

Adding a new feature/class to the project

Before creating your new feature branch in IntelliJ (to add a new class, e.g., CsvException), you **should pull the latest changes** from the remote (GitHub) repository first.

That ensures your local 'main' branch is up to date in case teammates have made commits since you last worked on it.

Here's a step-by-step guide when using the IntelliJ IDE

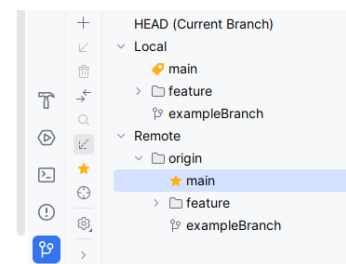
1. Update your local repo first

1. In IntelliJ, open the **Git** tool window. Look for the
Check the edge toolbar (bottom-right/left)
or
Menu → View → Tool Windows → Git



button.

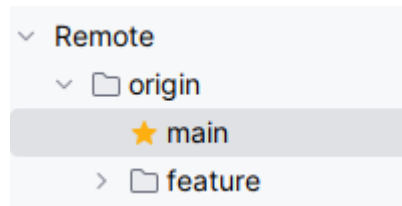
2. Switch to the **remote** branch you'll branch **from** (usually main):
 - Right-click the branch name.
 - Select main → **Checkout**.
3. Pull the latest commits:
 - Menu: **Git → Pull...**
 - Confirm the remote (usually origin/main).
 - IntelliJ will fetch and merge the newest changes.



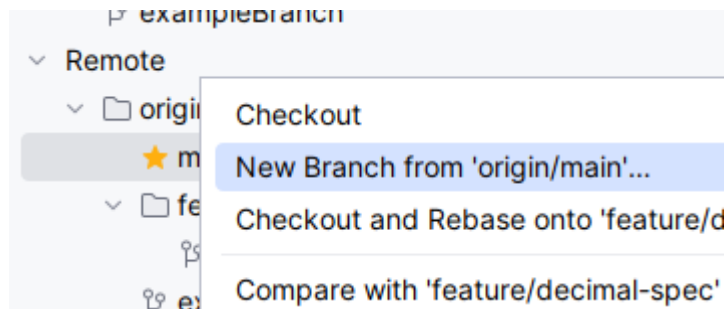
✅ *Now your base branch is up to date.*

2. Create a new feature branch

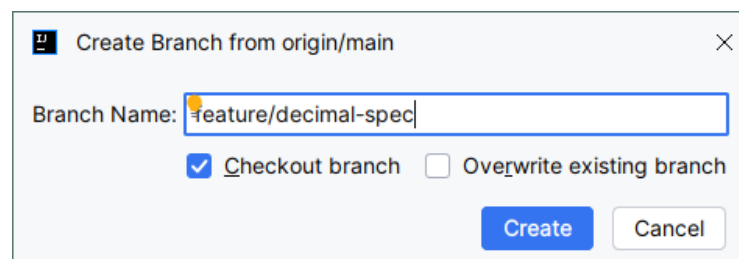
1. Right-click the **main** branch.



2. Choose **New Branch from origin/main...**

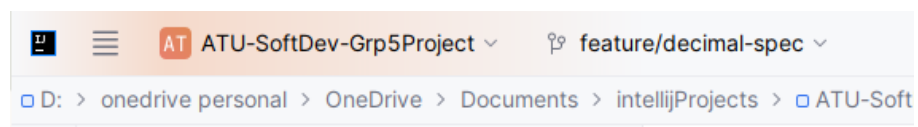


3. Enter a clear name, for example: feature/decimal-spec



4. Make sure “Checkout branch” is ticked so you immediately switch to it.
5. Press **Create**.

Now IntelliJ will show you’re on that branch.



3. Start working on your feature

- Create the new file: `src/main/java/.../DecimalSpec.java`.
 - Commit regularly with messages like:
 - `feat: add DecimalSpec for BigDecimal scale and rounding rules`
 - When done, push your branch:
Git → Push... → confirm `origin/feature/decimal-spec`.
-

4. Create a Pull Request (PR)

On GitHub (online):

- Open the repository.
 - You'll see a "Compare & pull request" banner for your new branch.
 - Fill in the description, link any related issues, and request reviews.
 - After the code review has been approved by a team member, the branch can be safely merged with the main branch.
-