Who am I

Donát Csikós

Engineer @ Gradle

Tooling team

Contact me
email: donat@gradle.com
twitter: @donatcsikos
github: @donat
Timeline

- 2016.10: ECE presentation
- 2017.01: Buildship 2.0
- 2017.07: Buildship 2.1
- 2017.12: Buildship 2.2
- 2018.11*: Buildship 3.0
  - Inclusion in Eclipse 2018-12
Project activities

- Part of the release train; installed by default in 3 packages
- Active community at discuss.gradle.org and github.com/eclipse/buildship
- Marketplace: around 9K/month, 16th most installed plugin
- Update site: 600,000 installation in September
Hierarchical preferences

- Top-level: workspace configuration
- Build configuration can either inherit or override workspace level
- Run configuration can either inherit or override build level
Runtime classpath separation

- The was no classpath separation in Eclipse
- Main and test source set merged
- Blocker for Spring Boot tooling
- Solution: custom runtime classpath provider
- Upcoming: https://github.com/eclipse/buildship/issues/689
Build scans

- Recording of a build providing insights
- Analyze environment, executed tasks, performance, etc.
- Free central service
- On-premise enterprise version with extra features
Build scans integration

- Convenience features
  - Auto-publish
  - Open action on toolbar
  - Add links to console output
Import broken projects

- Root project always imported
- Error markers on
  - line number
  - build script
  - root project
Public APIs

- Based on community feedback: [https://github.com/eclipse/buildship/issues/676](https://github.com/eclipse/buildship/issues/676)
- Let Buildship extensions
  - Execute tasks
  - Synchronize Gradle projects
Project synchronization

- “My plugin wants to import Gradle projects”
- New API lets clients
  - Import new projects
  - Synchronize existing projects
GradleWorkspace workspace = GradleCore.getWorkspace();
BuildConfiguration configuration = BuildConfiguration
  .forRootProjectDirectory(new File("path/to/project"))
  .overrideWorkspaceConfiguration(true)
  .gradleDistribution(GradleDistribution.forVersion("4.10.2"))
  .autoSync(true)
  .build();
GradleBuild newBuild = workspace.createBuild(configuration);
newBuild.synchronize(monitor);
IPrj ect project = ...
GradleBuild existingBuild = workspace.getBuild(project).get();
eexistingBuild.synchronize(monitor);
Task execution API

- "My plugin needs to execute Gradle tasks"
  - Technically possible with Tooling API
- Convenience API
  - `GradleBuild.withConnection()`
  - Input, outputs, progress reporting, cancellation automatically hooked in the IDE
  - Use-cases documented in the javadoc

```java
File projectRoot = new File("/path/to/root")

ProjectConnection connection =
    GradleConnector.newConnector()
    .forProjectDirectory(projectRoot)
    .connect();

connection.newBuild()
    .forTasks("build")
    .setStandardOutput(System.out)
    .run();
```
Task execution API

```java
IPreject project = ...
GradleWorkspace workspace = GradleCore.getWorkspace();
GradleBuild build = workspace.getBuild(project).get();
build.withConnection(connection -> {
    connection.newBuild().forTasks("build").run();
}, monitor)
```
Project configurators

- “My plugin wants to add configure additional tooling support for all Gradle projects”
- Extension point
- Yet to come: configurator ordering

```java
public interface ProjectConfigurator {
    void init(InitializationContext context, IProgressMonitor monitor);
    void configure(ProjectContext context, IProgressMonitor monitor);
    void unconfigure(ProjectContext context, IProgressMonitor monitor);
}
public interface InitializationContext {
    GradleBuild getGradleBuild();
}
public interface ProjectContext {
    IProject getProject();
}
```
3.1.0+
<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Status</th>
<th>Author</th>
<th>Labels</th>
<th>Projects</th>
<th>Milestones</th>
<th>Assignee</th>
<th>Sort by</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Run tasks on synchronization</td>
<td>An:enhancement</td>
<td>Bugzilla</td>
<td>Icebox</td>
<td></td>
<td></td>
<td></td>
<td>Newest</td>
</tr>
<tr>
<td>#2</td>
<td>Run tasks when Eclipse build is triggered</td>
<td>An:enhancement</td>
<td>Bugzilla</td>
<td>Icebox</td>
<td></td>
<td></td>
<td></td>
<td>Oldest</td>
</tr>
<tr>
<td>#3</td>
<td>Support debugging tests</td>
<td>An:enhancement</td>
<td>Bugzilla</td>
<td>Icebox</td>
<td></td>
<td></td>
<td></td>
<td>Most commented</td>
</tr>
<tr>
<td>#4</td>
<td>Annotation Processor configuration from Gradle Eclipse Plugin</td>
<td>An:new-feature</td>
<td>Bugzilla</td>
<td>Icebox</td>
<td></td>
<td></td>
<td></td>
<td>Least commented</td>
</tr>
<tr>
<td>#5</td>
<td>Provide an Eclipse Preference API</td>
<td>An:enhancement</td>
<td>Bugzilla</td>
<td>Icebox</td>
<td></td>
<td></td>
<td></td>
<td>Recently updated</td>
</tr>
<tr>
<td>#6</td>
<td>Add the &quot;test&quot; classpath attribute to test sources and dependencies</td>
<td>An:enhancement</td>
<td>Bugzilla</td>
<td>Icebox</td>
<td></td>
<td></td>
<td></td>
<td>Least recently updated</td>
</tr>
<tr>
<td>#7</td>
<td>&quot;Cannot run tasks for included builds&quot;</td>
<td>-</td>
<td>Fbricon</td>
<td>Backlog</td>
<td></td>
<td></td>
<td></td>
<td>Most reactions</td>
</tr>
<tr>
<td>#8</td>
<td>Add preferences to show Console and/or Executions view when launching a build</td>
<td>An:enhancement</td>
<td>Help-wanted</td>
<td>Icebox</td>
<td></td>
<td></td>
<td></td>
<td>Most reactions</td>
</tr>
</tbody>
</table>
Wrap-up

- Try the latest Buildship snapshot: [github.com/eclipse/buildship/blob/master/docs/user/Installation.md](github.com/eclipse/buildship/blob/master/docs/user/Installation.md)
- Build scans: [scans.gradle.com](scans.gradle.com)
- Gradle Enterprise: [gradle.com/enterprise/get-started](gradle.com/enterprise/get-started)
- Reach out to us: [discuss.gradle.org](discuss.gradle.org)
- We’re hiring: [gradle.com/careers](gradle.com/careers)
Thank you!