

Update User Preferences

Page Size:

Email Type:

Language:

Time Zone:

My Changes:

Filter & Dashboard Sharing:

Keyboard shortcuts:

JTest

File JUnit URL Test URL Category None

Summary

- New Types
- Workflows
- Schemes
- People
- Permissions
- Notifications
- Settings

New Types

Keep track of different types of issues, such as bugs or tasks. Each issue type can be configured differently.

Schema: Default Issue Type Scheme

- ☐ New
- ☐ New Feature
- ☐ Bug (default)
- ☐ Improvement

Workflows

Schemes can follow processes that mirror your team's practices. A workflow defines the sequence of steps that an issue will follow, e.g. "To Progress", "Resolved".

Schema: Default Workflow Scheme

- ☐ you (default)

Schemes

Schemes allow you to arrange the fields to be displayed for an issue. Different schemes can be used when an issue is created, moved, added, or transitioned through a workflow.

Schema: Default Issue Type Scheme Scheme

- ☐ Default Scheme Scheme (default)
- ☐ JTest Scheme Scheme

People

Different issues can have different information fields. A field configuration defines how fields behave for the project, e.g. required/optional, hidden/visible.

Schema: System Default Field Configuration

- ☐ Default Field Configuration (default)
- ☐ you (default)

Settings

Some general project configuration options.

- ☐ JUnit Modules
- ☐ Application Links
- ☐ Configuration Space: No instances available
- ☐ Subversion Repository: No instances available
- ☐ Maven Repository: None ()
- ☐ CVS Repository: No instances available
- ☐ Buildout Project: No instances available
- ☐ Reporting Path: None ()

People

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

Project Lead:

Versions

For software projects, JTest allows you to track different versions, e.g. 1.0, 2.0. Issues can be assigned to versions. This project has no associated versions.

Components

Projects can be broken down into components, e.g. "Database", "User Interface". Issues can then be categorized against different components. This project does not use any components.

Permissions

Project permissions allow you to control who can access your project, and what they can do, e.g. "Write an issue". Access to individual issues is granted to people by issue permissions.

Schema: Default Permission Scheme

Issue:

Notifications

JTest can notify the appropriate people of particular events in your project, e.g. "Issue Commented". You can choose specific people, groups, or roles to receive notifications.

Schema:

Email:

JTest

File JUnit URL Test URL Category None

Summary

- New Types
- Workflows
- Schemes
- People
- Permissions
- Notifications
- Settings

New Types

Keep track of different types of issues, such as bugs or tasks. Each issue type can be configured differently.

Schema: Default Issue Type Scheme

- ☐ New
- ☐ New Feature
- ☐ Bug (default)
- ☐ Improvement

Workflows

Schemes can follow processes that mirror your team's practices. A workflow defines the sequence of steps that an issue will follow, e.g. "To Progress", "Resolved".

Schema: Default Workflow Scheme

- ☐ you (default)

Schemes

Schemes allow you to arrange the fields to be displayed for an issue. Different schemes can be used when an issue is created, moved, added, or transitioned through a workflow.

Schema: Default Issue Type Scheme Scheme

- ☐ Default Scheme Scheme (default)
- ☐ JTest Scheme Scheme

People

Different issues can have different information fields. A field configuration defines how fields behave for the project, e.g. required/optional, hidden/visible.

Schema: System Default Field Configuration

- ☐ Default Field Configuration (default)
- ☐ you (default)

Settings

Some general project configuration options.

- ☐ JUnit Modules
- ☐ Application Links
- ☐ Configuration Space: No instances available
- ☐ Subversion Repository: No instances available
- ☐ Maven Repository: None ()
- ☐ CVS Repository: No instances available
- ☐ Buildout Project: No instances available
- ☐ Reporting Path: None ()

People

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

Project Lead:

Versions

For software projects, JTest allows you to track different versions, e.g. 1.0, 2.0. Issues can be assigned to versions. This project has no associated versions.

Components

Projects can be broken down into components, e.g. "Database", "User Interface". Issues can then be categorized against different components. This project does not use any components.

Permissions

Project permissions allow you to control who can access your project, and what they can do, e.g. "Write an issue". Access to individual issues is granted to people by issue permissions.

Schema: Default Permission Scheme

Issue:

Notifications

JTest can notify the appropriate people of particular events in your project, e.g. "Issue Commented". You can choose specific people, groups, or roles to receive notifications.

Schema:

Email:



Welcome to the JIRA setup process



Step 1 of 4: Database Configuration

Database type: ☒ MySQL
The Java database driver is included in the JIRA installation and no additional configuration is required to use MySQL. For more information, see the MySQL documentation.

☐ PostgreSQL
The Java database driver is included in the JIRA installation and no additional configuration is required to use PostgreSQL. For more information, see the PostgreSQL documentation.

☐ Microsoft SQL Server
The Java database driver is included in the JIRA installation and no additional configuration is required to use Microsoft SQL Server. For more information, see the Microsoft SQL Server documentation.

Choose Language

