

# Open API Changes

## Changes from 6.0 to 6.5

- Classes `j.b.serverSide.TestBlockBean`, `j.b.serverSide.TestInProject`, `j.b.serverSide.FailedTestBean`, `j.b.TestNameBean` are removed from the Open API. Interfaces `j.b.serverSide.STest`, `j.b.serverSide.STestRun` should be used instead.
- Classes `j.b.tests.TestName` and `j.b.tests.SuiteTestName` are combined together into `j.b.tests.TestName`.

## Changes from 5.1.2 to 6.0

- `j.b.vcs.TriggerRules` class was removed from Open API as part of API cleanup. Please let us know if your plugin is affected by the change.

New responsibility event methods added:

- `j.b.serverSide.BuildServerListener.responsibleChanged(SProject, Collection<SuiteTestName>, ResponsibilityEntry, boolean)`.
- `j.b.notification.Notificator.notifyResponsibleChanged(Collection<SuiteTestName>, ResponsibilityEntry, SProject, Set<SUser>)`, `j.b.notification.Notificator.notifyResponsibleAssigned(Collection<SuiteTestName>, ResponsibilityEntry, SProject, Set<SUser>)`.
- `j.b.notification.NotificationEventListener.responsibleChanged(SProject, Collection<SuiteTestName>, ResponsibilityEntry, boolean)`.
- `j.b.messages.ServiceMessageTranslator` is reworked to allow binding to arbitrary message type by name instead of only known types

Most methods in `j.b.agent.AgentLifecycleListener` interface were extended to receive `j.b.agent.BuildRunnerContext`.

`j.b.agent.AgentLifecycleListener#runnerFinished(...)` method added. It is called after build step is finished.

`j.b.agent.duplicates.DuplicatesReporter` and `j.b.duplicator.DuplicateInfo` are added for reporting code duplicates on agent side.

## Build Agent changes:

- `j.b.agent.AgentRunningBuild` does not extend `j.b.agent.AgentBuildInfo`, `j.b.agent.ResolvedParameters`. All methods from those interfaces were inlined into `AgentRunningBuild` interface.

Most methods from `j.b.agent.AgentRunningBuild` were splitted into `j.b.agent.BuildRunnerContext` and `j.b.agent.BuildContext`. We have added

Parameters required for build runner are represented with `j.b.agent.BuildRunnerContext` interface.

Every time `AgentRunningBuild` and `BuildRunnerContext` return resolved parameters back.

`j.b.agent.BuildRunnerContext` represents the context of current build runner. All `add*` methods modifies context for the runner. Those changes will be reverted when context is switched to next runner.

`j.b.agent.AgentRunningBuild` provides a context of a build (i.e. shared between all runners). All `addShared*` methods modifies the build context (and thus all build runner contexts).

`j.b.agent.BuildAgentConfiguration` now contains `getBuildParameters()` and `getConfigParameters()` methods to access parameters. Configuration parameters here are formed from properties from `buildAgent.properties` that does not start from 'system.' or 'env.' prefix. All parameters are returned with all references resolved.

`j.b.agent.AgentBuildRunner#createBuildProcess` method signature has been changed to receive `j.b.agent.BuildRunnerContext`.

`j.b.agent.CommandLineBuildService#initialize(...)` method signature has been changed to receive `j.b.agent.BuildRunnerContext`.

`j.b.agent.CommandLineBuildService#getRunnerContext(...)` added

`j.b.agent.CommandLineBuildService#afterProcessSuccessfullyFinished()` added

`j.b.agent.BuildServiceAdapter` is added to simplify as proposed base class for commandline base build runner service.

## Changes from 5.0 to 5.1

Web extensions:

- deprecated method removed:  
`j.b.web.openapi.WebControllerManager.addPageExtension(final WebPlace addTo, final WebExtension extension, Anchor<WebExtension> anchor)`
- deprecated class removed: `j.b.serverSide.Anchor`
- deprecated class removed: `j.b.notification.TemplatePatternProcessor`; `j.b.notification.TemplateProcessor` added instead, see [Extending Notification Templates Model](#)
- method removed: `j.b.notification.TemplateMessageBuilder.setPatternProcessor()`
- several methods in `j.b.serverSide.SBuildType` now return `boolean` instead of `void`. You will probably need to recompile your plugins that use the interface.

## Changes from 4.5.5 to 5.0

### Parameters

`j.b.serverSide.parameters.AbstractBuildParameterReferencesProvider` is renamed to `j.b.serverSide.parameters.AbstractBuildParametersProvider`  
`j.b.serverSide.parameters.BuildParameterReferencesProvider` is renamed into `j.b.serverSide.parameters.BuildParametersProvider`  
`BuildParameterReferencesProvider.getParameters(@NotNull final SBuild build)` changed signature to `getParameters(@NotNull final SBuild build, final boolean emulationMode)`  
`j.b.agent.BuildAgentConfiguration#getCacheDirectory` now receives `String` as argument  
`j.b.serverSide.buildDistribution.StartBuildPrecondition#canStart` second parameters (`Map<QueuedBuildInfo, BuildAgent>`) may contain null values for some queued builds

### Miscellaneous

Added new build server events:

```
j.b.serverSide.BuildServerListener.vcsRootRemoved(SVcsRoot),  
j.b.serverSide.BuildServerListener.responsibleChanged(SProject, TestNameResponsibilityEntry,  
TestNameResponsibilityEntry, boolean)
```

Added three notification methods:

```
j.b.notification.Notificator.notifyResponsibleAssigned(SBuildType, Set<SUser>),  
j.b.notification.Notificator.notifyResponsibleChanged(TestNameResponsibilityEntry,  
TestNameResponsibilityEntry, SProject, Set<SUser>), j.b.notification.Notificator.notifyResponsibleAssigned(  
TestNameResponsibilityEntry, TestNameResponsibilityEntry, SProject, Set<SUser>)
```

## Changes prior to 4.5.5

Not documented