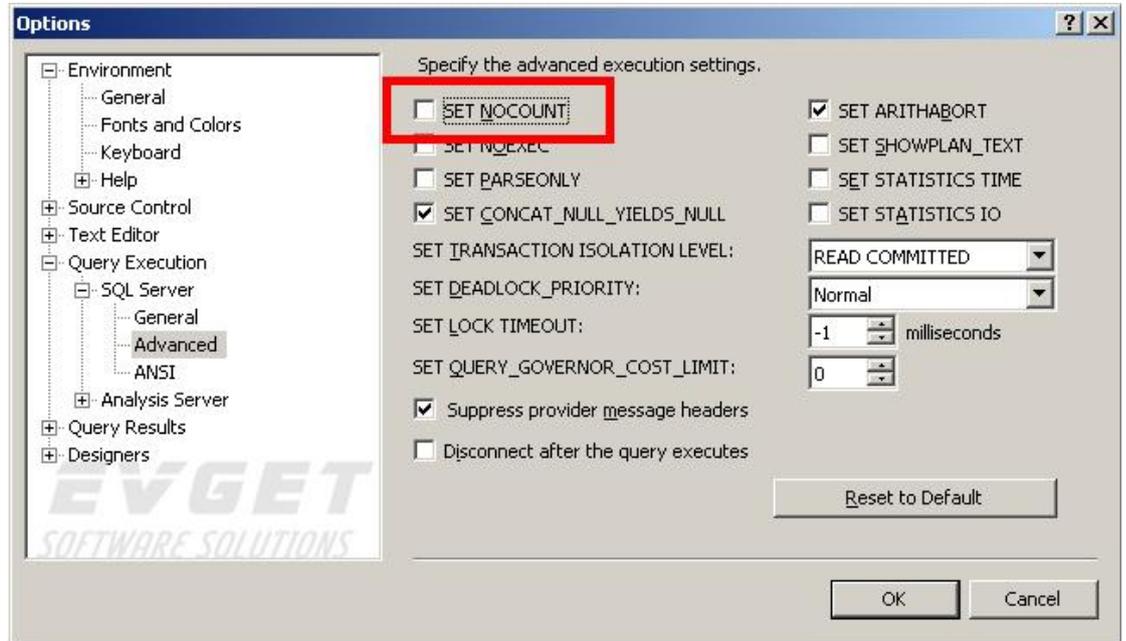
- 查看 SQL Server 的版本是否支持，查阅[支持的平台](#)。  
注意 SQL Server Express 是不支持的，但可以设置 JIRA 来使之与该数据库一起工作。正因为它不支持，因此没有该配置的文档。
- 如果是转移 JIRA 到另一台服务器，先导出数据为 XML 备份，然后将旧数据库中的数据传输到新的数据库。
- 在开始前关闭 JIRA，除非你正在运行 Setup Wizard。

### 1. 创建并配置 SQL Server 数据库

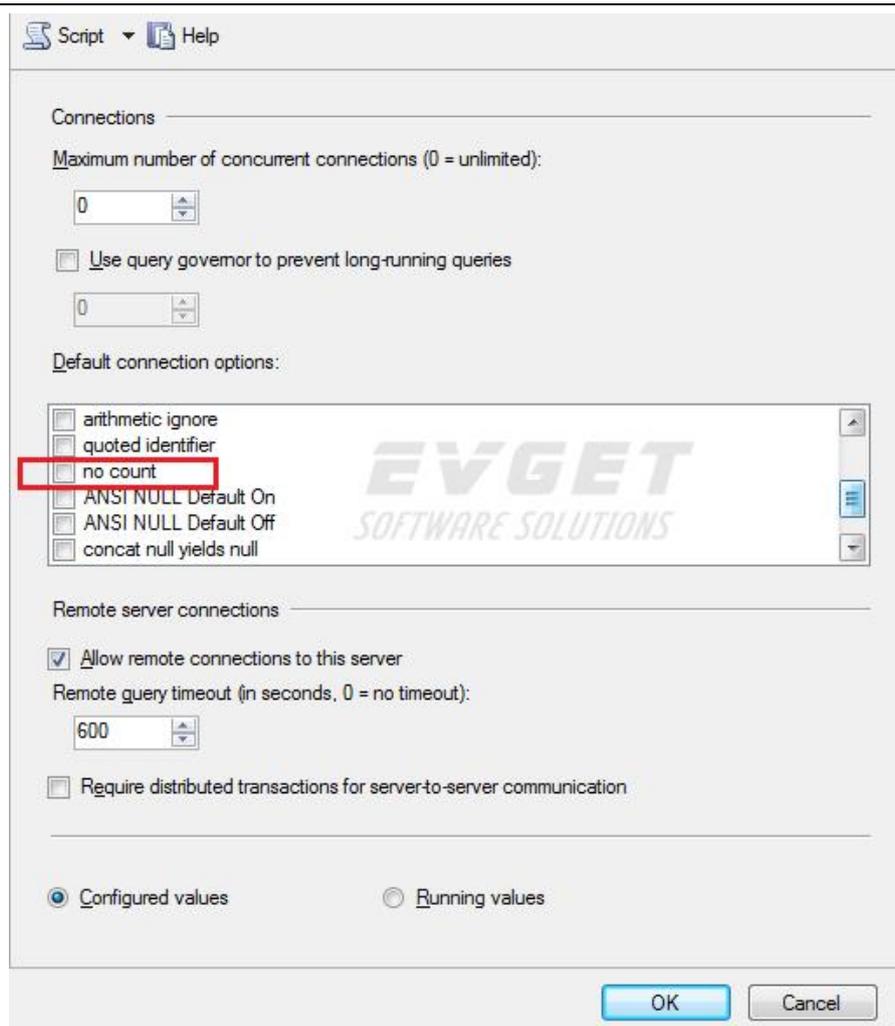
1. 创建一个 JIRA 用于存储问题的数据库，如：**jiradb**。
  - 。排序规则类型必须是 case-insensitive, accent-insensitive 以及 language neutral，例如：'SQL\_Latin1\_General\_CP437\_CI\_AI'。如果你的 SQL Server 安装的默认排序规则没有被修改，检查排序规则类型设置进行修改。
  - 。SQL Server 使用 Unicode 编码存储字符，这足以防止任何可能的问题。
2. 创建一个与 JIRA 连接的数据库用户，如：**jiradbuser**。
  - 。这个数据库用户不能是该数据库的拥有者，但应该在 db\_owner 角色中。
3. 在数据库中为 JIRA 表创建一个空的'schema'，如：**jiraschema**。
  - 。在 SQL Server 中的'schema'是用来包含对象的命名空间，并且不同于传统的数据库模式。你不需要创建任何 JIRA 表、字段或关系（JIRA 会在首次启动这个空模式时创建这些对象）。可参考 [Microsoft documentation](#) 查看更多 SQL Server 模式信息。
4. 确保该用户在新创建的模式中有连接数据库以及创建和写入表的权限。
  - 。如果在 JIRA 设置 SQL Server 数据库还有问题可参考 [Setting Up a SQL Server database for JIRA](#)。
5. 确保 TCP/IP 可用于 SQL Server 并且正在监听正确的端口（SQL Server 安装默认端口为 1433）。
  - 。阅读 [Microsoft documentation](#) 查看更多如何启用网络协议（TCP/IP）以及如何配置 SQL Server 来监听特定的端口。
6. 确保 SQL Server 在适当的身份验证模式下运行。
  - 。默认情况下，SQL Server 在'Windows Authentication Mode'下运行。但如果你的用户没有与受信任的 SQL 连接关联，如'Microsoft SQL Server, Error: 18452'就是在启动 JIRA 时出现的错误，这时你需要修改身份验证模式为'Mixed Authentication Mode'。阅读 [Microsoft documentation](#) 了解身份验证模式并修改身份验证模式为'Mixed Authentication Mode'。

7. 取消 SET NOCOUNT 选项(查看[由 SET NOCOUNT 引起的 MS SQL Server 错误](#))，方法如下：

。打开 **SQL Server Management Studio** 并导航到 **Tools > Options > Query Execution > SQL Server > Advanced**。下面的截屏显示的就是 MSSQL Server 中此设置的配置面板。确保 SET NOCOUNT 选项不被选上：



\*当连接 SQL Server 2008 时还需要在 **Server > Properties > Connections > Default Connections** 属性框中清除 **no count** 选项。



。通过选中新建的数据库点击右键连接 **Query Console** 并选择'New Query'。执行以下命令来设置隔离级别。

```
ALTER DATABASE THE-NEW-DATABASE-CREATED-FOR-JIRA SET
READ_COMMITTED_SNAPSHOT ON
```

## 2.将 SQL Server JDBC 驱动程序复制到应用服务器（仅 JIRA WAR）

**⚠** 如果你安装了包含 SQL Server JDBC 驱动的 JIRA'Recommended'分布可跳过此步。JIRA WAR 分布不包含该驱动。

1. 从 JTDS 下载 SQL Server JDBC 驱动(v1.2.4)。
  - i** 微软有他们自己的 JDBC 驱动，但在收到多次的间歇性断开(JRA-5760 and JRA-6872)， workflow问题(JRA-8443)以及汉语字符问题(JRA-5054)报告后，强烈建议不要使用这个。
2. 将 SQL Server JDBC 驱动 jar (jtds-1.2.4.jar)添加到<Tomcat install>/lib/目录。

## 3.配置 JIRA 服务器来连接 SQL Server 数据库

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

三中配置 JIRA 服务器来连接 SQL Server 数据库的方法：

- **使用 JIRA 安装向导**（不适用于 JIRA WAR）-如果你已经安装 JIRA 并且是第一次设置，你的设置将被保存到 JIRA 主目录的 dbconfig.xml 文件里。
- **使用 JIRA 配置工具**（不适用于 JIRA WAR）-如果有已存在的 JIRA 实例，你的设置将被保存到 JIRA 主目录的 dbconfig.xml 文件里。
- **手动配置**-只有当你有 JIRA WAR 实例或控制台时使用这种方法链接到 JIRA 服务器。你需要手动更新 JIRA 主目录里的 dbconfig.xml 文件。

每种配置方法的说明

<b>JIRA setup wizard</b>	<p>当你首次在浏览器上访问 JIRA 时，JIRA setup wizard 才会显示。</p> <ol style="list-style-type: none"> <li>1. 在最开始页面的'Configure Language and Database'设置 <b>Database Connection</b> 为 <b>My own database</b>。</li> <li>2. 设置 <b>Database Type</b> 为 <b>SQL Server</b>。</li> <li>3. 按照下面介绍的数据库连接字段部分描述填写字段。</li> <li>4. 测试连接并保存。</li> </ol>
<b>JIRA configuration tool</b>	<ol style="list-style-type: none"> <li>1. 按如下指示运行 JIRA 配置工具：  <b>Windows:</b>打开命令提示符，运行 JIRA 安装目录的 bin 子目录文件 config.bat。  <b>Linux/Unix:</b>打开控制台，运行 JIRA 安装目录的 bin 子目录文件 config.sh。  <span style="color: blue;">i</span>可能会出现失败的错误，参照此<a href="#">文章</a>的解决方法。</li> <li>2. 导航到 <b>Database</b> 选项卡，设置 <b>Database type</b> 为 <b>SQL Server</b>。</li> <li>3. 按下面描述的数据库连接字段填写字段。</li> <li>4. 测试连接并保存。</li> <li>5. 重启 JIRA。</li> </ol>
<b>Manually</b>	<ol style="list-style-type: none"> <li>1. 找到 JIRA 主目录根目录下的 dbconfig.xml 文件。如果该文件不存在，创建该文件，将 XML 示例（如下）复制粘贴到该文件。</li> <li>2. 根据下面数据库连接字段描述更新该文件。通过在每一个末端添加'&amp;'转义所有'&amp;'字符。  <span style="color: blue;">i</span>注意： &lt;database-type/&gt;元素必须指定数据库类型，如： mssql。如果你忘记指定数据库类型就启动 JIRA，那么数据库表将不会被正确创建。若遇到类似问题，参见<a href="#">指定不正确的数据库类型</a>进行解决。</li> <li>3. 保存文件并重启 JIRA。</li> </ol>

## 数据库连接字段

Setupwizard/configuration tool	dbconfig.xml	描述
<b>Hostname</b>	位于<url>标记（例如下面的粗体文字）： <url>jdbc:jtds:sqlserver:// <b>dbserver:1433</b> /jiradb</url>	安装 SQL Server 服务器的机器名称或 IP 地址。
<b>Port</b>	位于<url>标记（例如下面的粗体文字）： <url>jdbc:jtds:sqlserver://dbserver: <b>1433</b> /jiradb</url>	SQL Server 服务器正在监听的 TCP/IP 端口。也可留空使用默认的端口。
<b>Database</b>	位于<url>标记（例如下面的粗体文字）： <url>jdbc:jtds:sqlserver://dbserver:1433/ <b>jiradb</b> </url>	SQL Server 数据库（用于存储 JIRA 数据）的名称。
<b>Username</b>	位于<username>标记（例如下面的粗体字）： <username> <b>jiradbuser</b> </username>	JIRA 用于连接 SQL Server 服务器的用户。
<b>Password</b>	位于<password>标记（例如下面的粗体字）： <password> <b>jiradbuser</b> </password>	用于与 SQL Server 服务器身份验证的用户密码。
<b>Schema</b>	位于<schema-name>标记（例如下面的粗体字）： <schema-name> <b>dbo</b> </schema-name>	SQL Server 数据库使用的模式的名称。

dbconfig.xml 文件示例

```
<?xml version="1.0" encoding="UTF-8"?>

<jira-database-config>
  <name>defaultDS</name>
  <delegator-name>default</delegator-name>
  <database-type>mssql</database-type>
  <schema-name>dbo</schema-name>
  <jdbc-datasource>
    <url>jdbc:jtds:sqlserver://dbserver:1433/jiradb</url>
    <driver-class>net.sourceforge.jtds.jdbc.Driver</driver-class>
    <username>jiradbuser</username>
    <password>password</password>
    <pool-min-size>20</pool-min-size>
    <pool-max-size>20</pool-max-size>
    <pool-max-wait>30000</pool-max-wait>
    <pool-max-idle>20</pool-max-idle>
    <pool-remove-abandoned>true</pool-remove-abandoned>
    <pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>

    <validation-query>select 1</validation-query>
    <min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
    <time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>

    <pool-test-while-idle>true</pool-test-while-idle>
  </jdbc-datasource>
</jira-database-config>
```

## 4. 启动 JIRA

现在 JIRA 连接到 SQL Server 数据库就配置好了。下一步就是启动 JIRA。

- 如果正在使用 JIRA 的'recommended'分配，启动它，并观察日志中是否存在错误。
- 如果使用的是 JIRA WAR 分配，重建并重新部署在应用服务器的 Web 应用程序。

## 创建项目

一个 JIRA 项目是所有问题的集合，它可以根据你的需求帮助你的团队协调产品开发，跟踪项目，管理帮助或更多。注意，必须以 JIRA 管理员身份来添加或配置项目。

## 创建新项目

在 JIRA 中创建一个新项目：

1. 点击 Projects > Create project;

---

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

---

## 2. 按照向导来创建项目。

### 关于项目的类型：

- **Simple Issue Tracking:** 该项目为你提供了一种快速简便的方法来启动和运行 JIRA 进行简单的问题跟踪，详见创建简单问题跟踪项目。
- **Project Management:** 这种项目类型创建一个默认的 JIRA 项目。
- **Agile Software Development-Scrum:** 该项目为你提供了 Scrum 的软件开发模板，包括 Scrum 主板以及特定敏捷的问题类型，如 Story 和 Epic。
- **Agile Software Development-Kanban:** 该项目为你提供看板软件开发模板，包括看板以及特定敏捷的问题类型，如 Story 和 Epic。
- **Software Development:** 该项目提供了一个用于软件开发的模板，详见创建软件开发项目。
- **JIRA Default Schemes:** 这是使用共享 JIRA 默认方案创建项目(默认 JIRA 项目)的唯一项目类型。如果你使用任何其他项目类型创建一个项目，那么该项目将使用具体的项目方案(如问题类方案，工作流程方案，屏幕方案和问题类屏幕方案)。

### 关于项目的详细信息：

- 项目重点将作为该项目关键问题的前缀(如“TEST-100”)，选择一个描述性且易于输入的。
- 项目负责人是唯一的。选择负责管理该项目的项目负责人。如果 JIRA 系统中只有一个用户，那该用户为默认的项目负责人，但这个领域将不可用。

## 配置已有项目

### 在 JIRA 中配置项目：

1. 导航到项目的管理页面：
  - 。选择 **Projects** 或
  - 。导航到所需项目的汇总并点击项目导航栏底部的 **Project Administration** 按钮。
2. 使用左侧的选项卡并在不同项目设置之间进行切换。阅读以下部分，了解每种设置的说明。

[Project details](#) | [Issue types](#) | [Workflows](#) | [Screens](#) | [Fields](#) | [Settings](#) | [Roles](#) | [Versions](#) | [Components](#) | [Permissions](#) | [Notifications](#) | [Development tools](#)

## Project details

点击 **Project Summary** 页面顶部的 **Edit Project**，编辑想要的项目描述。注意以下几点：

- **编辑项目关键：**这不是一个简单的任务。编辑项目键之前阅读此页：编辑项目的键。
- **项目描述使用 HTML：**可以包括 HTML，但要确保所有的标签都关闭。注意，这是完全未经过滤的 HTML，因此，很容易受到跨站点脚本攻击。
- **选择一个项目头像：**如果你不想使用项目头像，你可以上传一个透明的像素。

---

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

## 关于项目类别：

项目类别不是在 Edit Project 对话框里编辑的。相反，点击项目管理员界面 **Category**（项目名称下面）旁边的链接。类别可以通过 **Administration > Projects > Project Categories** 创建或查看。

为什么类别有用？ JIRA 可以搜索特定项目类别中的所有问题（如高级搜索“building”类），并能按项目类别分类显示项目。一个 JIRA 项目只能属于一个类别。注意，项目类别不是项目层次结构的一部分。此外，JIRA 不支持子项目或父项目。

## Issue types

JIRA 使你通过使用不同类型的问题跟踪不同类型的东西 -如错误、任务等。你可以查看已经指定为你项目的配置了字段和工作流的问题类型。如果你有 JIRA 管理员权限（全局权限），你还可以配置这些问题类型。

点击左侧菜单的 **Issue Types** 或它下面的任一问题类型如：**Bug, Task, Story** 等：

- **Issue Types:** 单击该按钮可查看哪些问题类型适用于这个项目，（即问题类型方案）。也可以查看在该项目中的工作流、字段和屏幕问题类型，但它更容易通过点击问题类型之一来做到这一点。

如果是 JIRA 管理员，单击 **Actions** 菜单，在当前方案编辑问题类型或为你的项目使用不同的方案。

- **任一问题类型 (如 Bug, Task, Story):** 点击这个查看项目里的工作流或屏幕问题类型。

如果你是 JIRA 管理员，你同样可以通过标签来配置工作流和字段问题类型。

## Workflows

JIRA 问题遵循一个过程，来反映团队的工作。工作流定义一系列步骤（或状态）并且问题将会随之而来，如打开、在建、已解决。你可以配置使问题如何在状态间转换，如谁可以转换它们，在什么条件下，以及哪个转换对应哪个屏幕。

- **Workflow Scheme**—项目的工作流程方案确定哪些工作流程（问题状态转换）适用于本次的项目类型。

## Screens

JIRA 可以通过定义屏幕显示在特定时间特定问题的信息。屏幕仅仅是字段的集合。你可以选择当通过工作流中的特定步骤创建、查看、编辑或转换问题时显示哪个屏幕。

- **Screen Scheme**—该项目的屏幕方案决定了屏幕显示不同的问题操作（如查看、编辑、创建）；  
或
- **Issue Type Screen Scheme**—该项目的问题类型屏幕方案决定了哪个屏幕显示不同问题类型的操作（如查看、编辑、创建）。

## Fields

JIRA 允许定义字段：每个字段可以要求或自选、富文本或纯文本、隐藏或可见。通过使用字

---

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

---

段配置来定义这些行为。

- **Field Configuration Scheme**—项目的字段配置方案确定该项目的问题类型需要哪些字段配置。（字段配置将决定每个字段的整体可见性，需求度，格式（维基/富文本或纯文本）以及帮助文本）。

## Settings

- **Application Links (Configure Project Links)**—如果你已将自己的 JIRA 实例链接到其他 Atlassian 的应用程序，如 Confluence, FishEye 或其他 JIRA 实例，那么你能将该 JIRA 项目链接到那些包含你的项目或团队信息的应用领域。例如，Confluence 空间，FishEye 库，JIRA 项目（在另一个 JIRA 实例）等等。这使你可以利用这些应用程序之间的集成点优势。详情请参阅 [Linking to Another Application](#)。

## Roles

不同的人在不同的项目中承担不同的角色—同一个人可能是一个项目的负责人，也可能是另一个项目的遵循者。JIRA 根据你所在的项目分配具体的角色。

- **Project Lead**—用户履行项目负责人的角色。作为“默认受托人”（如下），也有可能在 JIRA 其他地方存在（如权限方案，通知方案，问题保障方案和工作流）。
- **Default Assignee**—在创建项目问题时初始分配的用户。可以是“项目负责人”（如上）或者在 JIRA 的一般配置中 **Allow unassigned issues** 被设置为 'On', 'Unassigned'。也有默认组件受托人。  
默认情况下，新的项目将 'Default Assignee' 设置为 'Unassigned'。你可将其设置为一个特定的角色，如 'Project Lead'。
- **Project Roles**—用户成员或组成员在项目中完成特定的功能。项目角色用于权限方案，通知方案，问题保障方案和工作流。

## Versions

如果使用 JIRA 来管理产品的开发，你可能要定义不同的版本来帮助跟踪哪些问题涉及到不同版本的产品（如 1.0、1.1、1.2、2.0 测试版、2.0）。JIRA 可以帮助你管理，发布和归档版本。版本可以设置发布日期，发布日期已过且尚未发布的将自动突显为 "overdue"。

- **Versions**—在项目中定义版本。详见 [version management](#)。

## Components

你可能需要定义各种部件进行分类和管理不同的问题。例如，对于一个软件开发项目，你可以定义组件“数据库”，“可用性”，“文档”（注意，一个问题可以属于多个组件）。你可以为每一个组件选个默认受托人，在项目中让不同的人负责不同的子团队是很有用的。

- **Components**—项目问题所在的逻辑组。详见 [component management](#)。

## Permissions

通过使用 JIRA 项目权限，你可以控制谁可以访问你的项目，他们可以做什么（例如“在问题作出评论”，“问题上开展工作”，“分配问题”）。你还可以通过使用安全级别来控制访问个别问题。你可以选择允许特定用户或组或角色有访问权限（注意这类角色往往是容易管理的）。

- **Permission Scheme**—项目的权限方案决定谁有权在项目中查看或更改问题。
- **Issue Security Scheme**—项目的问题安全方案决定了在这个项目中的可见性级别。

## Notifications

当项目发生特定事件（如“问题创建”，“问题解决”）时 JIRA 可以通知相关人员。当不同的事件发生，你可以选择特定的人或群体或角色收到通知邮件。（注意此类角色往往是最容易管理的）

- **Notification Scheme**—项目的通知方案决定当项目里的问题发生改变时谁可以收到通知邮件。
- **Email**—仅当 SMTP 邮件服务器已经配置在 JIRA，才可以指定“发件人”地址是该项目。  
⚠ 注意，Default Notification Scheme（JIRA 附带）默认与所有的新项目关联。这意味着如果你有一个发送（SMTP）邮件服务器设置，当新项目中有任何活动（如问题创建）时，通知邮件将立刻发送。

## Development tools

开发工具部分为你提供了连接的开发工具的概述以及哪些用户可以使用下面的集成功能：

- **View permission**—这部分列出了哪些用户可以在问题屏幕上看到的集成功能的开发工具（如创建分支链接），以及其他开发相关的信息，如提交，审查和构建信息。这个功能是通过“View Development Tools”项目权限控制。
- **Applications**—这部分将展示什么开发工具与 JIRA 通过应用链接连接以及和谁有资格使用 JIRA 的开发工具功能。

## 有关项目管理员的注意事项：

JIRA 中的项目管理员是有具体项目的管理项目权限，但不一定是 JIRA 管理员全局权限。尽管没有 JIRA 管理员的全局权限，但项目管理员可以做如下操作：

- 编辑项目名称
- 编辑项目描述
- 编辑项目头像图片
- 编辑项目 URL
- 编辑项目负责人
- 编辑项目角色成员
- 定义项目组成部分
- 定义项目版本
- 查看但不允许选择或编辑项目方案（通知方案、权限方案等）

---

更改 JIRA 项目的类别需要 JIRA 管理员全局权限。

## 编辑项目键

编辑 JIRA 项目的键不是一个简单的任务。当创建项目时，你应该选择一个满足长期需求的键，而不是根据项目创建之后再更改。不过，也有需要更改现有项目键的情况，如更改产品名称。本文主要教大家如何更改项目的键，以及描述这种变化所带来的影响。

### 首先

- 项目键的格式必须是 JIRA 限制规定的格式。默认情况下，项目键格式必须至少为 2 个字符，并且只包含大写字母。  
你可以更改项目键的格式来实现不同的限制。
- 在低使用率期间执行此更改-当保存更新项目键时 JIRA 将启动后台重新索引。这可能会对实例的性能产生影响。注意，不能选择“Lock JIRA and rebuild index”，尽管它会使后台索引变快，但是将受限项目的问题。
- 将更改传达给用户 - 确保你知道更改项目键的后果，且用户已做好充分准备。

## 编辑项目键

1. 导航到 JIRA 项目，并访问项目管理页面。选择 **Project Administration**（项目导航栏的底部）。
2. 选择 **Edit Project** 按钮。
3. 选择 **Key** 字段旁边的 **edit key**。
4. 更改键并点击 **Update**。

### Edit Project: Sample Scrum Project

Name \*

Key \*  [edit key](#)

URL

Project Avatar \*

### 注意：

- 页面上的其它字段，保存后可以直接看到更改后的信息。而不需要等待重新索引完成。
- 不要取消后台重新索引。如果取消，那么搜索与项目有关的问题时将出错。如果需要取消，你可以稍后再取消。

## 更新任务

- **Fix the project entity links**-当你把 JIRA 和其它 Atlassian 应用程序链接时，JIRA 项目和其他应用程序相关的项目（如 Confluence spaces）之间的实体链接会被自动创建。如果 JIRA 项目的键更改了，那你需要修改项目的实体链接。
- **Updating JIRA Agile board filters**-如果 JIRA 敏捷主板使用旧项目键，过滤器板需要更新到新的项目键。否则，版上可能不会显示更名后的项目问题。

## 变更管理注意事项：

编辑项目的键是一个大的更改，多数情况下，期望 JIRA 项目以新键运行。注意以下少数情况。

- 旧的项目键可用于 JQL 查询 - 用户将不必更新引用旧项目键的问题过滤器。
- 如果将 Confluence 和 JIRA 一起使用，Confluence 中的 JIRA 问题宏将继续有效。注意，如果你没有马上看到更改，是因为缓存更新需要时间。
- 你不能使用旧的项目键来创建一个新的项目。但你可以将重命名的项目改回旧的项目键。如果删除该项目，所有相关的键将被释放，你就可以重新使用它们。
- 无论是来自 JIRA 内部或是外部的链接都将正常工作。然而，链接别名不会被更新 - 例如，如果有一个链接到名为“EXAMPLE-1”的链接，将该项目的键“EXAMPLE”改为“DEMO”后，别名 'EXAMPLE-1' 不会更新为 'DEMO-1'，但该链接还是会链接到 DEMO-1。
- 如果你在全局过滤器使用 Activity Stream 小工具，你需要在项目重命名后更新过滤器。
- 当项目的键更改后所有附件也可访问。但注意，它们的存储目录（\data\attachments）将保留旧的项目键。例如，如果把项目的键从 TEST 改为 DEMO 后，附件的存储目录仍是\data\attachments\TEST。
- 如果导出一个重命名的项目再将其导入，将显示更新后的项目键，即不会保留原来的项目键。事实上，该项目的历史键都将被移除。有一种解决方法，就是直接在你的数据库修改数据。

## 相关话题

**Changing the maximum project key length**-你可以修改项目键的最大字符长度。导航到 JIRA 管理控制台的 General Configuration 页面修改 **Maximum project key size** 字段。

---

**Changing the project key format**–你可以修改项目键的格式，但当创建或编辑项目键时，这将限制其格式。

### 开发人员注意事项：

- 在项目键更改后 REST API 调用指定关键问题时仍用旧的项目键。例如，项目键从 EXAMPLE 改为 DEMO 后，/rest/api/issue/EXAMPLE-100 仍可用。
- ProjectUpdatedEvent 事件被创建，任何时候项目的细节包括项目的键发生改变该事件将被触发。
- 要想检索所有问题键和项目键，你可以通过以下方式：

#### REST:

- 获取所有项目键: /rest/api/2/project/?expand=projectKeys

#### Java API:

- 获取所有项目键: com.atlassian.jira.project.ProjectManager#getAllProjectKeys
- 获取所有问题键: com.atlassian.jira.issue.IssueManager#getAllIssueKeys

## 创建软件开发项目

[JIRA](#) 为软件开发项目提供了一个模板。包括简单工作流程以及开发评审过程中的各种任务状态。你也可以自定义模板来满足团队不断发展的过程。

### 创建软件开发项目

根据以下步骤来创建软件开发项目，确保在第三步时选择 **Software Development**。

1. 以 JIRA 全局权限身份登录。
2. 选择 **Projects > Create Project**。
3. 选择想要创建的项目类型。
4. 点击进入 **project details**。注意谨慎选择项目键，因为在项目创建后不容易更改。
5. 点击 **Submit** 提交，然后创建下一个。

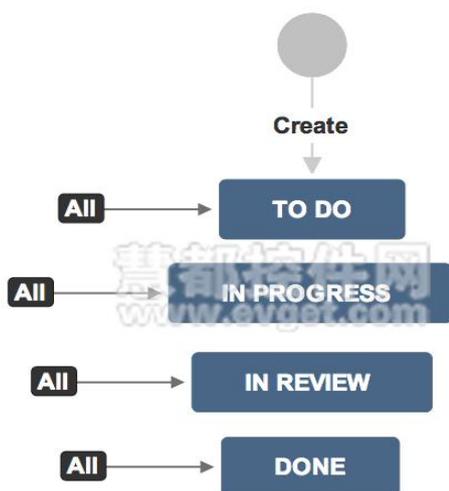
### 模板详细信息

#### 问题类型

- 父类型: **Story**、**Task**、**New Feature** 和 **Bug**
- 子任务类型: **Sub-Task**

JIRA 自动创建一个名为<项目键>: Software Development issue type scheme 的方案。

软件开发项目工作流程图：



### 流程描述

- 四个状态：**To Do**、**In Progress**、**In Review** 和 **Done**。
- 直到状态 **Done** 项目结束。
- 你可以从任何一个状态转换到关联的状态。

JIRA 自动创建名为<项目键>: Software Development workflow scheme 的方案。

### 创建简单问题跟踪项目

[JIRA](#) 项目提供了一个快速简便的方法来启动 JIRA 进行简单的问题跟踪。旨在利用 JIRA 的其它强大功能的优势为用户提供简单的三步工作流程来进行问题跟踪。同时也很容易延伸至 JIRA 提供的其它更复杂的工作流程。

### 创建简单问题跟踪

根据下面的步骤来创建简单问题跟踪，确保第三步时选择 **Simple Issue Tracking**。

1. 以 JIRA 全局权限管理员身份登录。

2. 选择 **Projects > Create Project**。
3. 选择你想要创建的 **type of project**。
4. 点击进入 **project details**。注意谨慎选择项目键，因为在项目创建后不容易更改。
5. 点击 **Submit** 提交，然后创建下一个。

## 模板详细信息

### 问题类型

- 父类型：**New Feature**、**Task**
- 子任务类型：**Sub-Task**

JIRA 自动创建一个名为<项目键>: **Simple Issue Tracking issue type scheme** 的方案。

简单的问题项目工作流程

项目工作流程图如下：



## 流程描述

- 三个状态：**To Do**、**In Progress** 和 **Done**
- 直到状态 **Done** 项目结束

JIRA 自动创建一个名为<项目键>: **Simple Issue Tracking workflow scheme** 的方案。

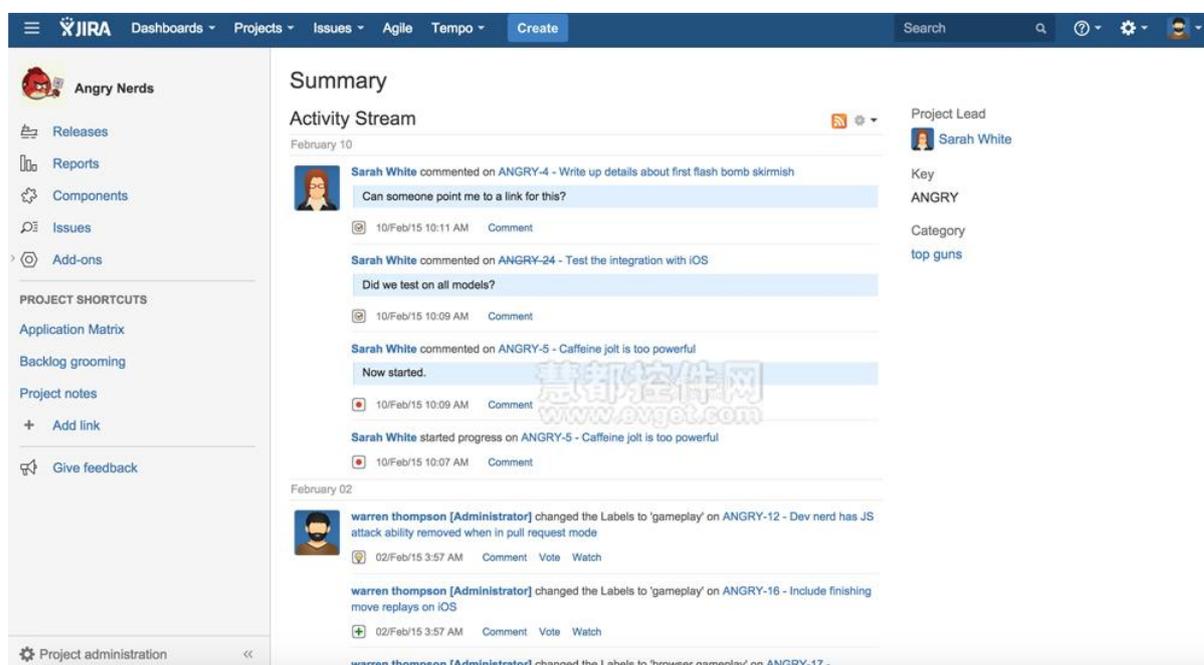
## 流程转换

转换具有以下特性：

- 当前用户指定'Start Progress'转换完成。
- 当前用户指定'Reopen and Start Progress'转换完成。
- 当任何'Done'执行转换完成。
- 当'Reopen'字段被清除时转换完成。
- 没有转换页面。
- 没有转换受理条件（除了整个方案的权限）。

## 查看项目

当你在 [JIRA](#) 中查看一个项目时，首先呈现的是项目概览页面。该页面呈现出项目的所有动态概览，并且可以根据导航查看详细信息。你也可以添加快捷方式到信息中心和其他网页，以便所有用户访问该项目。



点击 **Projects** 下拉菜单选择你的项目进入项目概览页面。如果列表中没有你的项目，选择 **View all projects** 找到你的项目。

在项目概览页面，你可以通过选择项目导航栏的链接查看以下内容：

- **Project:**
  - **Summary** — 显示项目最近的动态、项目负责人、项目中最活跃的用户以及该项目的键。
  - **Releases \*** — 显示项目所有版本的快照，并且可以通过发布与未发布进行筛选。
  - **Reports** — 显示特殊用户、版本、问题或问题在其他领域的统计报告。

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- **Components** \* — 显示项目所有组成部分的摘要。
- **Issues** — 链接到项目的问题列表或详细视图。
- **Add-ons** — 显示 JIRA 附加组件的链接，如日历或测试阶段。如果你集成了 JIRA 的其他产品，还可能看到其他链接。

\* **Releases** 和 **Components** 只有当项目管理员创建项目版本或组件后才可用。

- **Project Shortcuts:**

- 项目的快捷方式可以被添加到项目导航页面以及你的团队可以访问的任何在线资源。这些快捷链接仅供有权访问该项目的用户使用。

## 创建问题

要创建一个 [JIRA](#) 问题，你需要有这一问题相关项目的创建问题权限。如果没有此权限，请联系 JIRA 管理员。

### 创建一个新的 JIRA 问题

1. 点击屏幕上方的 **Create** 打开 **Create Issue** 对话框。
  - ✔ 键盘快捷键：c
2. 在 **Create Issue** 对话框中选择相应的项目以及问题类型。
3. 对问题进行总结，并完成相应的字段 - \*为必填。
  - ❗ 如果要访问未显示在此对话框中的字段或要想隐藏现有字段：
    - 点击屏幕右上方的 **Configure Fields** 按钮。
    - 单击 **Custom**，然后通过复选框来分别隐藏或显示相关字段，或是单击 **All** 显示所有字段。
    - ❗ 当你下次创建问题时，JIRA 会记住你上次选择的字段。
4. 选择：选择对话框底部的 **Create another** 复选框来创建一系列相同项目和类型的相似问题。
5. 当对你的问题内容满意时，单击 **Create** 按钮。

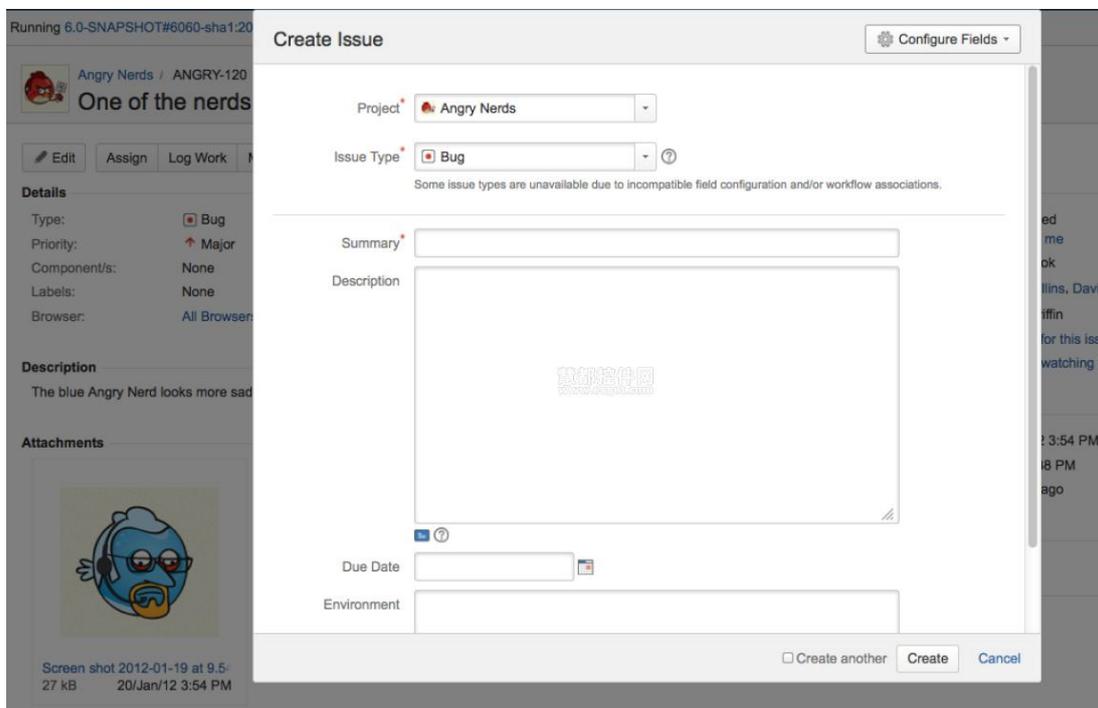
❗ 如果选择了 **Create another** 复选框，将出现一个新的问题创建对话框。某些字段将根据以前的配置预先填充。请务必在创建下一个问题之前确保所有字段都正确。

#### ✔ 温馨提示

- 可以通过 **Description** 或 **Comment** 字段提醒其他用户，点击 **Update** 按钮后邮件将发送到用户邮箱（注册 JIRA 账号时用的）。

- 某些文本字段的问题，可以链接到其他问题、插入宏、插入图片等等。
- 要想查看所有已创建但未解决的问题列表，到你的用户名下点击 **Profile** > **Filters** > **Reported & Open**。
- 你可以自动成为你创建的问题的查看者，这由你的用户配置文件里的 **Autowatch** 设置的。注意，如果没有更改这个设置，你将继承由 JIRA 管理员设置的权限（通过  > **System** > **User Preferences** 设置）。
- 通过 JIRA 管理员适当的配置后，还可以通过邮件来创建问题。
- 如果你正在使用 **Scrum** 敏捷板，你可以通过创建内嵌问题将问题添加到你的待办事项。

创建问题示例：



## 搜索问题

[JIRA](#) 提供了强大的问题搜索工具，使你可以通过一系列搜索条件在整个项目、版本以及组件中搜索问题。 JIRA 也可自定义搜索条件来进一步优化搜索。搜索可以保存在 JIRA 过滤器中，以便再次调用相同条件的搜索，并且可以与其他用户共享。

你可以在 JIRA 搜索中：

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- 修改和重复使用搜索结果
- 排序和操作个别问题
- 共享和输出问题
- 将问题显示在仪表板上

## 第一步:定义搜索条件

搜索问题的第一步是定义新的搜索条件。三种定义搜索条件的方式：**quick search**、**basic search**、**advanced search**。

如果你已经保存了以前的搜索（即作为一个过滤器），并希望再次运行它，你可以运行过滤器来加载保存的搜索条件。

搜索方法	描述
<b>Quick search</b>	最快的定义搜索条件方式。然而，在复杂的查询时相比其他检索方法不太精确（如 <code>project = JIRA AND status = Open AND priority = High</code> ）。快速搜索在导航栏里可见。
<b>Basic search</b>	比快速搜索更精确，并且比高级搜索更容易使用。提供了一个友好的用户界面允许自定义复杂的查询，且无需知道如何使用 JQL。
<b>Advanced search</b>	是三种搜索方式中最强大的。可以定义无法在其他搜索方式里定义的条件（如 <code>ORDER BY</code> ）。但是，你需要知道如何构建使用 JIRA 查询语言（JQL）。

## 关于 filters

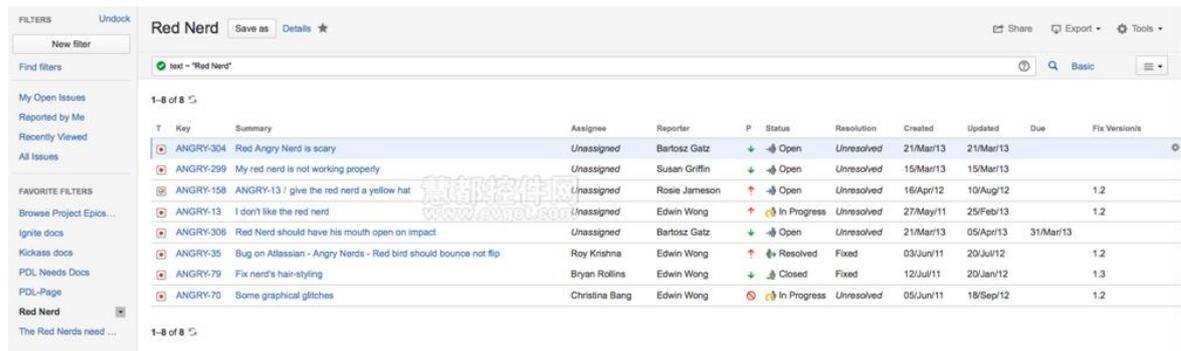
JIRA 过滤器是保存搜索条件的集，类似于书签。一些过滤器是在 JIRA 中预定义的— 如 **My Open Issues**, **Reported by Me**, **Recently Viewed**, **All Issues** — 这些被称为系统过滤器。过滤器也可以是一组由用户自定义和保存的搜索条件。点击 **filter**，你最喜欢的过滤器将示于左侧的面板。你可以在这里查看自己所有的以及共享给你的过滤器。

运行过滤器时，之前保存的搜索条件将加载到问题视图，这些条件既可以显示在基本搜索中又可以显示在高级搜索中，这取决于你所用的搜索方式。

## 第二步:修改搜索结果

一旦定义并运行搜索后，你可以修改搜索结果。 JIRA 允许更改搜索结果的排序顺序，以及对个别问题的进行操作。

问题导航搜索结果示例：



### 第三步:保存搜索

要想以后再次运行此搜索，点击 **Save as** 输入过滤器名字，把搜索存储为一个过滤器。该过滤器被创建并添加到你最喜欢的过滤器。

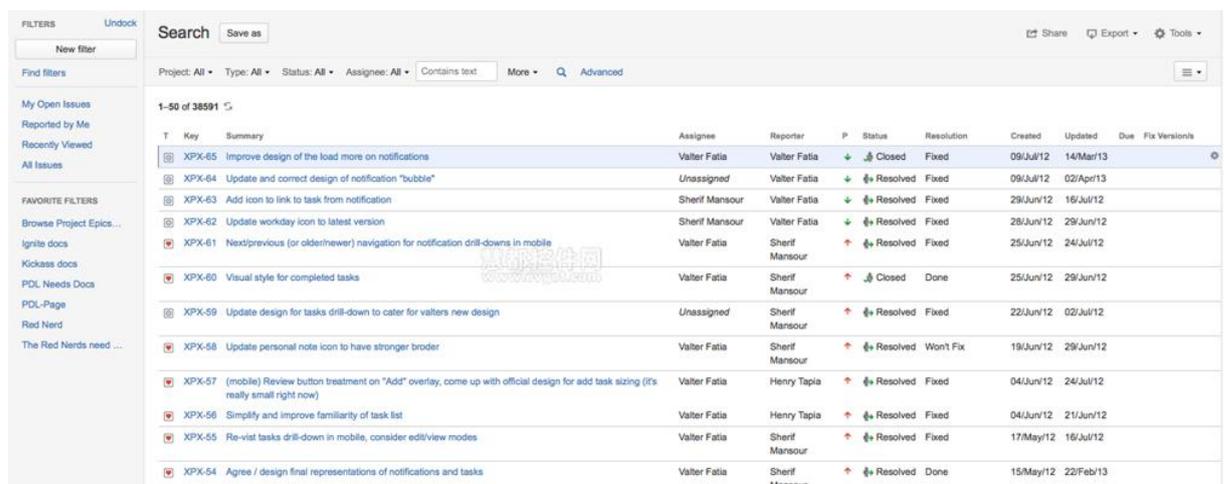
### 第四步:对搜索结果进行操作

搜索结果可以导出为多种格式（如 Microsoft Excel）并且可通过邮件或 RSS [共享搜索结果](#)。同样你也可以将搜索结果作为列表或图表显示在仪表板上。

## 基础搜索

本文主要介绍如何在 [JIRA](#) 中定义和执行基础搜索。你也可以定义和执行使用快速搜索或使用 JQL（高级搜索）。

基础搜索截屏：



JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

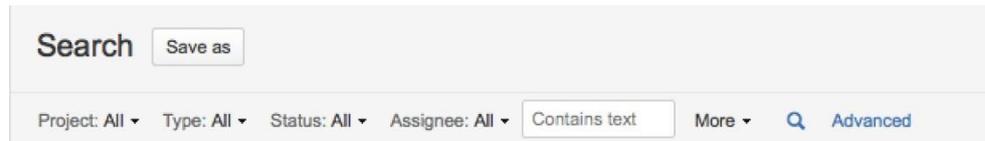
邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

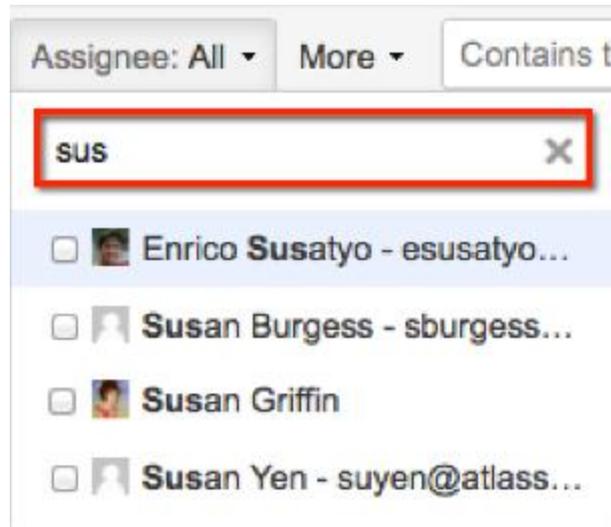
## 执行基础搜索

### 在 JIR 中执行基础搜索

1. 选择 **Issues > Search for Issues** 显示问题导航仪。
  - 如果有现有搜索条件，点击 **New filter** 按钮重置搜索条件。
  - 如果 **Basic** 是以链接形式存在，那么点击 **Basic** 切换到 **Basic** 搜索模型，基础搜索截图如下：



2. 输入搜索条件。你可以搜索特定的字段或搜索文本，如下所述：
  - **按特定的字段搜索**-你需要选择相应的字段然后指定该字段取值。
    - 选择字段作为条件- **Project, Issue Type, Status and Assignee** 字段作为搜索条件一直都可用，如果你需要其它字段作为搜索条件，点击 **More**，勾选你想要的字段。
    - **i** 找不到想要的字段？字段成灰色？见下面的问题部分。
    - 给字段指定值-点击字段下拉菜单选择或输入希望的字段值。如果下拉菜单中没有想要的值，在下面标红的框内输入该字段的值：



如果你最近选择过 **Assignee/Reporter** 字段来查找问题或查看了某个用户的个人资料页面，那么在选择 **Assignee/Reporter** 字段值时，系统会推荐该用户或组。

- **按特定文本搜索**-在 **Contains text** 文本框输入查询条件然后点击 **Enter**。  
**Summary, Description, Comments, Environment** 等所有的文本自定义字段都会被检索。你可以在搜索文本里使用修改器，如通配符和逻辑运算符，详见[执行文本搜索](#)。
- 3. 搜索结果会自动更新，除非管理员关闭了自动更新搜索结果功能。如果是这样的话，在每次更改搜索条件后都要点击字段下拉菜单里的 **Update**。

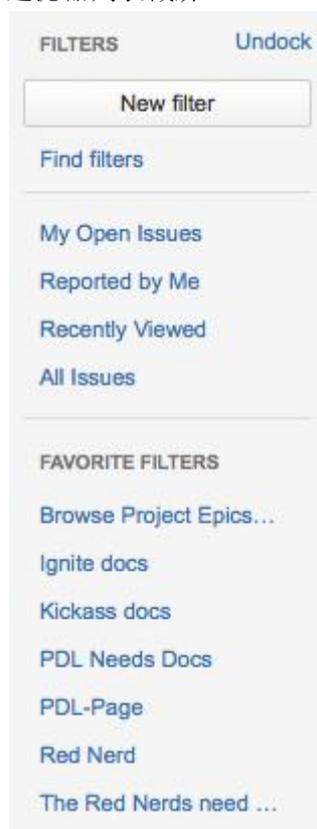
## 保存搜索

将搜索结果存储为过滤器。

**执行保存搜索：**

1. 选择 **Issues > Search for Issues**。
2. 选择左边列表里的任何过滤器：
  - System filter — **My Open Issues, Reported by Me, Recently Viewed, All Issues**
  - Favorite filters (按字母序列出)
  - **Find filters** 可以搜索任何被共享的过滤器，你也可以订阅（添加到你的 **Favorite Filters**）。
3. 选择某个 filter 后，搜索结果和搜索条件也都被显示出来。
  - 如果选择了系统过滤器里的 **Recently Viewed**，JIRA 将切换到高级搜索，因为在过滤器中基础搜索不能呈现 **ORDER BY** 子句。
4. 你可以添加，删除或修改搜索条件来细化搜索结果。你还可以保存修改后的搜索条件（更新自己的过滤器），或者将搜索条件的副本保存为一个新的过滤器。

过滤器列表截屏：



## 可能遇到的问题：

- **为什么找不到想要选择的字段？** 某些字段只适用于特定项目/问题类型。因此你必须选择适用的项目/问题类型，否则不能选择该字段。
- **为什么字段选项呈灰色？** 某些字段只适用于特定项目/问题类型。如果在搜索中选择某字段，然后删除引用该字段的所有项目/问题类型，则该字段是无效的。无效字段不能用于搜索并且显示为灰色文本。

- **为什么会在字段里出现❗️?** 有些字段值仅适用于特定项目/问题类型。例如，你可能已经配置了一个项目，在其工作流中使用 `status` 进行 QA 审核。如果在你的搜索中选择此项目和 `status`，然后更改不使用 QA 审查的项目搜索为过滤器，那么该状态将是无效的并且在搜索中被忽略。
- **为什么搜索结果不自动更新?** 当任何字段改变时搜索结果会自动更新，前提是你的管理员没有禁用搜索结果自动更新的功能。向管理员咨询是否关闭搜索结果自动更新。

## Next steps

- 阅读[使用问题导航](#)查询如何将搜索结果进行分来排序或查看个人的问题。
- 跳转到[对搜索结果数据进行操作](#)，查询如何将搜索结果保存为一个问题过滤器，输出搜索结果并共享。

## 快捷搜索

有时候你只希望去了解你感兴趣的特定 [JIRA](#) 问题，但不记得具体问题是什么，只记得是一个分配给你未决的问题。这种情况下快捷搜索可以帮助你。

### 跳转到一个问题

Quick Search 框在屏幕的右上方。如果输入一个项目的键，将直接跳转到该问题。例如，输入 'ABC-107' (或 'abc-107') 然后单击回车键，将直接跳转到 JIRA 问题 'ABC-107'。



很多情况下，你甚至不需要输入完整的键，只需数字部分即可。如果你当前正在 'ABC' 项目上里操作，输入 '123' 就可以直接跳转到 'ABC-123'。

### 智能查询

快捷搜索可以使你通过最少的输入进行智能查询。例如，要查询 'TEST' 项目里所有的 open bugs，你只需简单输入 'test open bugs'，快捷搜索就可以找到它们。搜索结果将会呈现在问题导航里，你可以通过多种可用格式 (Excel, XML, 等) 来查看。

快速搜索关键词：

关键词	描述	示例
my	查找分配给自己的问题。	my open bugs
r:	通过在 r:前缀后面添加用户来查找问题 注意"r:"和用户之间没有空格。	r:me - 查找自己发布的问题 r:Samuel - 查找 "samuel"用户发布的问题 r:none - 查找没有发布者的问题
项目名称 或项目键	查找具体项目里的问题。	test project TST tst
overdue	查找已经逾期的问题 。	overdue
created: updated: due:	通过使用 created:, updated:和 due:前缀来搜索问题。你可以采用 today, tomorrow, yesterday, 一个日期范围(如 '-1w'), 或两个日期范围 (如'-1w,1w')。注意日期范围里不能有 空格。有效日期 / 时间的缩写: 'w'(week),'d'(day),'h'(hour),'m'(minute)。	created:today created:yesterday updated:-1w - 搜索上一周更新的问题 due:1w - 搜索下一周即将逾期的问题 due:-1d,1w - 搜索从昨天到下一周之间的问题 created:-1w,-30m - 搜索从过去一周到 30 分钟之前创建的问题 created:-1d updated:-4h - 搜索昨天创建并且 4 小时前更新的问题
优先	通过具体的优先查找问题 。	blocker major trivial
问题类型	通过具体的问题类型搜索问题。注意问题类型也可以用复数。	bug task bugs tasks
解决方案	通过具体的解决方案搜索问题。	fixed duplicate cannot reproduce
c:	通过具体组件搜索问题，注意"c:"和组件名称之间无空格。	c:security - 查找组件名

		称里含有"security"的问题
v:	通过版本号搜索问题。使用通配符"*"查找包含该版本号的所有版本问题。 注意"v:"和版本名称之间无空格。	v:3.0 - 搜索匹配该版本的所有问题如： <ul style="list-style-type: none"> <li>• 3.0</li> <li>• 3.0 eap</li> <li>• 3.0 beta</li> </ul> 但不会匹配如下版本，如： <ul style="list-style-type: none"> <li>• 3.0.1</li> <li>• 3.0.0.4</li> </ul>
ff:	查找特定修复版本，使用方法与 v:相同。	
*	通配符'*'可以用于 v:和 ff:。	v:3.2* - 搜索包含有 v:3.2 的所有版本相关问题，如： <ul style="list-style-type: none"> <li>• 3.2</li> <li>• 3.2-beta</li> <li>• 3.2.1</li> <li>• 3.2.x</li> </ul>

## 自由文本搜索

你可以通过输入你想查找的问题里的任何词，该词是包含在下列任一字段中的内容：

- Summary
- Description
- Comments

## 通过浏览器的查询框搜索 JIRA 问题

如果你是用 Firefox 或 IE8 浏览器，你可以通过浏览器搜索框旁边的下拉菜单添加你的 JIRA 网站作为一个搜索引擎。一旦添加了，你可以任何时候在该 JIRA 网站执行快捷搜索。

---

## 高级搜索

### 什么是高级搜索？

高级搜索可以让你使用结构化查询搜索 [JIRA](#) 问题。搜索结果将显示在问题导航器，你可以将其导出到 MS Excel 等多种格式。你还可以保存和订阅你的高级搜索。

当你执行高级搜索时，你是用的是 JIRA 搜索语言（JQL）。JQL 简单查询（也称为一个“语句”）包括一个字段，包括一个运算符、一个或多个值或功能。例如，下面的简单的查询会搜索在“TEST”项目中的所有问题：

**project = "TEST"**

这个例子使用 Project 字段，EQUALS 运算符以及“TEST”值。

**i** JQL 提供一些类似于 SQL 的语法，比如 ORDER BY SQL 关键字和 ISNULL () SQL 函数。然而，JQL 不是数据库查询语言。例如，JQL 不具有 SELECT 语句。

### 如何执行高级搜索

1. 选择 **Issues > Search for Issues** 将出现问题导航界面。
  - 如果有现有的搜索条件，请单击 **New filter** 按钮复位搜索条件。
  - 如果 **Advanced** 是以链接显示，点击该链接切换到高级搜索。
2. 使用字段、运算符、字段值或函数输入查询条件。
3. 点击 **Search** 按钮执行查询。

### 执行文本搜索

当在以下字段执行搜索时，可以使用 Lucene 的文本搜索功能的 CONTAINS 运算符：

- Summary
- Description
- Environment
- Comments
- custom 字段采用“自由文本搜索器”，包括以下内置自定义字段类型：
  - Free Text Field (无限制文本)
  - Text Field (< 255 字节)
  - Read-only Text Field

在 JQL 中使用文本字段的“some words”值搜索问题。这会搜索所有文本自定义字段如总结、描述、环境、评论。如果你有很多文本自定义字段，你可以通过搜索特定的字段来改进搜索，例如：

Summary ~ "some words" OR Description ~ "some words"

---

## 使用自动填写功能

当你在输入查询条件时，JIRA 会自动识别上下文并提供一个自动填写推荐列表。自动填写推荐列表按字母顺序排列，包括前 15 个匹配值。注意自动填写推荐不提供函数参数。

### **i** 注意：

如果自动填写推荐没有出现，可能是因为你的管理员关闭了 JIRA 实例的该功自动填写推荐不会出现在所有字段。点击字段参考查看那些字段支持自动填写。

## 在高级搜索和简单搜索之间切换

通常情况下，使用“简单搜索”创建查询可能会被转换成“高级搜索”（JQL），然后再简单查询查询。

然而，使用“高级搜索”创建查询就不会被转换成“简单搜索”，特别是当：

- 查询包含 OR 运算符（注意你可以使用 IN 运算符，但它将被转换，例如：project in (A, B)）
  - 例：尽管(project = JRA OR project = CONF)查询等同于(project in (JRA, CONF))，但只有第二个查询会被转换
- 查询包含 NOT 运算符
- 查询包含 EMPTY 运算符
- 查询包含任何!=, IS, IS NOT, >, >=, <, <=运算符
- 查询指定相关项目中的字段和值（如版本，组成，自定义字段），并且该项目没有明确包括在查询中（如 fixVersion="4.0"，没有 AND project=JRA）。这些自定义字段特别棘手，因为它们可以在一个项目/问题类型的基础上进行配置。一般的经验规则是，如果该查询不能在“简单搜索”的形式创建，那么如果使用“高级搜索”创造该查询，那么该查询将不会被转换成'简单搜索'。

## 设置运算符优先

你可以通过使用括号来强调 JQL 语句中运算符的优先级。例如，如果你想查找 SysAdmin 项目中所有被解决的问题，以及目前被分配给系统管理员 (bobsmith) 的问题。你可以在你的查询中使用括号强制执行布尔运算符的优先级，即：

```
tatus=resolved AND project=SysAdmin) OR assignee=bobsmith
```

需要注意的是，如果你不使用括号，该语句将使用运算符优先级评估。你还可以使用括号来组合语句，这样就可以将 NOT 运算符应用于组。

**i** 如果你添加括号强制符的优先级，但他们没有严格要求的运算符优先级执行但返回相同的结果，当你保存查询时 JIRA 会删除你查询语句里面的优先级。

---

## 高级搜索字段参考

JQL 中的字段是 JIRA 中的字段（或已在 JIRA 中定义的自定义字段）。在一个语句中，字段紧跟在运算符后面，而字段后面又跟着一个或多个值（或函数）。运算符比较字段的一个或多个值，使得查询语句仅返回真实的结果。

### 字段列表：

- [AffectedVersion](#)
- [Assignee](#)
- [Attachments](#)
- [Category](#)
- [Comment](#)
- [Component](#)
- [Created](#)
- [Creator](#)
- [CustomField](#)
- [Description](#)
- [Due](#)
- [Environment](#)
- [Epic Link](#)
- [Filter](#)
- [Fix Version](#)
- [Issue Key](#)
- [LastViewed](#)
- [Level](#)
- [Original Estimate](#)
- [Parent](#)
- [Priority](#)
- [Project](#)
- [Remaining Estimate](#)
- [Reporter](#)
- [Resolution](#)
- [Resolved](#)
- [Sprint](#)
- [Status](#)
- [Summary](#)
- [Text](#)
- [Type](#)
- [Time Spent](#)
- [Updated](#)

- [Voter](#)
- [Votes](#)
- [Watcher](#)
- [Watchers](#)
- [Work Ratio](#)

## Affected Version

搜索特定版本的问题。你可以通过版本名称或版本 ID（JIRA 自动分配的版本号）进行搜索。通过版本 ID 搜索比版本名称搜索更安全。

不同的项目可能有相同的版本名称，因此通过版本名称可能搜索出多个项目。也有可能 JIRA 的管理员更改了项目版本名称，破坏了已保存搜索。然而版本 ID 是独一无二的，而且不能被更改。

注：这个字段支持自动填写功能。

### 语法

affectedVersion

### 字段类型

VERSION

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

注意，比较运算符（如“>”）使用项目管理员设置的版本顺序，而不是按数字或字母顺序排列的。

### 支持的函数

当使用 **IN** 和 **NOT IN** 运算符时，该字段支持：

- releasedVersions()
- latestReleasedVersion()
- unreleasedVersions()
- earliestUnreleasedVersion()
- versionMatch()

### 示例

- 查询 AffectedVersion 为 3.14 的问题：  
affectedVersion = "3.14"  
(注意全站点保留字符要用引号)

- 查询 AffectedVersion 为"Big Ted"的问题:  
affectedVersion = "Big Ted"
- 查询 AffectedVersion ID 为 10350 的问题:  
affectedVersion = 10350

## Assignee

查询分配给特定用户的问题。你可以通过用户的全名、ID 或邮件地址进行查询。  
注：该字段支持自动填写功能。

### 语法

assignee

### 字段类型

USER

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓

### 支持的函数

当使用 **IN** 和 **NOT IN** 运算符时，该字段支持：

- membersOf()

当使用 **EQUALS** 和 **NOT EQUALS** 运算符时，该字段支持：

- currentUser()

### 示例

- 查找分配给 John Smith 的问题:  
assignee = "John Smith"  
或  
assignee = jsmith
- 查找当前或已经分配给 John Smith 的问题:  
assignee WAS "John Smith"  
或  
assignee WAS jsmith
- 查找分配给邮箱为"bob@mycompany.com"的用户的问题:  
assignee = "bob@mycompany.com"  
(注意全站点字符和“@”符号是保留字符，因此邮箱要加引号。)

## Attachments

查询带有附件或没有附件的问题。你可以仅使用该字段的 EMPTY 或 IS NOT EMPTY 运算符进行查询。

注：该字段支持自动填充功能。

### 语法

attachments

### 字段类型

ATTACHMENT

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✘	✘	✘	✘	✘	✘	✘	✘	✔	✔	✘	✘	✘	✘	✘	✘	✘

### 支持的函数

无

### 示例

- 查询带有附件的问题：  
attachments IS NOT EMPTY
- 查询不带附件的问题  
attachments IS EMPTY

## Category

查询某个具体分类项目里的问题。

注：该字段支持自动填写功能。

### 语法

category

### 字段类型

CATEGORY

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

- 查找属于"Alphabet Projects"类的项目的问题：  
category = "Alphabet Projects"

## Comment

查找包含特定文本评论的问题。也可以使用 JIRA 文本搜索语法。  
注：该字段不支持自动填写功能。

### 语法

comment

### 字段类型

TEXT

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

- 查找评论里包含有"My PC is quite old"的问题（模糊匹配）：  
comment ~ "My PC is quite old"
- 查询评论里含有完整语句"My PC is quite old"的问题：  
comment ~ "\"My PC is quite old\""

## Component

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

查找某个项目特定分组里的问题。你可以通过组名和组 ID（JIRA 自动分配的）进行查询。通过组 ID 查询比组名更安全，因为组 ID 是唯一的。

注：该字段支持自动填写功能。

### 语法

component

### 字段类型

COMPONENT

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

当使用 **IN** 和 **NOT IN** 运算符时支持：

- componentsLeadByUser()

### 示例

- 查找在"Comp1"或"Comp2"中的问题：  
component in (Comp1, Comp2)
- 查找同时在"Comp1"和"Comp2"中的问题：  
component in (Comp1) and component in (Comp2)  
或  
component = Comp1 and component = Comp2
- 查找组 ID 为 20500 的问题：  
component = 20500

## Created

查找某个特定日期前或后创建的问题（或某个日期范围）。注意，如果没有指定具体时间，默认为凌晨 00:00。查询结果是根据你的 JIRA 服务器设置的时区。

使用如下格式：

"yyyy/MM/ddHH:mm"

"yyyy-MM-ddHH:mm"

"yyyy/MM/dd"

"yyyy-MM-dd"

或者使用"w" (周), "d" (天), "h" (小时)或"m" (分钟)来指定相对于当前时间的日期。默认为“M”（分钟）。一定要使用引号（"）;如果省略了引号标记，你提供的数字将被解释为纪

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

元（1970-1-1）之后的毫秒。

注：该字段不支持自动填写功能。

### 语法

created

别名:

createdDate

### 字段类型

DATE

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

当使用 **EQUALS**, **NOT EQUALS**, **GREATER THAN**, **GREATER THAN EQUALS**, **LESS THAN** 或 **LESS THAN EQUALS** 运算符时，该字段支持：

- currentLogin()
- lastLogin()
- now()
- startOfDay()
- startOfWeek()
- startOfMonth()
- startOfYear()
- endOfDay()
- endOfWeek()
- endOfMonth()
- endOfYear()

### 示例

- 查找 2010 年 12 月 12 日 00:00 点之前创建的所有问题：  
created < "2010/12/12"
- 查找 2010 年 12 月 12 日创建或 2010 年 12 月 12 日之前创建的所有问题：  
created <= "2010/12/13"
- 查找 2010 年 12 月 12 日下午 2 点之前创建的所有问题：  
created > "2010/12/12" and created < "2010/12/12 14:00"
- 查找创建不足一天的问题：  
created > "-1d"
- 查找在 2011 年 1 月创建的所有问题：

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

created> "2011/01/01" and created < "2011/02/01"

- 查找在 2011 年 1 月 15 日创建的所有问题：  
created> "2011/01/15" and created < "2011/01/16"

## Creator

查找具体某个用户创建的问题，可以通过用户的全名、ID 或邮件地址进行查询。  
注：该字段支持自动填写功能。

### 语法

Creator

### 字段类型

USER

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓

### 支持的函数

当使用 **IN** 和 **NOT IN** 运算符时，该字段支持：

- membersOf()

当使用 **EQUALS** 和 **NOT EQUALS** 运算符时，该字段支持：

- currentUser()

### 示例

- 查找 Jill Jones 创建的所有问题：  
creator = "Jill Jones"  
或  
creator = jjones
- 查找邮箱为"bob@mycompany.com"的用户创建的所有问题：  
creator = "bob@mycompany.com"

## Custom Field

只有在 JIRA 管理员已创建自定义字段的情况下可用。

查找某个具体值的自定义字段的问题。你可以通过自定义字段名称或自定义字段 ID (JIRA 自动分配的) 进行查找。

通过 ID 查找比按名称查找更安全，因为 ID 是唯一的。

注：

JIRA 文本搜索语法可用于类型为“Text”的自定义字段。

自动填写功能支持自定义字段的 picker, group picker, select, check-box 和 radio button 字段类型。

### 语法

CustomFieldName

别名:

cf[CustomFieldID]

### 字段类型

取决于自定义字段的配置。

### 支持的运算符

不同类型的自定义运算符支持不同的运算符。

**Number** 和 **date/time** 类型:

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

**Picker, select, check-box** 和 **radio button** 类型:

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

**Text** 类型:

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✗	✗	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗

### 支持的函数

不同类型的自定义字段支持不同的函数：

Date/time 类型：当使用 **EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN** 或 **LESS THAN EQUALS** 运算符时，该字段支持：

- currentLogin()
- lastLogin()
- now()
- startOfDay()
- startOfWeek()
- startOfMonth()

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- startOfYear()
- endOfDay()
- endOfWeek()
- endOfMonth()
- endOfYear()

Version picker 类型: 当使用 **IN** 和 **NOT IN** 运算符时, 该字段支持:

- releasedVersions()
- latestReleasedVersion()
- unreleasedVersions()
- earliestUnreleasedVersion()
- versionMatch()

示例

- 查找自定义字段"Location"值为"New York"的问题:  
location = "New York"
- 查找 ID 为 10003 的自定义字段值为"New York"的问题:  
cf[10003] = "New York"
- 查找名为"Location"的自定义字段值为"London"或"Milan"或"Paris"的问题:  
cf[10003] in ("London", "Milan", "Paris")
- 查找名为"Location"自定义字段的所有问题:  
location != empty

## Description

通过描述查找包含具体文本的问题。也可用 JIRA 文本搜索语法。

注: 该字段不支持自动填写功能。

语法

description

字段类型

TEXT

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED

支持的函数

n/a

示例

- 查找描述文本里含有"Please see screenshot"的问题 (模糊匹配) :

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式:

电话: 023-66090381|400-700-1020

邮箱: [sales@evget.com](mailto:sales@evget.com)

网址: <http://www.evget.com/product/3516>

description ~ "Please see screenshot"

- 查找描述文本里含有完整语句"Please see screenshot"的问题:  
description ~ "\"Please see screenshot\""

## Due

根据某个具体的截止日期之前或之后（或某个时间段）来搜索问题。注意截止日期仅和日期（而不是时间）有关。格式如下：

"yyyy/MM/dd"

"yyyy-MM-dd"

或使用"w"(周)或"d"(天)指定当前日期，记住要使用双引号。该字段不支持自动填写功能。

### 语法

due

别名：

dueDate

### 字段类型

DATE

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

当使用 **EQUALS**, **NOT EQUALS**, **GREATER THAN**, **GREATER THAN EQUALS**, **LESS THAN** 或 **LESS THAN EQUALS** 运算符时，该字段支持：

- currentLogin()
- lastLogin()
- now()
- startOfDay()
- startOfWeek()
- startOfMonth()
- startOfYear()
- endOfDay()
- endOfWeek()
- endOfMonth()

- endOfYear()

示例:

- 查找在 2010 年 12 月 31 日前截止的所有问题:  
due < "2010/12/31"
- 查找截止到 2010 年 11 月 31 日的所有问题:  
due <= "2011/01/01"
- 查找明天截止的所有问题:  
due = "1d"
- 查找所有在 2011 年 1 月这个月截止的问题:  
due >= "2011/01/01" and due <= "2011/01/31"
- 查找所有在 2011 年 1 月 15 日这天截止的问题:  
due = "2011/01/15"

## Environment

查找 Environment 字段包含某个特定文本的问题，也可用 JIRA 文本搜索。注：该字段不支持自动填写功能。

语法

environment

字段类型

TEXT

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED

支持的函数

n/a

示例

- 查找 Environment 字段包含"Third floor"的问题（模糊匹配）：  
environment ~ "Third floor"
- 查找 Environment 字段包含完整"Third floor"语句的问题：  
environment ~ "\"Third floor\""

## Epic Link

仅适用于 JIRA Agile 6.1.2 或以后的版本。

查找属于 JIRA Agile 中某个具体 Epic 的问题，可以根据 Epic 名称、问题键或问题 ID 进行查询。注：该字段不支持自动填写功能。

### 语法

"epic link"

### 字段类型

Epic Link Relationship (这是通过 JIRA Agile 创建的自定义类型).

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

当使用 **IN** 或 **NOT IN** 运算符时，该字段支持：

- issueHistory()
- linkedIssues()
- votedIssues()
- watchedIssues()

### 示例

- 查找属于名为"Jupiter"，问题键为 ANERDS-317 的 epic 的问题。

"epic link" = ANERDS-317

或

"epic link" = Jupiter

## Filter

可以使用保存的过滤器来缩小搜索范围，通过过滤器名称或过滤器的 ID 进行搜索。

注：

在保存的过滤器中键入的高级搜索-字段参考声明将覆盖 **ORDER BY** 声明。

不能运行或保存可能引起无限循环的过滤器（例如：你不能参考已保存的参考正在运行的过滤器的过滤器）。

支持自动填写功能。

### 语法

filter

别名:

request

savedFilter

searchRequest

字段类型

FILTER

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗

支持的函数

n/a

示例

- 查询 ID 为 12000, 名称为 "My Saved Filter" 的过滤器授权给 jsmith 用户的问题:  
filter = "My Saved Filter" and assignee = jsmith  
或  
filter = 12000 and assignee = jsmith

## Fix Version

查询某个具体修复版本的问题，可通过版本名称或版本 ID 进行查询。注：该字段支持自动填写。

语法

fixVersion

字段类型

VERSION

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

支持的函数

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式:

电话: 023-66090381|400-700-1020

邮箱: [sales@evget.com](mailto:sales@evget.com)

网址: <http://www.evget.com/product/3516>

当使用 **IN** 和 **NOT IN** 运算符时，该字段支持：

- releasedVersions()
- latestReleasedVersion()
- unreleasedVersions()
- earliestUnreleasedVersion()
- versionMatch()

示例

- 查找修复版本为 3.14 或 4.2 的问题：  
fixVersion in ("3.14", "4.2")  
(注意保留字符要加双引号)
- 查询修复版本为"Little Ted"的问题：  
fixVersion = "Little Ted"
- 查询修复版本 ID 为 10001 的问题：  
fixVersion = 10001
- 查询修复版本为 15.2 或其子版本的问题：  
fixVersion in (versionMatch("15.2.\*"))

## Issue Key

通过具体的问题键或问题 ID 查询问题。注：该字段不支持自动填写功能。

语法

issueKey

别名：

id

issue

key

字段类型

ISSUE

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

支持的函数

当使用 **IN** 或 **NOT IN** 运算符时，该字段支持：

- issueHistory()

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- linkedIssues()
- votedIssues()
- watchedIssues()

#### 示例

- 查找问题键为"ABC-123"的问题：  
issueKey = ABC-123

## LastViewed

查找最后一次某个具体的日期或时间段浏览过的问题，默认午夜 00:00 为时间分量，格式如下：

"yyyy/MM/ddHH:mm"

"yyyy-MM-ddHH:mm"

"yyyy/MM/dd"

"yyyy-MM-dd"

或使用"w" (周), "d" (天), "h" (时) 或 "m" (分)来指定当前相对时间。注：该字段不支持自动填写功能。

#### 语法

lastViewed

#### 字段类型

DATE

#### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

#### 支持的函数

当使用 **EQUALS**, **NOT EQUALS**, **GREATER THAN**, **GREATER THAN EQUALS**, **LESS THAN** 或 **LESS THAN EQUALS** 运算符时，该字段支持：

- currentLogin()
- lastLogin()
- now()
- startOfDay()
- startOfWeek()
- startOfMonth()
- startOfYear()

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- endOfDay()
- endOfWeek()
- endOfMonth()
- endOfYear()

#### 示例

- 查找最后一次浏览是在 2010 年 12 月 12 日之前的所有问题：  
lastViewed< "2010/12/12"
- 查找最后一次浏览是在 2010 年 12 月 12 日当天或之前的所有问题：  
lastViewed<= "2010/12/13"
- 查找最后一次浏览是在 2010 年 12 月 12 日这一天并且是在下午 2 点之前的所有问题：  
lastViewed> "2010/12/12" and created < "2010/12/12 14:00"
- 查找最后一次浏览是在一天以前的问题：  
lastViewed> "-1d"
- 查找最后一次浏览是在 2011 年 1 月的所有问题：  
lastViewed> "2011/01/01" and created < "2011/02/01"
- 查找最后一次浏览是在 2011 年 1 月 15 日这一天的问题：  
lastViewed> "2011/01/15" and created < "2011/01/16"

## Level

仅当 Issue Level Security 被 JIRA 管理员启用时可用。

通过具体的安全级别查找问题，可通过问题安全级别名称或 ID 进行查询。注：该字段支持自动填写功能。

### 语法

level

### 字段类型

SECURITY LEVEL

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

- 查找问题安全级别为"Really High"或"level1"的问题：

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

level in ("Really High", level1)

- 查找问题安全级别 ID 为 123 的问题：  
level = 123

## Original Estimate

仅当 JIRA 管理员启用时间追踪后可用。

通过具体的初始估计值（是数字，而不是日期或日期范围）查找问题。使用 "w", "d", "h" 和 "m" 来指定周、天、时、分。注：该字段不支持自动填写功能。

### 语法

originalEstimate

别名：

timeOriginalEstimate

### 字段类型

DURATION

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

- 查找初始估计为 1 小时的问题：  
originalEstimate = 1h
- 查找初始估计大于 2 天的问题：  
originalEstimate > 2d

## Parent

仅当 JIRA 管理员启用子任务后可用。

查找某个具体问题的所有子任务，可通过问题键或问题 ID 进行查找。注：该字段不支持自动填写功能。

### 语法

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

parent

字段类型

ISSUE

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗

支持的函数

n/a

示例

- 查找 TEST-1234 问题的子任务：  
parent = TEST-1234

## Priority

根据具体的优先级搜索问题，可通过优先级名称和优先级 ID 进行查询。注：该字段支持自动填写功能。

语法

priority

字段类型

PRIORITY

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

支持的函数

n/a

示例

- 查找优先级为"High"的问题：  
priority = High
- 查找优先级 ID 为 10000 的问题：

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

priority = 10000

## Project

查找属于某个具体项目的问题，可通过项目名称、项目键或项目 ID 进行搜索。极少数情况，若项目键相同时，搜索结果按偏好隐藏搜索出来的其它项目。注：该字段支持自动填写功能。

### 语法

Project

字段类型

PROJECT

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

当使用 **IN** 和 **NOT IN** 运算符时，该字段支持：

- projectsLeadByUser()
- projectsWhereUserHasPermission()
- projectsWhereUserHasRole()

### 示例

- 查找属于名为"ABC Project"项目的问题：  
project = "ABC Project"
- 查找属于键为"ABC"的项目的问题：  
project = "ABC"
- 查找属于项目 ID 为"1234"的项目的问题：  
project = 1234

## Remaining Estimate

仅当 JIRA 管理员启用 time-tracking 后可用。

查询剩余时间设置为某个具体值（是数值而不是日期或时间段）的问题。使用"w","d","h"和"m"来指定周、天、时和分。注：该字段不支持自动填写功能。

### 语法

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

remainingEstimate

别名:

timeEstimate

字段类型

DURATION

支持的语法

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

支持的函数

n/a

示例

- 查找剩余时间超过 4 小时的问题:  
remainingEstimate> 4h

## Reporter

搜索被某个具体用户报告的问题, 和问题创建者类似, 但又有所不同, 可通过用户全名、ID 或邮件地址进行查询。注: 该字段支持自动填写功能。

语法

reporter

字段类型

USER

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓

支持的函数

当使用 **IN** 和 **NOT IN** 运算符时, 该字段支持:

- membersOf()

当使用 **EQUALS** 和 **NOT EQUALS** 运算符时, 该字段支持:

- currentUser()

#### 示例

- 查找被 Jill Jones 报告的问题:  
reporter = "Jill Jones"  
或  
reporter = jjones
- 查找被用户邮箱为"bob@mycompany.com"报告的问题:  
reporter = "bob@mycompany.com"  
(注意全站保留字符和@符号要加双引号。)

## Resolution

搜索已经有具体解决方案的问题，可通过解决方案名称或 ID 进行查询。注：该字段支持自动填写功能。

#### 语法

resolution

#### 字段类型

RESOLUTION

#### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

#### 支持的函数

n/a

#### 示例

- 查找解决方案为"Cannot Reproduce"或"Won't Fix"的问题:  
resolution in ("Cannot Reproduce", "Won't Fix")
- 查找解决方案 ID 为 5 的问题:  
resolution = 5
- 查找还没有解决方案的问题:  
resolution = unresolved

## Resolved

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式:

电话: 023-66090381|400-700-1020

邮箱: [sales@evget.com](mailto:sales@evget.com)

网址: <http://www.evget.com/product/3516>

搜索在某个具体的日期或日期范围内解决的问题。注意默认时间分量为午夜 00:00，搜索结果参照 JIRA 服务器时间。格式如下：

"yyyy/MM/ddHH:mm"

"yyyy-MM-ddHH:mm"

"yyyy/MM/dd"

"yyyy-MM-dd"

或使用"w" (周), "d" (天), "h" (时)或"m" (分)来指定当前相对时间。注意加双引号，否则将按照(1970-1-1)后的毫秒计算。注：该字段不支持自动填写功能。

### 语法

resolved

别名：

resolutionDate

### 字段类型

DATE

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

当使用 **EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN** 或 **LESS THAN EQUALS** 运算符时，该字段支持：

- currentLogin()
- lastLogin()
- now()
- startOfDay()
- startOfWeek()
- startOfMonth()
- startOfYear()
- endOfDay()
- endOfWeek()
- endOfMonth()
- endOfYear()

### 示例

- 查找 2010 年 12 月 31 日前（不包括 31 日）解决的所有问题：  
resolved<= "2010/12/31"
- 查找在 2010 年 12 月 31 日下午 2 点之前解决的所有问题：

resolved < "2010/12/31 14:00"

- 查找在 2010 年 12 月 31 日这一天或之前解决的所有问题：

resolved <= "2011/01/01"

- 查找在 2011 年 1 月分解决的所有问题：

resolved > "2011/01/01" and resolved < "2011/02/01"

- 查找在 2011 年 1 月 15 日这一天解决的所有问题：

resolved > "2011/01/15" and resolved < "2011/01/16"

- 查找在过去一个小时内解决的问题：

resolved > -1h

## Sprint

仅当使用 JIRA Agile 时可用。

搜索分配给 JIRA Agile 中特定 sprint 的问题，可通过 sprint 名称和 ID 进行查询。

### 语法

sprint

如果你的多个 sprint 拥有相似的名称，你可以通过名称进行简单查询-或这名称的部分。可能匹配的将自动填充在下拉框中，同时日期和 ID 也会显示出来以供区别。

### 字段类型

Number

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

- openSprints()
- closedSprints()

### 示例

- 查找属于 sprint999 的问题：  
sprint = 999
- 查找属于 sprint "February 1"的问题：  
sprint = "February 1"
- 查找属于"February 1"，"February 2"或"February 3"的问题：  
sprint in ("February 1","February 2","February 3")
- 查找属于 sprint 的所有问题：

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

sprint is not empty

## Status

搜索某个具体状态的问题，可通过状态名称或 ID 进行查询。

注意：WAS, WAS\_NOT, WAS\_IN and WAS\_NOT\_IN 运算符只能用于状态名称（不能用于 ID）。

注：该字段支持自动填写功能。

### 语法

Status

字段类型

STATUS

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓

支持的函数

n/a

示例

- 查找状态为"Open"的问题：  
status = Open
- 查找状态 ID 为 1 的问题：  
status = 1
- 查找当前或曾经状态为"Open"的问题：  
status WAS Open

## Summary

搜索总结里包含具体文本的问题，也可使用文本搜索。注：该字段不支持自动填写功能。

### 语法

Summary

字段类型

TEXT

支持的运算符

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✗	✗	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗

## 支持的函数

n/a

## 示例

- 查找总结里含有"Error saving file"的问题（模糊匹配）：  
summary ~ "Error saving file"
- 查找总结里含有"Error saving file"精准语句的问题：  
summary ~ "\"Error saving file\""

## Text

该字段允许搜索所有文本类字段，如：

- Summary
- Description
- Environment
- Comments
- custom 该字段使用自由文本搜索器；自定义字段类型如下：
  1. Free Text Field (无限制文本)
  2. Text Field (< 255 字节)
  3. Read-only Text Field

尽管 JQL 中的 Text 字段可以搜索所有文本类字段的问题如：Summary, Description, Environment, Comments 等，但为了使搜索更精确，在搜索的时候尽量使用某个具体的字段，如：

Summary ~ "some words" OR Description ~ "some words"

注：

text 字段只能和 CONTAINS 运算符("~")一起使用。

## 语法

text

## 字段类型

TEXT

## 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

- 查找文本字段里包含"Fred"的问题：  
text ~ "Fred"  
或  
text ~ Fred
- 查找文本字段里包含"full screen"精准语句的问题：  
text ~ "\"full screen\""

## Type

搜索某个具体类型的问题，可通过类型名称或 ID 进行搜索。注：该字段支持自动填写功能。

### 语法

type

别名：

issueType

字段类型

ISSUE\_TYPE

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

- 查找类型为"Bug"的问题：  
type = Bug

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- 查找类型为"Bug"或"Improvement"的问题:  
issueType in (Bug,Improvement)
- 查找类型 ID 为 2 的问题:  
issueType = 2

## Time Spent

仅当 JIRA 管理员启用 time-tracking 后可用。

搜索已耗费时间为某个具体值的问题。使用"w", "d", "h"和"m"来指定周、天、时和分。

注：该字段不支持自动填写功能。

### 语法

timeSpent

字段类型

DURATION

支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

支持的函数

n/a

示例

- 查找已耗费时间超过 5 天的问题:  
timeSpent> 5d

## Updated

搜索上一次更新是在某个具体的日期或日期段的问题,注意午夜 00:00 为默认时间分量,查询参照 JIRA 服务器时间。格式如下:

"yyyy/MM/ddHH:mm"

"yyyy-MM-ddHH:mm"

"yyyy/MM/dd"

"yyyy-MM-dd"

或使用"w" (周), "d" (天), "h" (时)或"m" (分)来指定当前相对时间,默认是分。确保使用双引号(""),否则将按照(1970-1-1)后的毫秒来计算。注:该字段不支持自动填写功能。

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式:

电话: 023-66090381|400-700-1020

邮箱: [sales@evget.com](mailto:sales@evget.com)

网址: <http://www.evget.com/product/3516>

## 语法

updated

别名:

updatedAt

## 字段类型

DATE

## 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

## 支持的函数

当使用 **EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN** 或 **LESS THAN EQUALS** 运算符时，该字段支持:

- currentLogin()
- lastLogin()
- now()
- startOfDay()
- startOfWeek()
- startOfMonth()
- startOfYear()
- endOfDay()
- endOfWeek()
- endOfMonth()
- endOfYear()

## 示例

- 查找上一次更新是在 2010 年 12 月 12 日之前的问题:  
updated< "2010/12/12"
- 查找上一次更新是在 2010 年 12 月 12 日或之前的问题:  
updated< "2010/12/13"
- 查找上一次更新是在 2010 年 12 月 31 日下午 2 点之前的所有问题:  
updated< "2010/12/31 14:00"
- 查找上一次更新离现在不超过 2 周的问题:  
updated< "-2w"
- 查找上一次更新是在 2011 年 1 月 15 日这一天的问题:  
updated> "2011/01/15" and updated < "2011/01/16"
- 查找上一次更新是在 2011 年 1 月份的问题:

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式:

电话: 023-66090381|400-700-1020

邮箱: [sales@evget.com](mailto:sales@evget.com)

网址: <http://www.evget.com/product/3516>

updated> "20011/01/01" and updated < "2011/02/01"

## Voter

搜索某个具体的用户投票的问题，可通过用户全称、ID、邮箱地址进行查询。你只能搜索你拥有“查看投票人和浏览人”权限的问题，否则只能搜索你自己投的问题。注：该字段支持自动填写功能。

### 语法

voter

### 字段类型

USER

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

当使用 **IN** 和 **NOT IN** 运算符时，该字段支持：

- membersOf()

当使用 **EQUALS** 和 **NOT EQUALS** 运算符时，该字段支持：

- currentUser()

### 示例

- 查找自己投过的问题：  
voter = currentUser()
- 查找用户"jsmith"投过的问题：  
voter = "jsmith"
- 查找"jira-developers"组的成员投过的问题：  
voter in membersOf("jira-developers")

## Votes

搜索有具体票数的的问题。注：该字段不支持自动填写功能。

### 语法

votes

## 字段类型

NUMBER

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

- 查找所有拥有 12 票的问题：  
votes>= 12

## Watcher

搜索某个具体的用户浏览过的问题，可通过用户全称、ID、邮箱地址进行查询。只能搜索自己有“查看投票人和浏览人”权限的问题，否则只能搜索你自己浏览过的问题。注：该字段支持自动填写功能。

### 语法

Watcher

### 字段类型

USER

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

当使用 **IN** 和 **NOT IN** 运算符时，该字段支持：

- membersOf()

当使用 **EQUALS** 和 **NOT EQUALS** 运算符时，该字段支持：

- currentUser()

### 示例

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

- 查找所有自己浏览过的问题：  
watcher = currentUser()
- 查找用户"jsmith"浏览过的问题：  
watcher = "jsmith"
- 查找"jira-developers"组的成员浏览过的问题：  
watcher in membersOf("jira-developers")

## Watchers

按具体浏览的用户量搜索问题。注：该字段不支持自动填写功能。

### 语法

watchers

### 字段类型

NUMBER

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

- 查找浏览量大于 3 的问题：  
watchers > 3

## Work Ratio

仅当 JIRA 管理员启用 time-tracking 后可用。

搜索拥有具体工作比值的问题，计算公式如下：

$workRatio = timeSpent / originalEstimate \times 100$  (即比值=已耗费时间/初始估计时间\*100)

注：该字段不支持自动填写功能。

### 语法

workRatio

### 字段类型

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

## NUMBER

### 支持的运算符

=	!=	~	!~	>	>=	<	<=	IS	IS NOT	IN	NOT IN	WAS	WAS IN	WAS NOT	WAS NOT IN	CHANGED
✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗

### 支持的函数

n/a

### 示例

查找比值超过 75%的问题:

workRatio> 75

## 共享搜索结果

[JIRA](#) 可以使你很容易通过电子邮件或其它方式与其他用户共享搜索结果或过滤器链接。还可以在电子邮件中选择性添加注释。

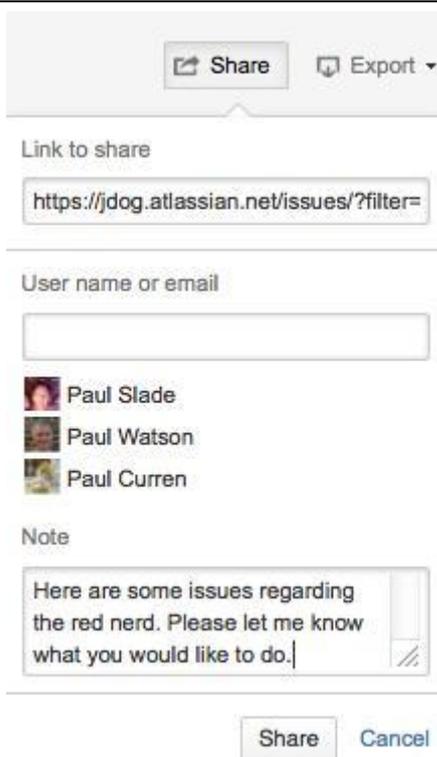
要访问共享按钮，你的 JIRA 系统管理员必须配置有 JIRA 的 SMTP 邮件服务器。此外，还需要全局用户权限。

与一个或多个 JIRA 用户或任何电子邮件地址共享检索结果方法如下：

1. 选择 Issues > Search for Issues.
2. 精确搜索，然后选择共享图标。

键盘快捷键：s

3. 指定 JIRA 用户(通过输入其用户名或其注册 JIRA 帐户全名的部分/全部)，或键入想分享的任何人的电子邮件地址。当开始输入一个 JIRA 用户的用户名或姓名，或先前指定的电子邮件地址，自动完成下拉菜单就会出现。



Share Export

Link to share

https://jdog.atlassian.net/issues/?filter=

User name or email

Paul Slade

Paul Watson

Paul Curren

Note

Here are some issues regarding the red nerd. Please let me know what you would like to do.

Share Cancel

4. 添加注释。
5. 点击 Share 按钮。

## 更新概览

### v6.4 版本更新

#### 新增 Release Hub

JIRA6.4 版本引进了 Release Hub，通过一个简单的指示板让团队可以实时查看现状和即将到来的项目新进展，并且降低了风险，为管理者提供了一个更自信的无压力的管理过程。

#### 高效的进度管理

Release Hub 将团队已熟悉的开发工具整合到一起，从每一个项目的开始进行追踪的，因此当管理日期结束，项目也就跟着完成了。

---

## 在 HipChat 中解决所有问题

团队可在 HipChat 中通过实时交流来更快地解决问题，并且任何与 JIRA 项目有关的信息都存在 rooms 里。

### 快速验证 workflow

JIRA 有所有新的工具来帮助修复故障和调试 workflow。新的 workflow 验证器通过快速浏览和识别任何 workflow 里的潜在问题，为用户节约了大量调试的时间。

### 快速上手

新版本的登录和运行更简单，通过全新的管理体验新队员将会有更多的时间来做项目而不是熟悉 JIRA 的基础知识。

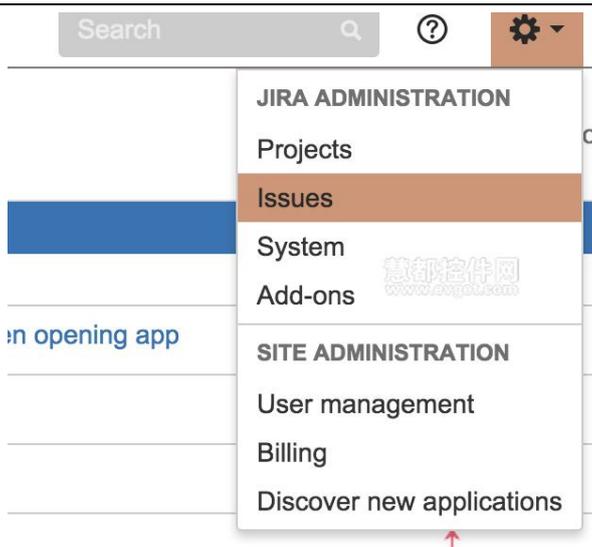
## 案例赏析

### 配置项目权限

[JIRA](#) 是一个非常强大的工具，正确使用它可以帮助你组织工作以及管理团队的进度。默认情况下，所有的项目对所有用户都完全开放的。这是为了方便初次使用 JIRA 的人使用，但大多数用户不喜欢这样，下面介绍如何配置使得一个项目只能被该项目的用户所见。举个例子，我们创建一个名为 Project Awesome 的项目。接下来的几个步骤，一旦掌握你就会发现其实很简单。

#### 第一步：分组

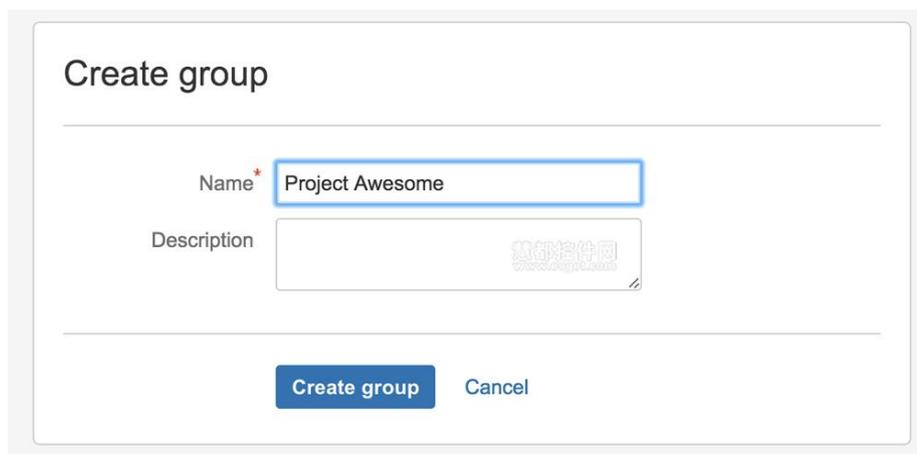
JIRA 里的组是一群用户的集合，一个组可以有多个用户，并且一个用户可以同时属于多个组。点击 **User management** 。



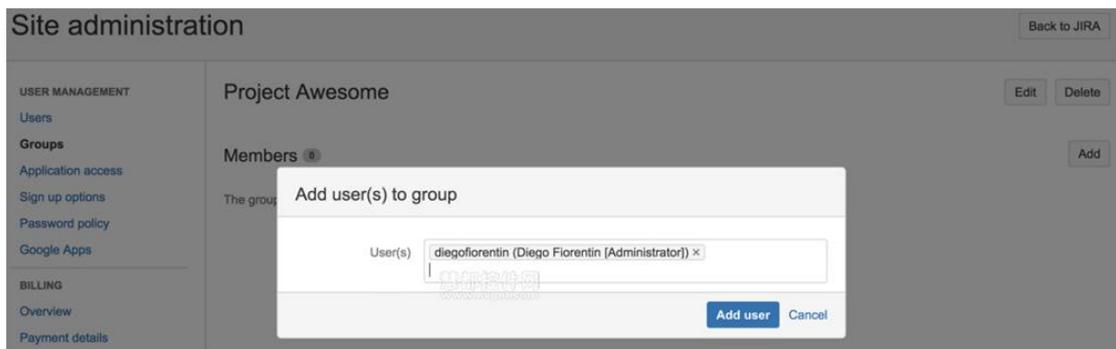
在此点击 **Groups** 然后点击 **Create group**



在这里输入组的名称，最好是把组名和项目名称设置成同一个。

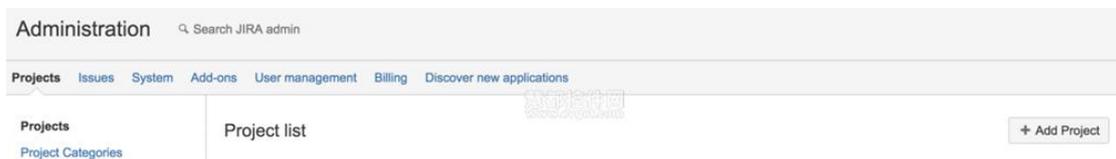


一旦组创建完成，你就可以往组里添加用户了。输入用户名，点击 **Add user** 就可以了。记住你可以添加多个用户。

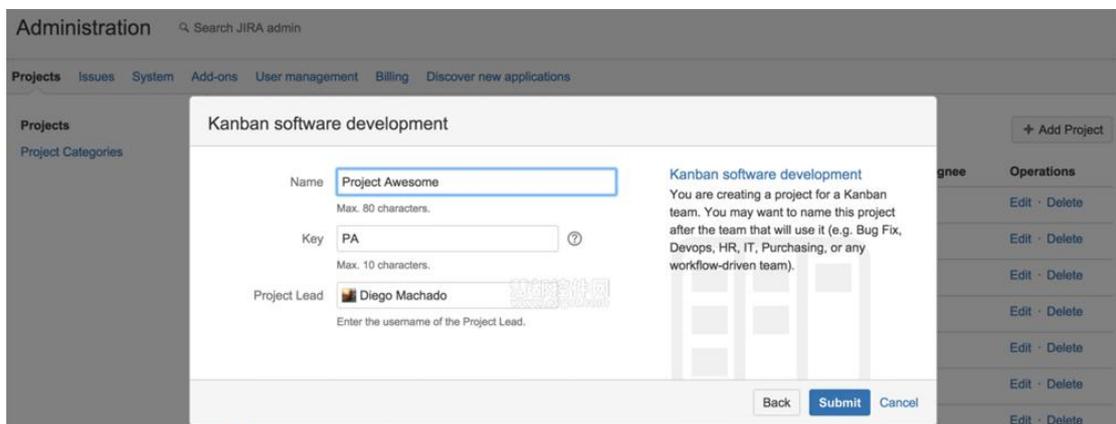


## 第二步：创建项目

返回 JIRA，我们开始创建项目



点击 **Add Project** 然后跟着步骤来，记住必须选择项目类型和项目的名称。



JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

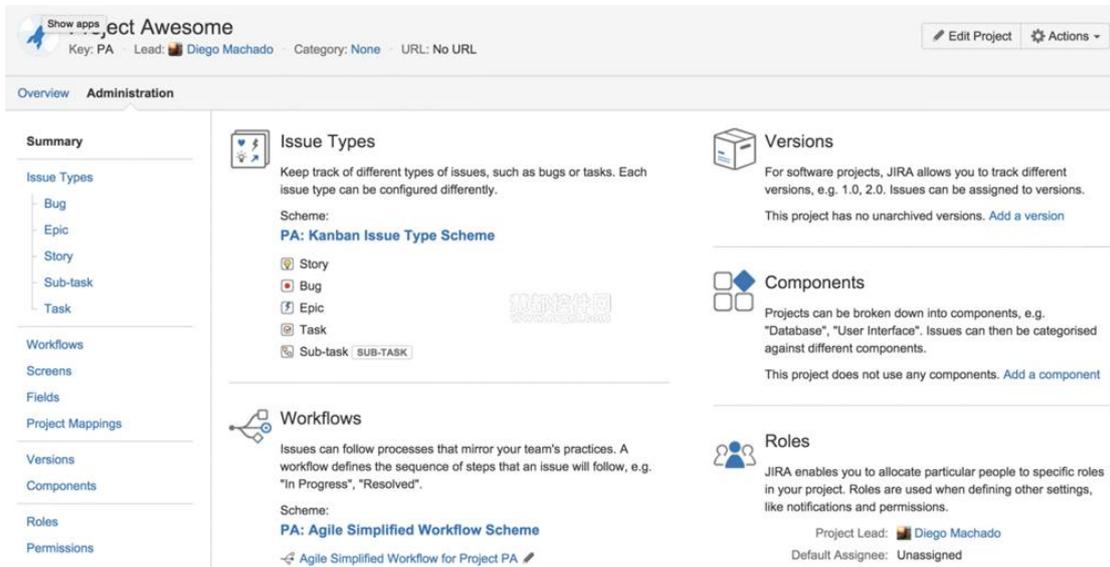
现在为了使你的项目组的用户可以访问该项目，是时候配置你的项目了。

### 第三步：角色

在项目列表里点击你的项目进入项目概览界面。



点击左边列表里的 Roles



### Roles

JIRA enables you to allocate particular people to specific roles in your project. Roles are used when defining other settings, like notifications and permissions.

- Project Lead: Diego Machado
- Default Assignee: Unassigned

Project Roles	Users	Groups
Administrators		administrators
Developers		jira-developers
Partners		
Tempo Project Managers		
Users		jira-users

在这里你可以看到项目的所有不同角色，**jira-users** 组是默认的。现在我们要做的就是从 User 角色里删除 **jira-users** 组，增加我们在第一步创建名为 **Project Awesome** 的组。提示：你的用户要做的不仅仅是查看任务，意味着你需要你的用户能创建任务，完成任务以及评论。你也可以将你的组添加到

JIRA-团队项目跟踪管理专家

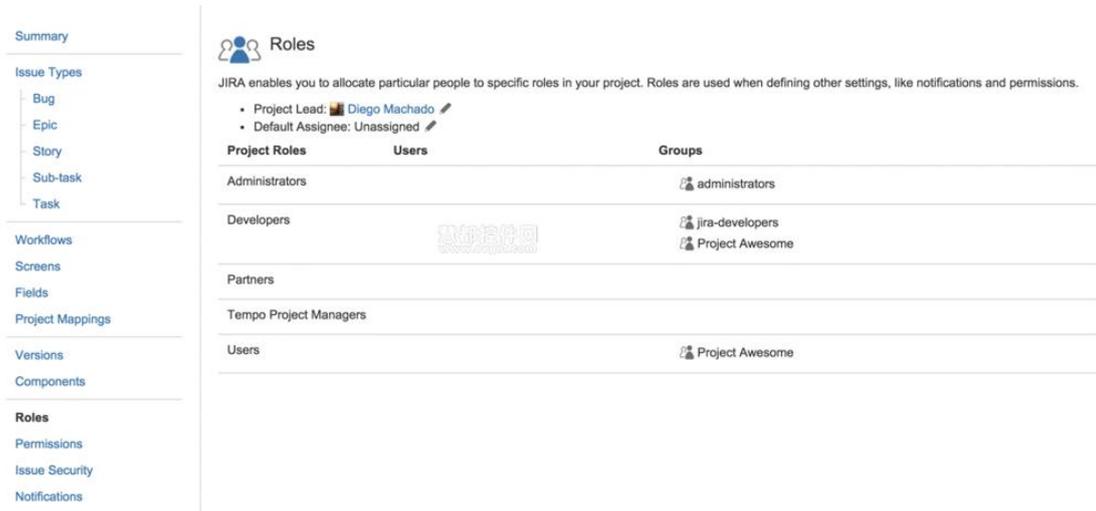
购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

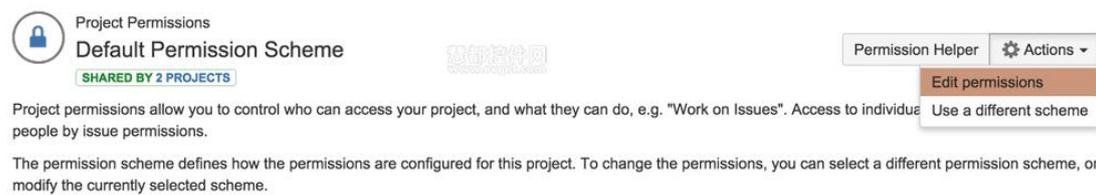
网址：<http://www.evget.com/product/3516>

Developers 角色，不要删除 jira-developers，因为这里有你所有的开发人员。



#### 第四步：分配权限

这是最棘手的一部分，点击左侧的 **Permissions** 链接



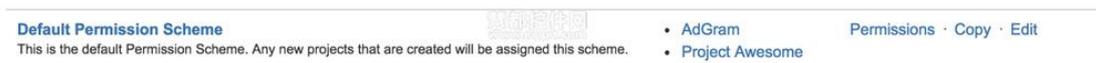
#### Edit Permissions — Default Permission Scheme

SHARED BY 2 PROJECTS

On this page you can edit the permissions for the "Default Permission Scheme" permission scheme.

- [Grant permission](#)
- [View all permission schemes](#)

Default Permission Scheme 默认分配给所有项目。点击 Default Permission Scheme 的 Copy 链接创建一个此方案的副本



然后你会看到新建的方案如下图所示

JIRA-团队项目跟踪管理专家

购买及相关事宜咨询方式：

电话：023-66090381|400-700-1020

邮箱：[sales@evget.com](mailto:sales@evget.com)

网址：<http://www.evget.com/product/3516>

**Copy of Default Permission Scheme** Permissions · Copy · Edit · Delete  
 This is the default Permission Scheme. Any new projects that are created will be assigned this scheme.

---

**Default Permission Scheme** Permissions · Copy · Edit  
 This is the default Permission Scheme. Any new projects that are created will be assigned this scheme.

- AdGram
- Project Awesome

点击 **Copy of Default Permission Scheme** 的 **Edit** 按钮修改它的名称，输入描述然后点击 **Update**

Edit Permission Scheme: Copy of Default Permission Scheme

Name

Description

接着，点击 **Permissions** 修改权限。在接下来的页面，你会看到所有权限列表，我们只关心一个特定的权限列表，那就是 **Browse Projects**，此权限允许一个组的用户查看搜索项目结果列表。正如你所见默认情况下该权限分配的组是 **jira-users**，这意味着所有用户将能够浏览该项目。让我们做一下修改。点击 **Delete** 删除 **jira-users** 组，然后点击 **Add** 添加一个新的组

Edit Permissions — Project Awesome ?

On this page you can edit the permissions for the "Project Awesome" permission scheme.

- [Grant permission](#)
- [View all permission schemes](#)

Project Permissions	Users / Groups / Project Roles	Operations
<b>Administer Projects</b> Ability to administer a project in JIRA.	<ul style="list-style-type: none"> <li>• Project Role (Administrators) <a href="#">(Delete)</a></li> </ul>	<a href="#">Add</a>
<b>Browse Projects</b> Ability to browse projects and the issues within them.	<ul style="list-style-type: none"> <li>• Group (jira-users) <a href="#">(Delete)</a></li> </ul>	<a href="#">Add</a>
<b>View Development Tools</b> Allows users to view development-related information on the view issue screen, like commits, reviews and build information.	<ul style="list-style-type: none"> <li>• Project Role (Developers) <a href="#">(Delete)</a></li> </ul>	<a href="#">Add</a>
<b>View Read-Only Workflow</b> Users with this permission may view a read-only version of a workflow.	<ul style="list-style-type: none"> <li>• Project Role (Users) <a href="#">(Delete)</a></li> </ul>	<a href="#">Add</a>
Issue Permissions	Users / Groups / Project Roles	Operations

接下来，你必须选择组选项，然后键入我们的组名称，然后点击 **Add**

### Add New Permission

Permission Scheme: **Project Awesome**

Please select the type of permission you wish to add to this Permission Scheme

- Permissions
- Add Comments
  - Administer Projects
  - Assignable User
  - Assign Issues
  - Browse Projects**
  - Close Issues
  - Create Attachments

(Select the permissions that you want to assign).

- User Custom Field Value
- Project Role
- Reporter
- Current Assignee
- Single User   
Start typing to get a list of possible matches.
- Project Lead
- Group
- Group Custom Field Value

Add Cancel

现在我们的 **Browse Projects** 就有我们想要的组了。然后点击 **Back to project: Project Awesome**

Project Permissions  
Default Permission Scheme  
SHARED BY 2 PROJECTS

Project permissions allow you to control who can access your project, and what they can do, e.g. "Work on Issues". Access to individual people by issue permissions.

The permission scheme defines how the permissions are configured for this project. To change the permissions, you can select a different permission scheme, or modify the currently selected scheme.

Permission Helper Actions  
Edit permissions  
Use a different scheme

点击 **Use a different scheme** 更改我们项目的模式。选择我们刚创建的模式，点击 **Associate**

---

## Associate Permission Scheme to Project

---

This page allows you to associate a permission scheme with this project.

Scheme  

慧都控件网  
www.evget.com

Associate

Cancel

现在，就只有我们 Project Awesome 组的用户能访问我们的 Project Awesome 项目了。

### [JIRA Dashboards 入门演示](#)

### [创建、共享和定制仪表板案例演示](#)

### [创建问题演示](#)

**[【在线免费试用申请】](#)**

本文档由慧都控件网 ([www.evget.com](http://www.evget.com)) 译制，版权所有，欢迎任何形式的转载和学习交流，未经许可禁止用于商业用途，请尊重他人劳动成果，违者必究。