The following has been tested against IntelliJ IDEA 2016.2.2

Steps

Within your locally cloned spring-framework working directory:

1. Precompile `spring-oxm` with `./gradlew :spring-oxm:compileTestJava`
2. Import into IntelliJ (File -> New -> Project from Existing Sources -> Navigate to directory -> Select build.gradle)
3. When prompted exclude the `spring-aspects` module (or after the import via File-> Project Structure -> Modules)
4. Code away

Known issues

1. `spring-core` and `spring-oxm` should be pre-compiled due to repackaged dependencies. See `*RepackJar` tasks in the build and https://youtrack.jetbrains.com/issue/IDEA-160605).
2. `spring-aspects` does not compile due to references to aspect types unknown to IntelliJ IDEA. See https://youtrack.jetbrains.com/issue/IDEA-64446 for details. In the meantime, the 'spring-aspects' can be excluded from the project to avoid compilation errors.
3. While JUnit tests pass from the command line with Gradle, some may fail when run from IntelliJ IDEA. Resolving this is a work in progress. If attempting to run all JUnit tests from within IntelliJ IDEA, you will likely need to set the following VM options to avoid out of memory errors: -XX:MaxPermSize=2048m -Xmx2048m -XX:MaxHeapSize=2048m
4. If you invoke "Rebuild Project" in the IDE, you'll have to generate some test resources of the `spring-oxm` module again (`./gradlew :spring-oxm:compileTestJava`)

Tips

In any case, please do not check in your own generated .iml, .ipr, or .iws files. You'll notice these files are already intentionally in .gitignore. The same policy goes for eclipse metadata.

FAQ

Q. What about IntelliJ IDEA's own Gradle support?

A. Keep an eye on https://youtrack.jetbrains.com/issue/IDEA-53476