An Introduction to Eclipse Che

Lets build a custom cloud IDE

October 2015
Tyler Jewell, Eclipse Che Project Lead
@TylerJewell

{ Codenvy

Codenvy

Goal

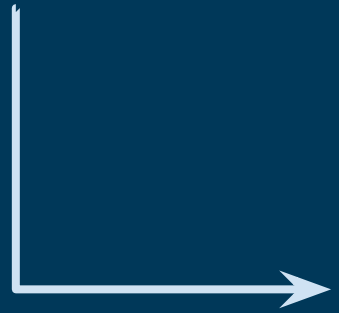
Let anyone contribute to any project anywhere at any time.

no pre-installation required
ready to participate in any branch flow
always compiles and runs

{ Codenvy

Goal

http://someurl/factory?id=a_project



1. Create new, or load existing, workspace
2. Populate workspace with 1..n projects
3. Clone, fetch, or pull source
4. Inject developer tools (ie, compiler, GitHub, etc)
5. Create runtime “machine” environment
6. Execute onLoad commands (ie, pre-compile)
7. Onboard developer into workspace

{ } Codenvy

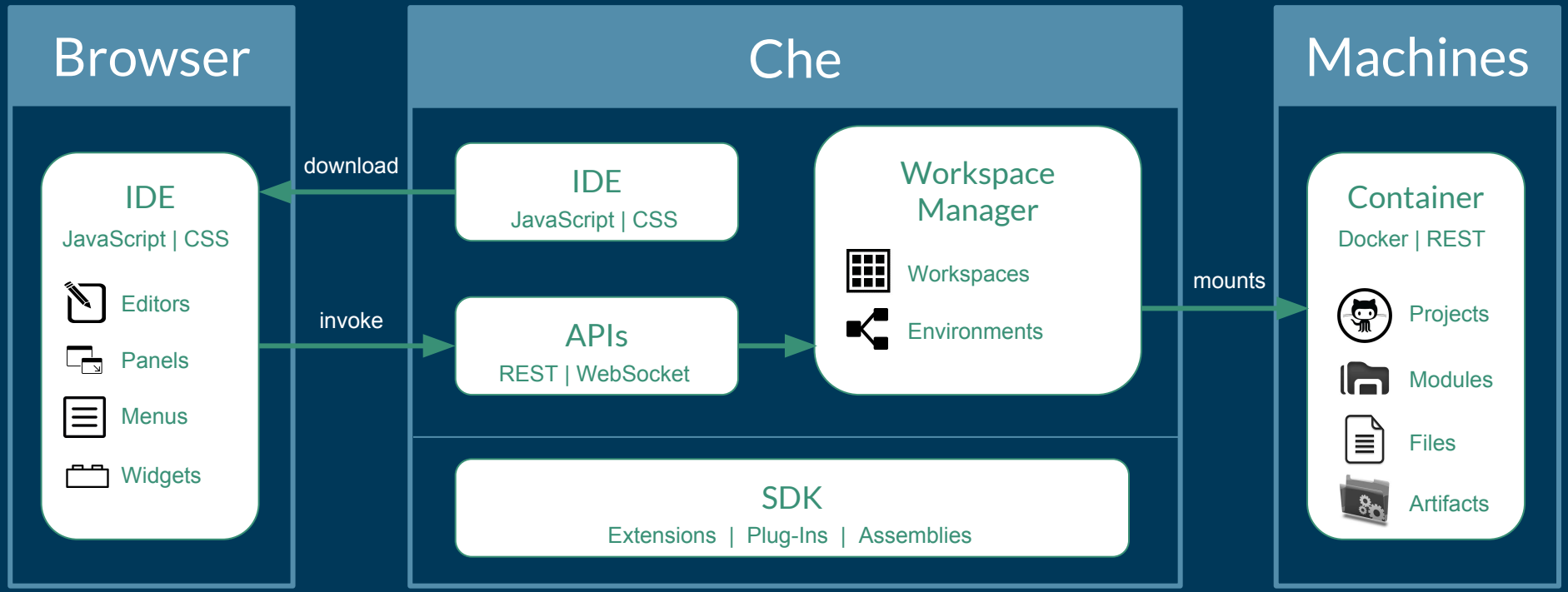

JIRA
Codenvy

Eclipse Che provides open source workspaces packaged as a cloud IDE installable on your desktop or server.

Use Che's built-in plug-ins or write your own to transform Che's IDE or server into new tools.

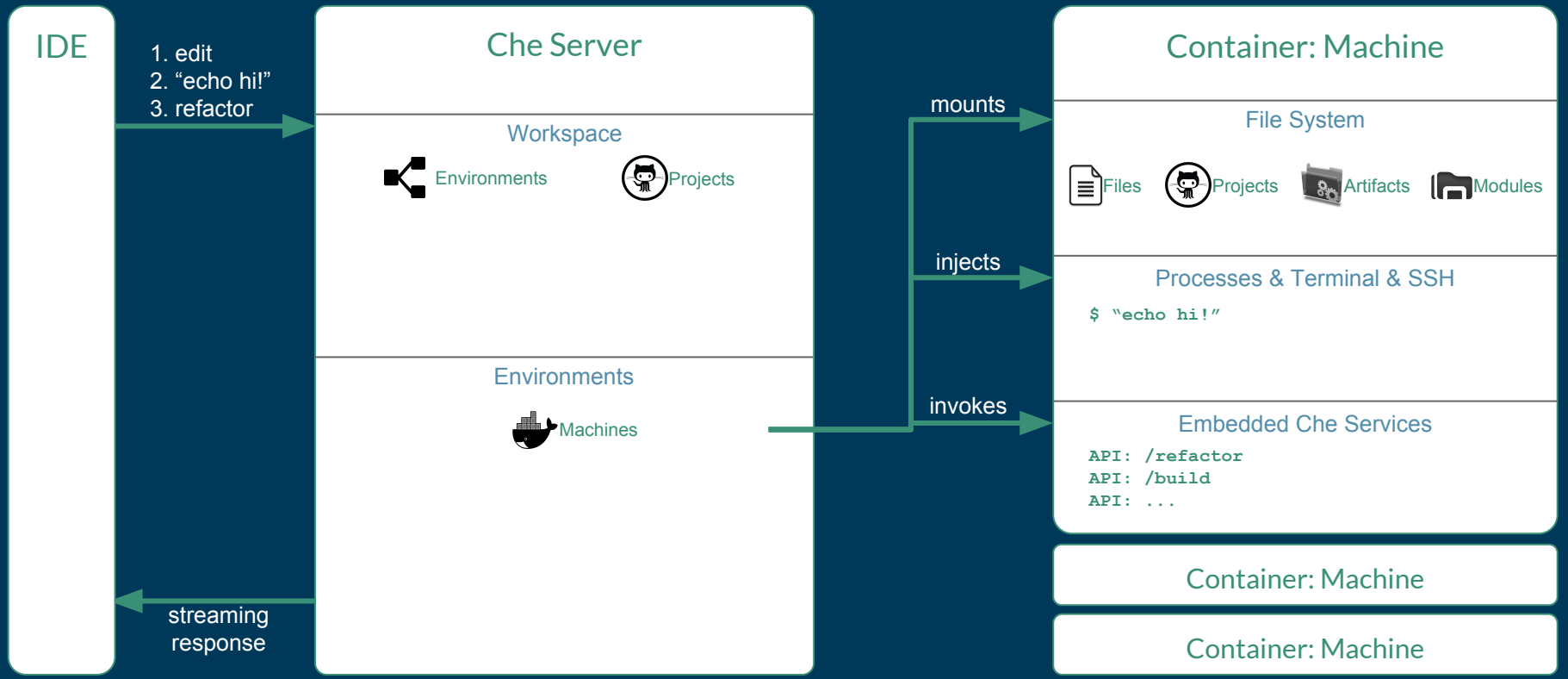
{ Codenvy

The Developer Workspace

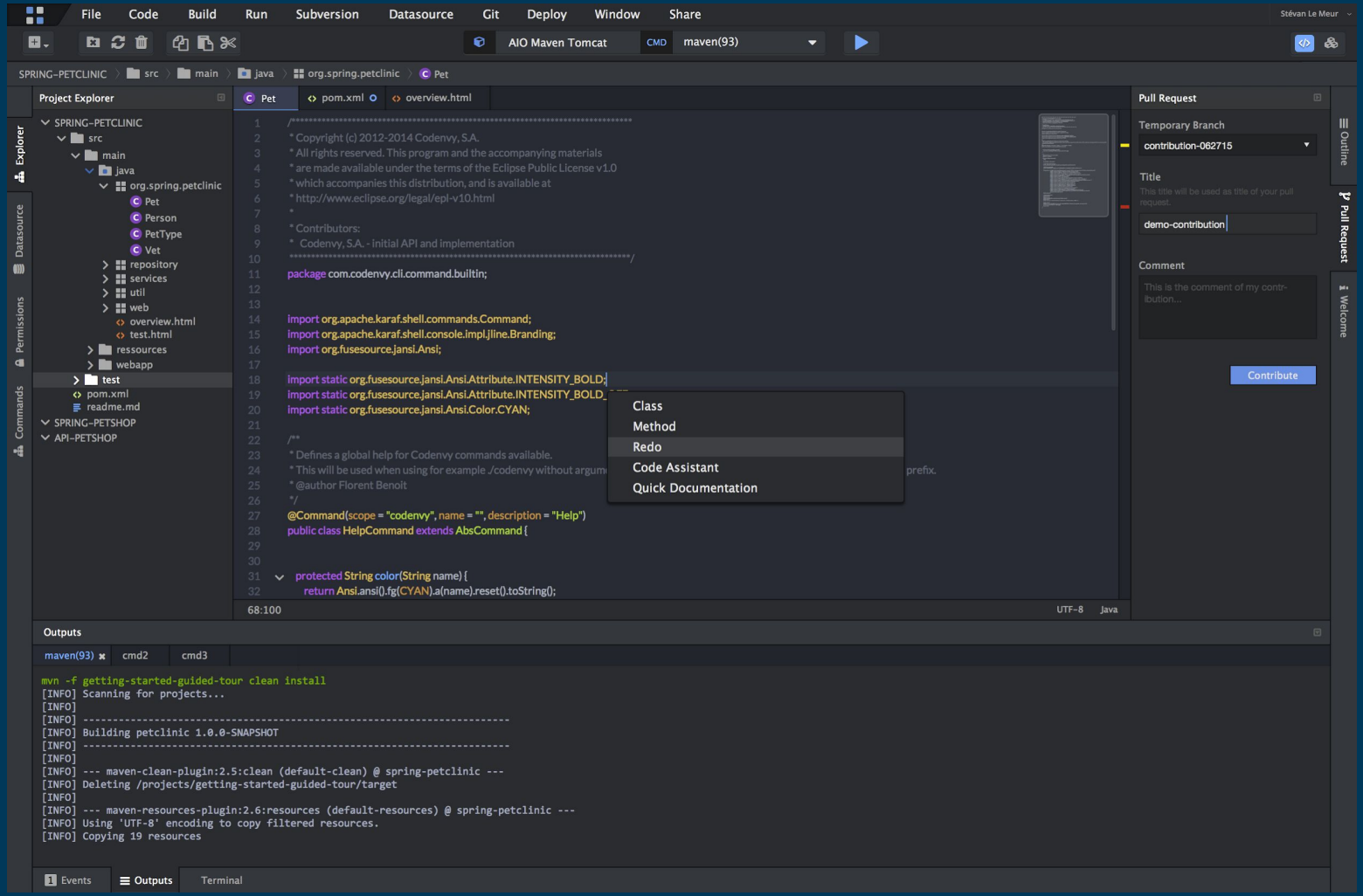


{ Codenvy

Containers Isolate Environments

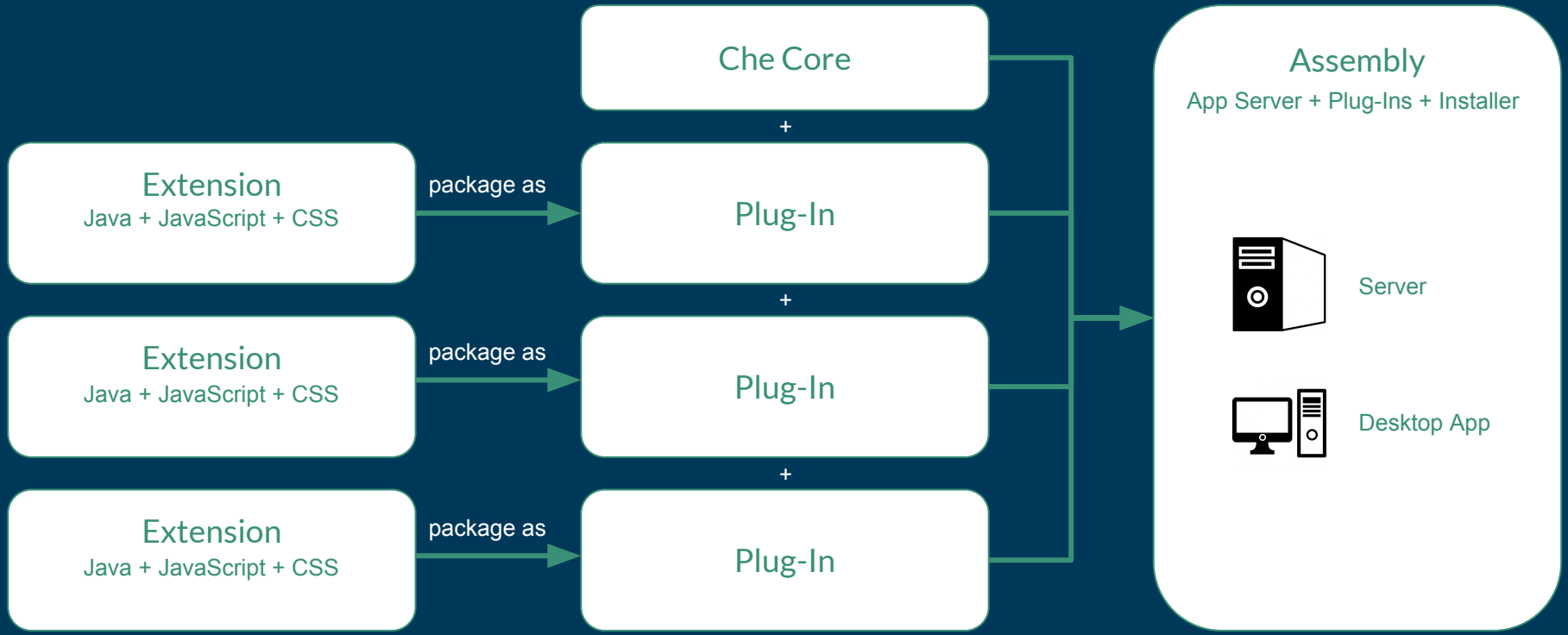


{ Codenvy

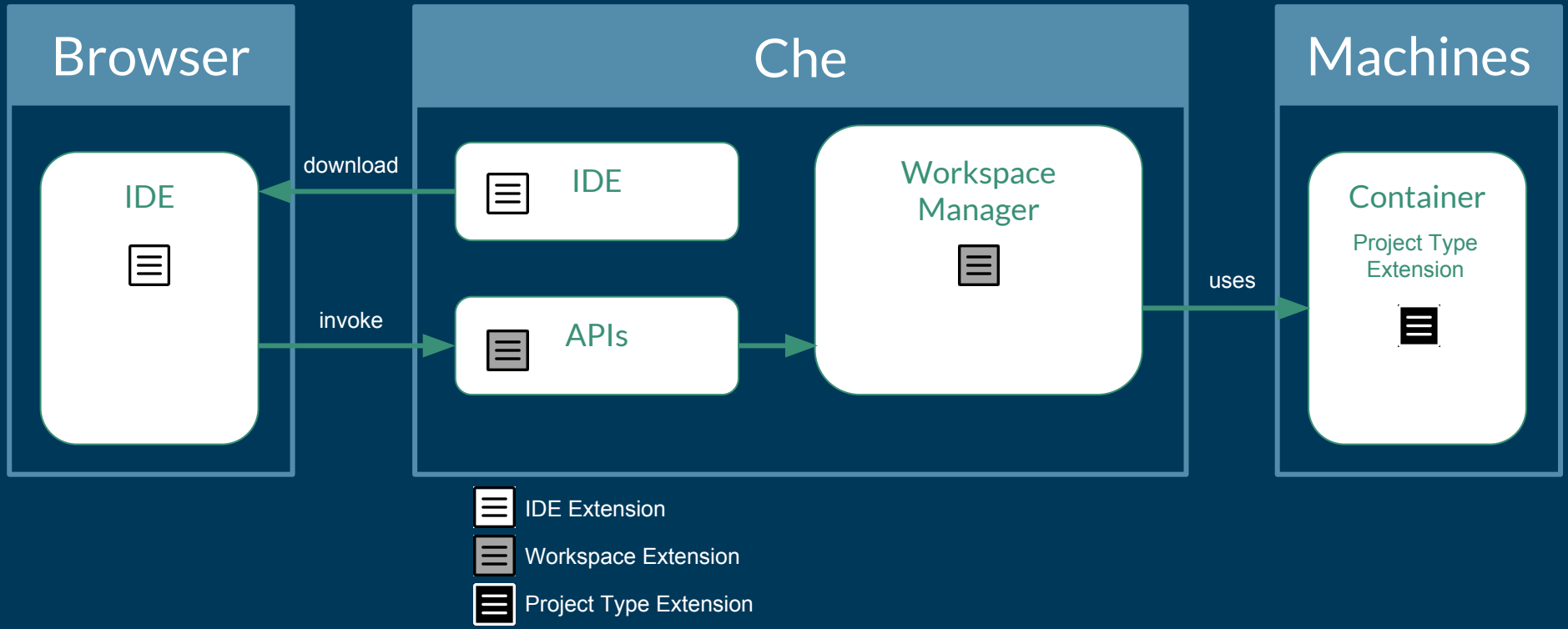


4.0 Demo - Java Refactor

Che Extensibility



Extensiblity Points



Let’s Build Our Own Plug-Ins

1. Install Che 3.13.1
2. Verify you can edit, build & run a project.
3. Add new plug-ins to Che & rebuild assembly.
4. Create new IDE & server extensions.
5. Package extensions as plug-ins.
6. Create drag-and-drop descriptions.

Getting Started with Che

Source:	github.com/codenvy/che
Developer List:	che-dev@eclipse.org
IRC:	#eclipseche (routes to slack)

Note: We are moving Che issues in Codenvy Jira to bugzilla.

git checkout 3.13.1

{ Codenvy

Cloning Che

```
git clone https://github.com/codenvy/che
git checkout 3.13.1
cd assembly-sdk
mvn clean install

# This will install che into /assembly-sdk/target directory
```

{ Codenvy

Plug-In Templates

```
# Community generator that creates new extensions from templates.
git clone https://github.com/stour/che-archetypes
mvn clean install

# Use the archetype generator to create a new extension from template.
# Do not run the generator in the che-archetypes source directory.
#
# che-client-extension-archetype = simple IDE extension with menu item
# che-multi-module-archetype     = single plug-in with multiple extensions
# che-project-type-archetype     = adds a new project type to Che
# che-server-extension-archetype = simple server-side extension with new REST API
mvn archetype:generate -DarchetypeCatalog=local
                       -DarchetypeGroupId=org.eclipse.che.archetype
                       -DinteractiveMode=false
                       -DarchetypeArtifactId=che-client-extension-archetype


# Compile the new extension.
mvn clean install
```

Codenvy

Che Extension

plugin-angularjs/core/client/src/main/java/com/codenvy/plugin/angularjs/core/client/

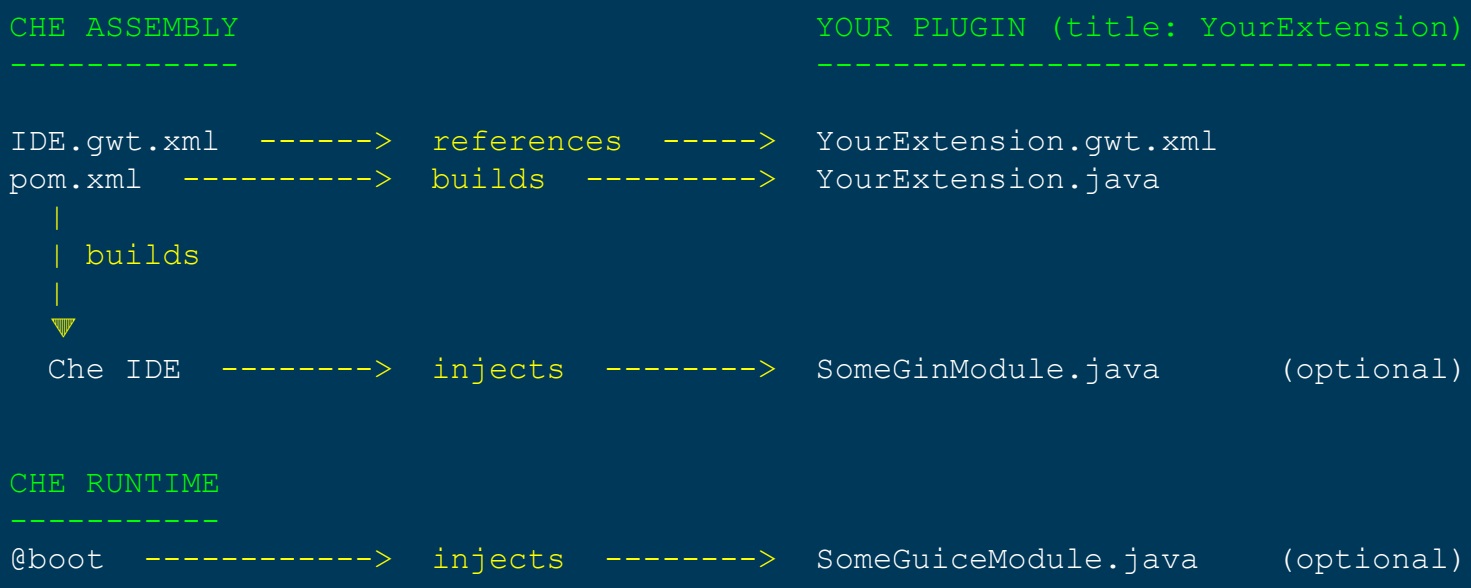
```
@Singleton
@Extension(title = "AngularJS")
public class AngularJsExtension extends JsExtension {
    @Inject
    public AngularJsExtension(IconRegistry iconRegistry, AngularJSResources resources) {
        super(Const.ANGULAR_JS_ID, iconRegistry, resources);
    }
}
```

plugin-angularjs/core/server/src/main/java/com/codenvy/plugin/angularjs/core/server/project/type

```
@Singleton
public class AngularJSProjectType extends ProjectType {
    public AngularJSProjectType() {
        super("AngularJS", "AngularJS Project", true, false);
        setDefaultRunner("system:/javascript/webapp/grunt");
        addRunnerCategories(Arrays.asList(RunnerCategory.JAVASCRIPT.toString()));
    }
}
```

Codenvy

Extension Structure



Codenvy

Eclipse Che

www.eclipse.org/che
Try it live: codenvy.com

{ Codenvy