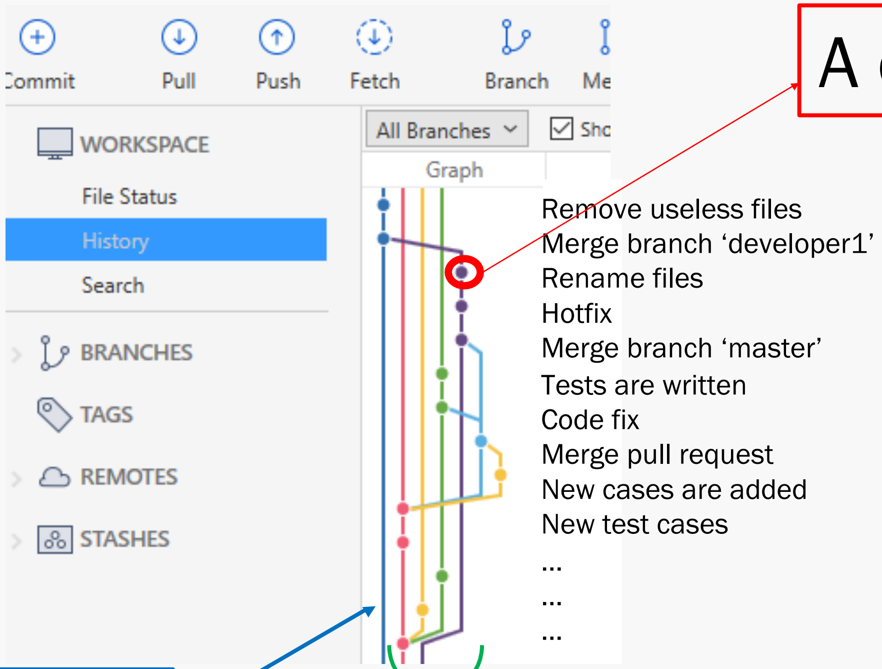
SOFTWARE ARCHITECTURES

Outline

- Using Git
  - *Branches*
  - *Pull, add, commit and push*
  - *Cloning a repository (GitHub and GitLab)*
- Breaking the ice with NetBeans IDE
  - *Run a web application*
  - *Working with Payara Server*
- The Best Practises of EJBs
  - *Stateless Session Bean*
  - *Statefull Session Bean*
  - *Singleton Session Bean*
- Data persistence
  - *Connecting, starting and creating a database*

Using Git

branches



Remove useless files
Merge branch ‘developer1’
Rename files
Hotfix
Merge branch ‘master’
Tests are written
Code fix
Merge pull request
New cases are added
New test cases
…
…
…

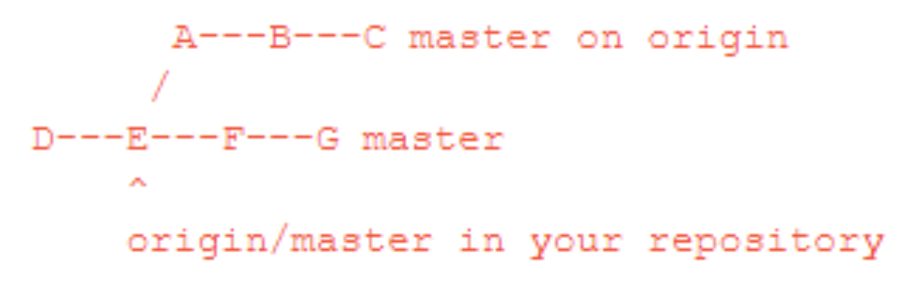
A commit

master branch

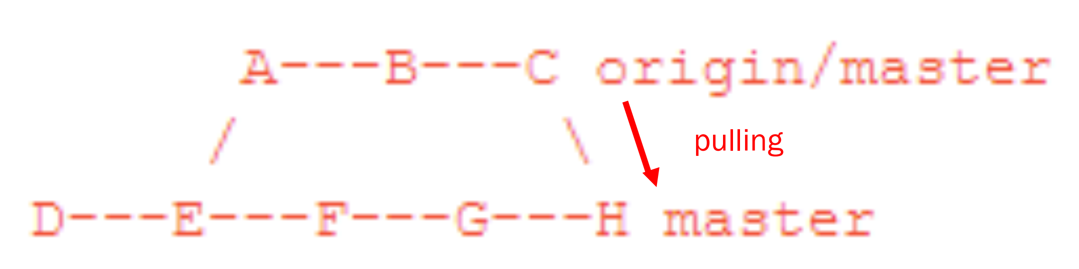
other branches

pull [1]

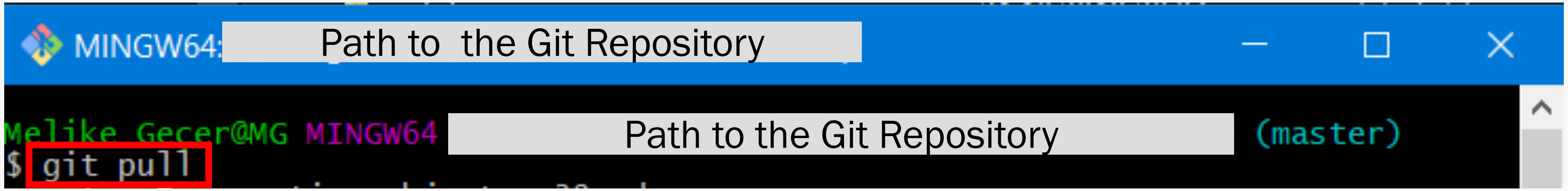
git pull - Fetch from and integrate with another repository or a local branch



```
      A---B---C master on origin
     /
D---E---F---G master
    ^
    origin/master in your repository
```



```
      A---B---C origin/master
     /         \
D---E---F---G---H master
```



```
Melike Gecer@MG MINGW64 Path to the Git Repository (master)
$ git pull
```

[1] https://git-scm.com/docs/git-pull

add [2], commit [3] and push [4]

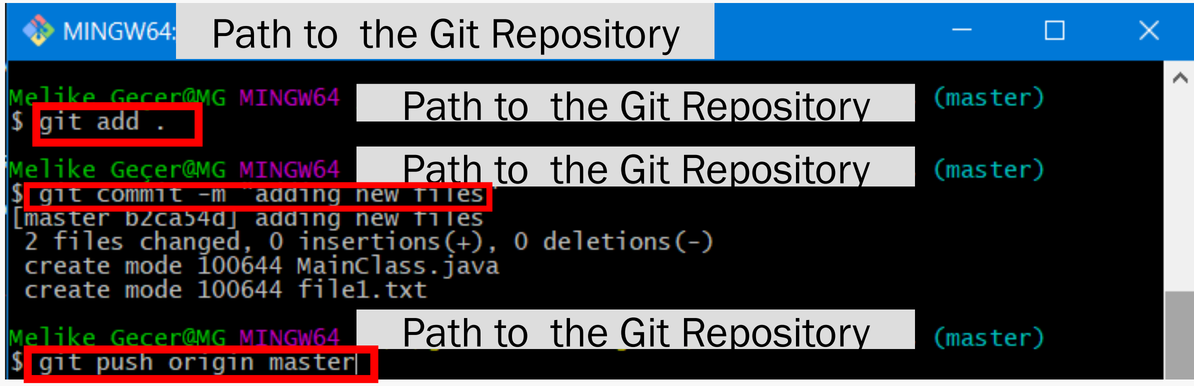
git commit - Record changes to the repository

git push - Update remote refs along with associated objects

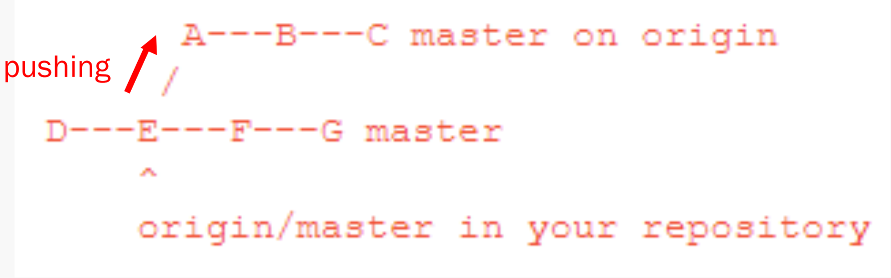
git add <file_name> [or git add .]

git commit –m <commit_message>

git push origin <branch_name>



```
Melike Geçer@MG MINGW64 Path to  the Git Repository (master)
$ git add .

Melike Geçer@MG MINGW64 Path to  the Git Repository (master)
$ git commit -m "adding new files"
[master b2ca54d] adding new files
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 MainClass.java
 create mode 100644 file1.txt

Melike Geçer@MG MINGW64 Path to  the Git Repository (master)
$ git push origin master
```

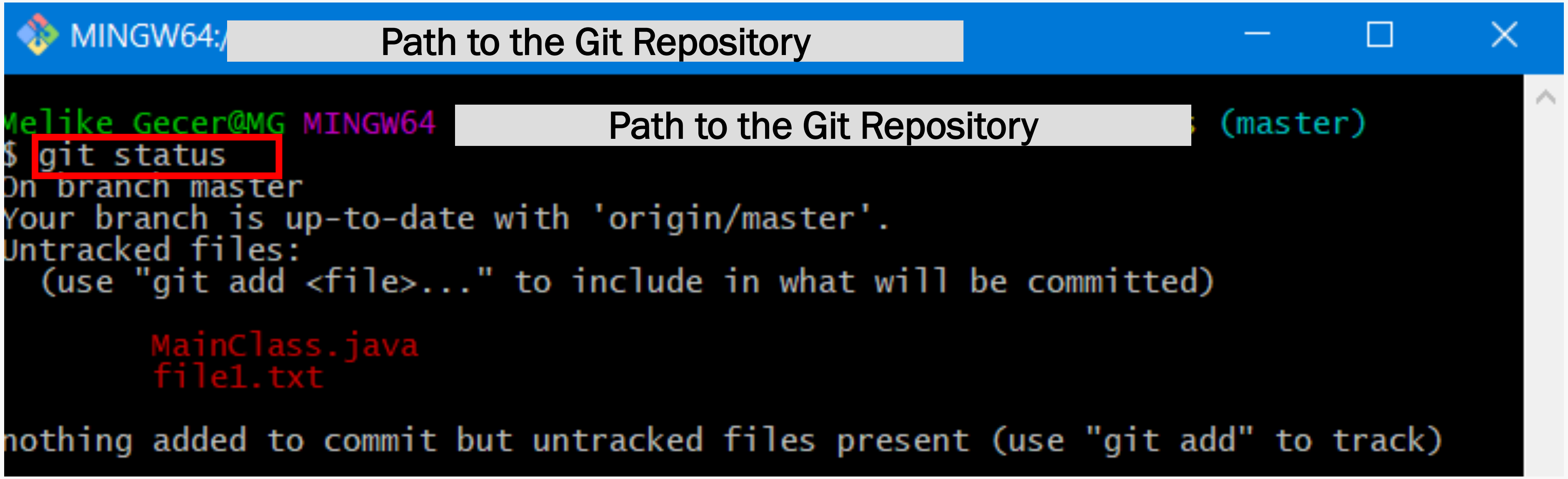


```
          A---B---C master on origin
         /
    D---E---F---G master
        ^
        origin/master in your repository
```

[2] https://git-scm.com/docs/git-add
[3] https://git-scm.com/docs/git-commit
[4] https://git-scm.com/docs/git-push

Other useful Git commands – status [5]

git status – Show the working tree status

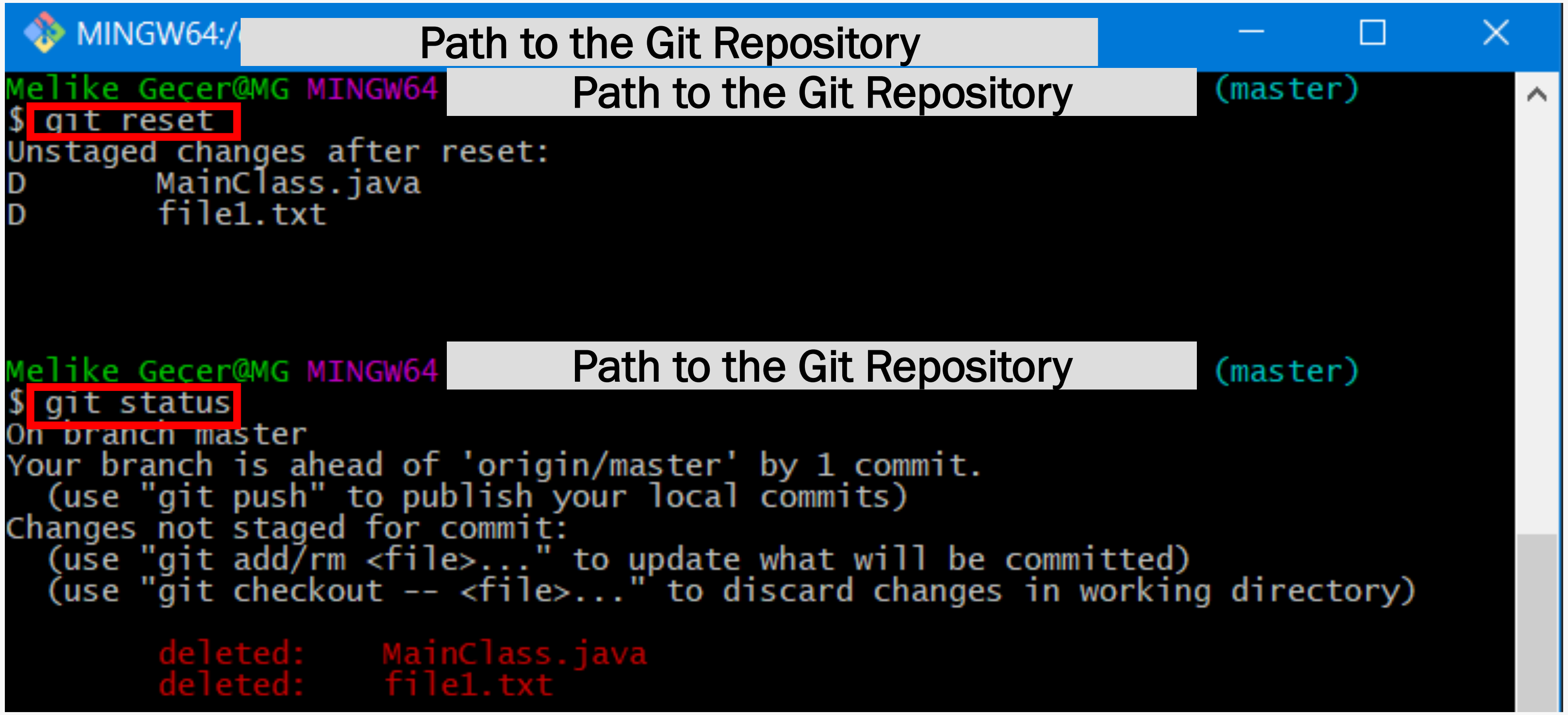


```
Melike Gecer@MG MINGW64 Path to the Git Repository (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        MainClass.java
        file1.txt

nothing added to commit but untracked files present (use "git add" to track)
```

[5] https://git-scm.com/docs/git-status

Other useful Git commands – reset [6]

git reset – Reset current HEAD* to the specified state



```
Melike Gecer@MG MINGW64 Path to the Git Repository (master)
$ git reset
Unstaged changes after reset:
D       MainClass.java
D       file1.txt



Melike Gecer@MG MINGW64 Path to the Git Repository (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    MainClass.java
        deleted:    file1.txt
```

[6] https://git-scm.com/docs/git-reset

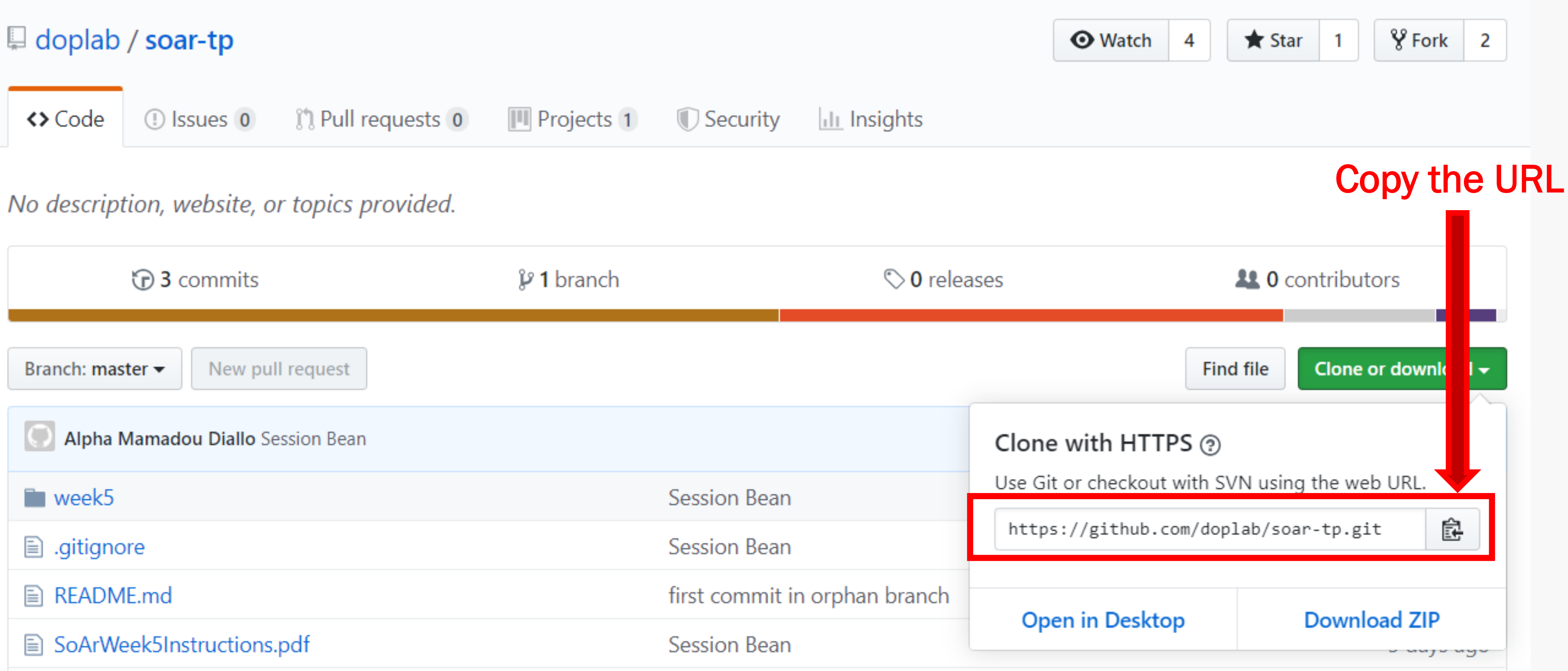
* HEAD is the current branch

Cloning a repository

1. Go to https://gitlab.unil.ch/users/sign_in
2. Login using your UniL account details (username and password)
3. Go to the given git repository URL. (i.e. https://github.com/doplab/soar-tp)

Cloning a repository

Copy the https:// clone URL of the given repository.



No description, website, or topics provided.

Alpha Mamadou Diallo Session Bean

week5	Session Bean
.gitignore	Session Bean
README.md	first commit in orphan branch
SoArWeek5Instructions.pdf	Session Bean

Clone with HTTPS

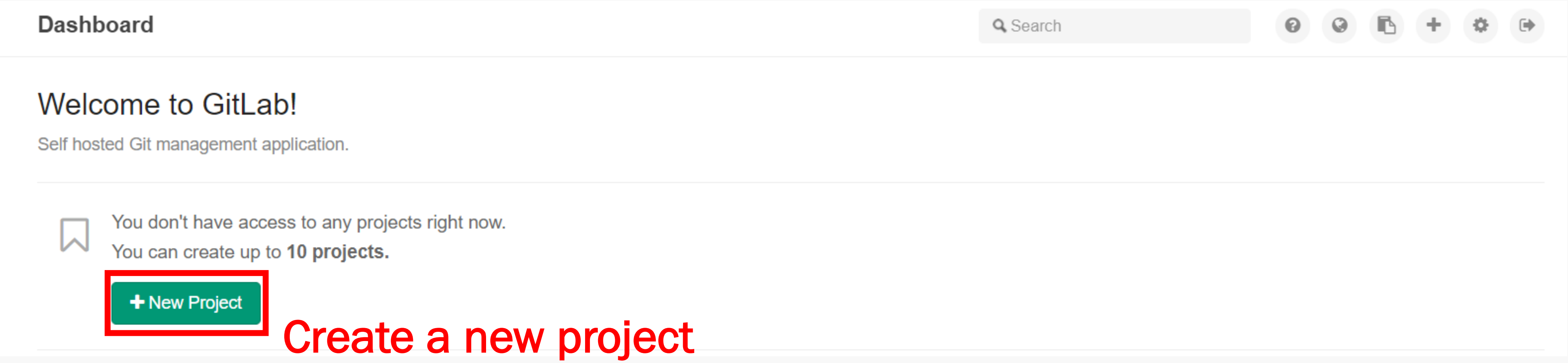
Use Git or checkout with SVN using the web URL.

```
https://github.com/doplab/soar-tp.git
```

Copy the URL

Cloning a repository

Create a new project on GitLab



Dashboard

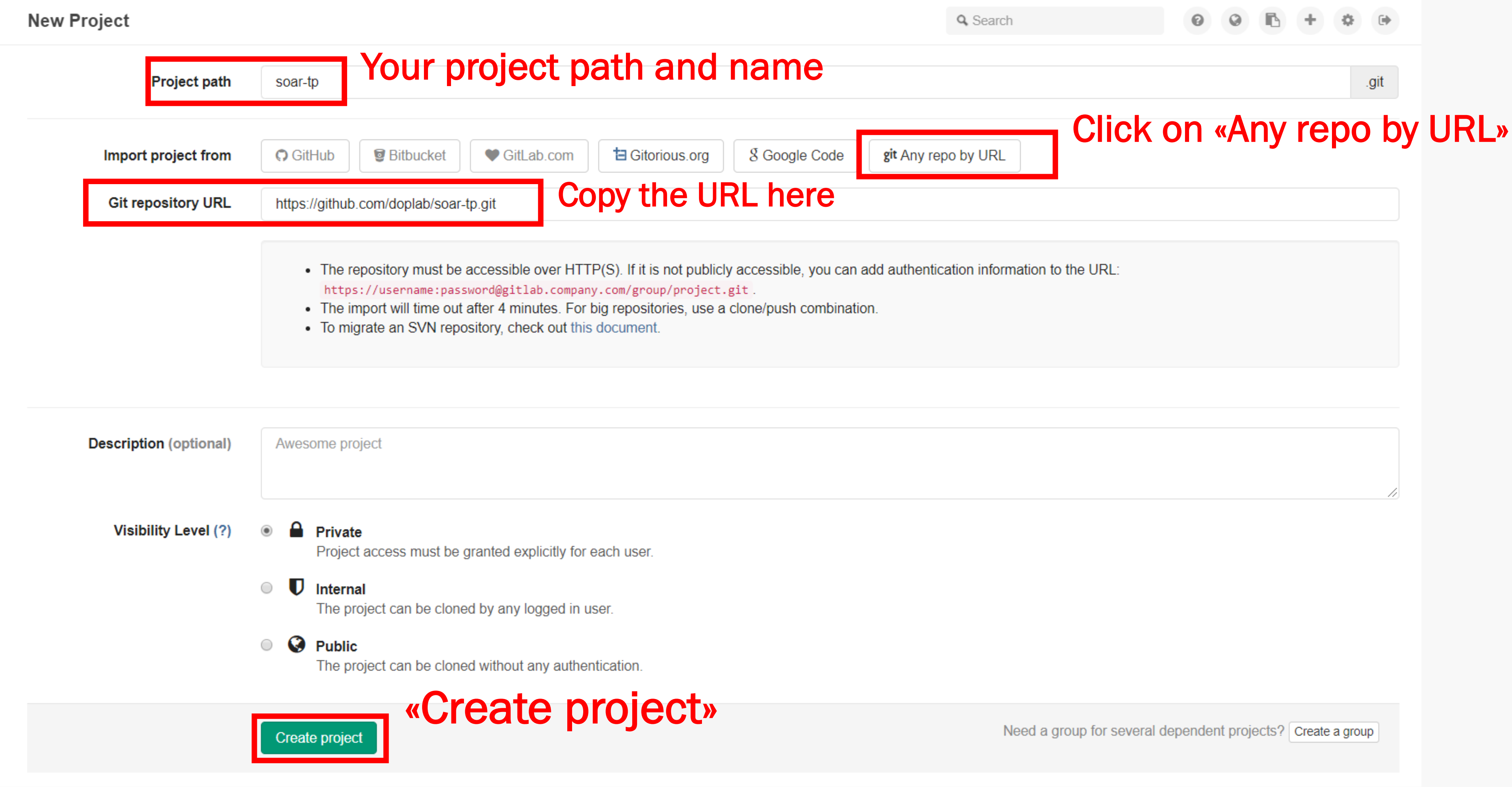
Welcome to GitLab!

Self hosted Git management application.

You don't have access to any projects right now.
You can create up to **10 projects.**

Create a new project

Cloning a repository



Project path: soar-tp .git

Your project path and name

Import project from: GitHub | Bitbucket | GitLab.com | Gitorious.org | Google Code | git Any repo by URL

Click on «Any repo by URL»

Git repository URL: https://github.com/doplab/soar-tp.git

Copy the URL here

- The repository must be accessible over HTTP(S). If it is not publicly accessible, you can add authentication information to the URL: `https://username:password@gitlab.company.com/group/project.git`.
- The import will time out after 4 minutes. For big repositories, use a clone/push combination.
- To migrate an SVN repository, check out this document.

Description (optional): Awesome project

Visibility Level (?)

● Private
Project access must be granted explicitly for each user.

○ Internal
The project can be cloned by any logged in user.

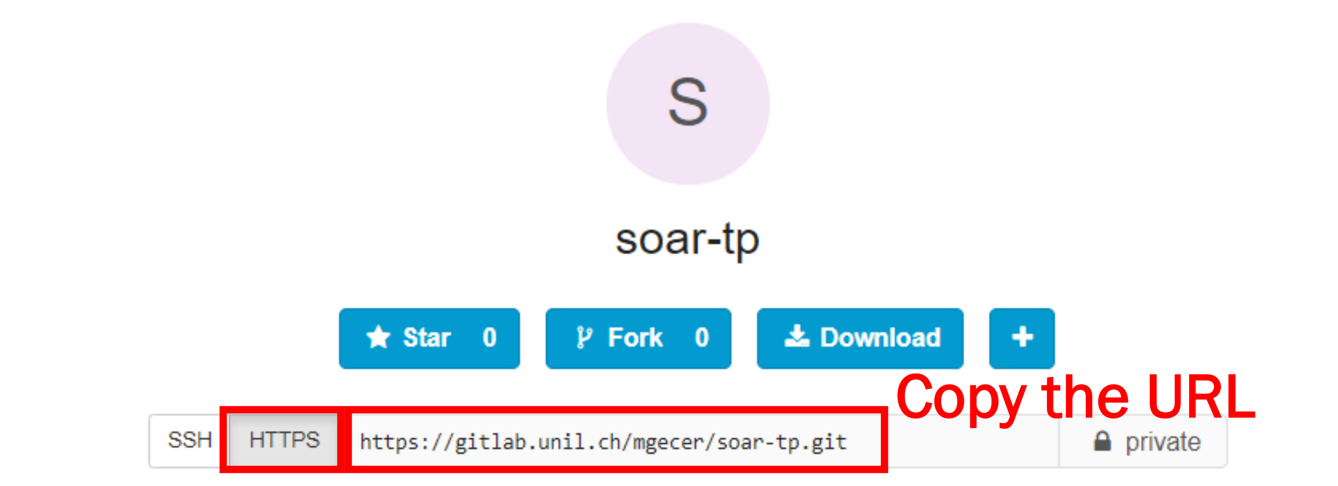
○ Public
The project can be cloned without any authentication.

Create project

«Create project»

Need a group for several dependent projects? Create a group

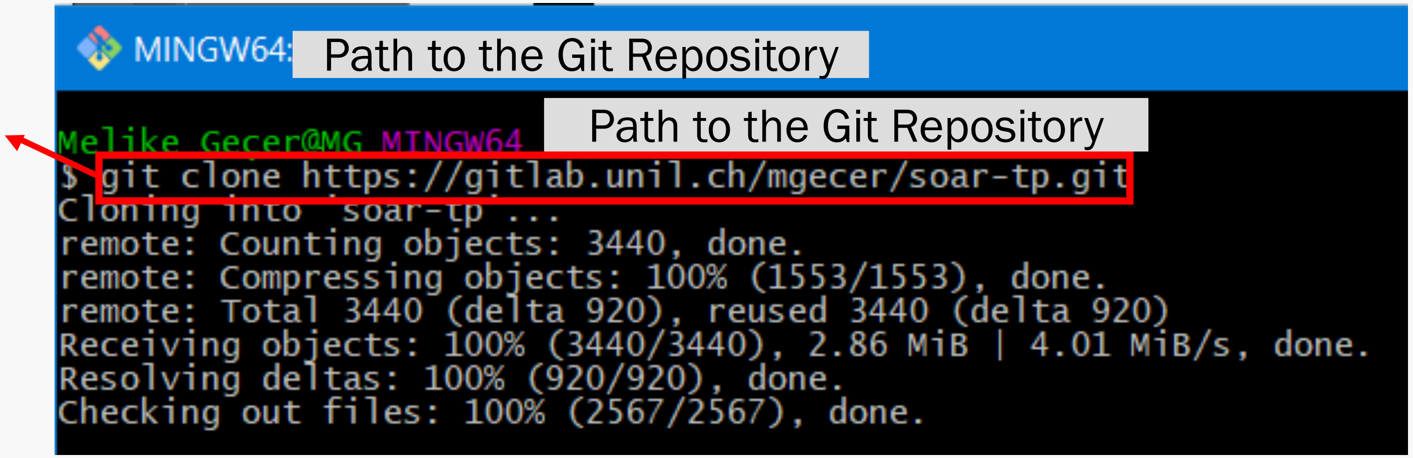
Cloning a repository



https://gitlab.unil.ch/mgecer/soar-tp.git

Copy the URL

Use
git clone https://...
to clone the repository from your GitLab account to your local machine.

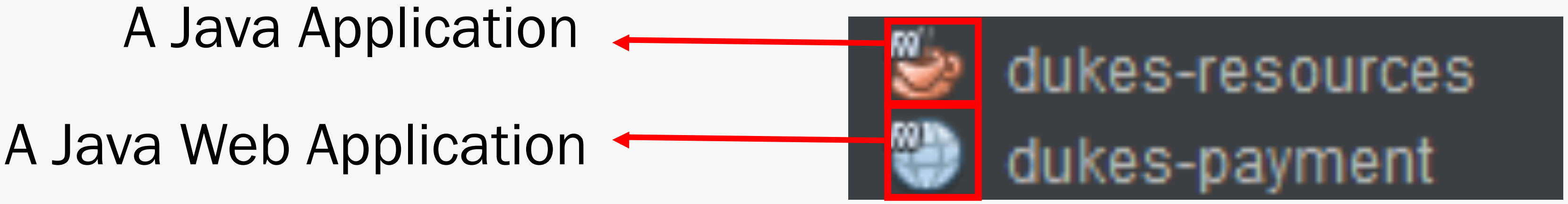


```
Melike Gecer@MG MINGW64   Path to the Git Repository
$ git clone https://gitlab.unil.ch/mgecer/soar-tp.git
Cloning into 'soar-tp'...
remote: Counting objects: 3440, done.
remote: Compressing objects: 100% (1553/1553), done.
remote: Total 3440 (delta 920), reused 3440 (delta 920)
Receiving objects: 100% (3440/3440), 2.86 MiB | 4.01 MiB/s, done.
Resolving deltas: 100% (920/920), done.
Checking out files: 100% (2567/2567), done.
```

Breaking the ice with NetBeans IDE

Run a Web Application

Unlike Java projects you've seen before, a Java web application **DOESN'T have a MainClass.**

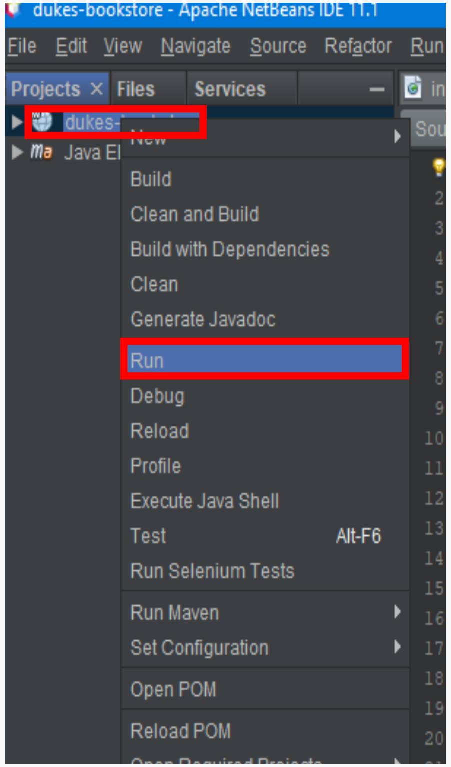


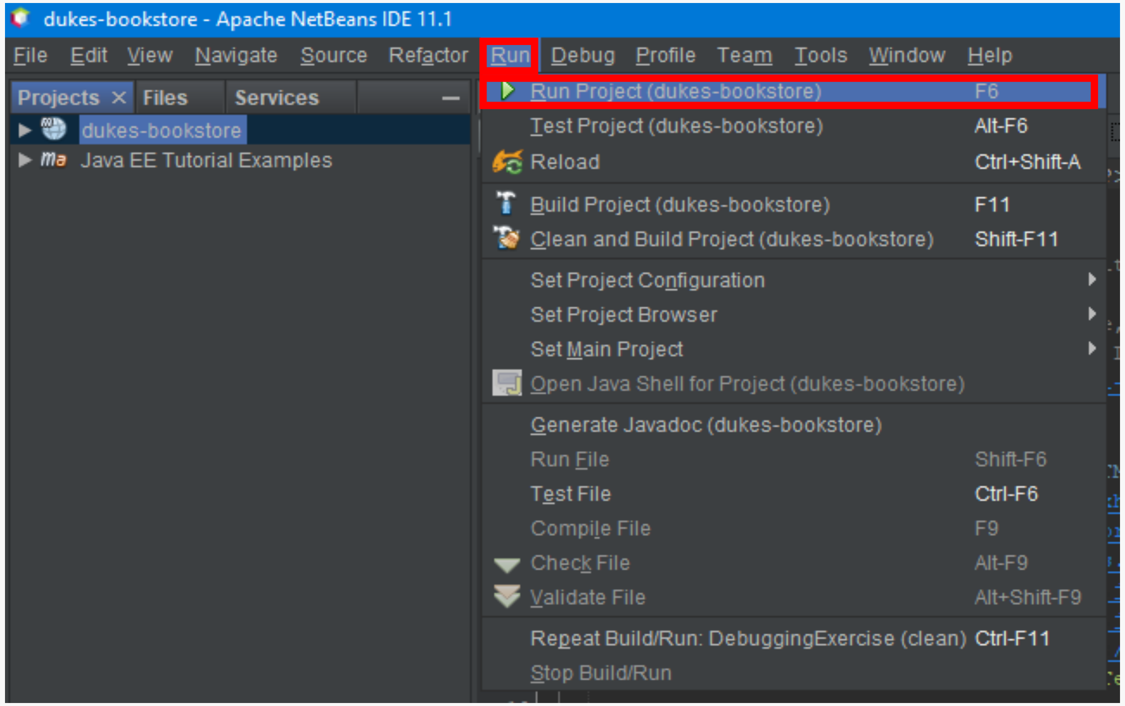
A Java Application

A Java Web Application

Run a Web Application

There are two ways to run a project;



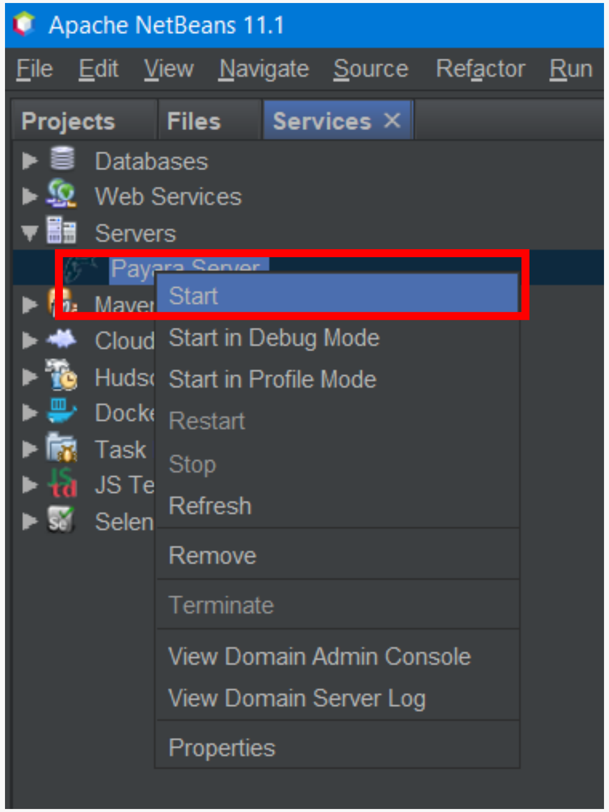

dukes-bookstore - Apache NetBeans IDE 11.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
dukes-bookstore
Java EE Tutorial Examples
Run Project (dukes-bookstore) F6
Test Project (dukes-bookstore) Alt-F6
Reload Ctrl+Shift-A
Build Project (dukes-bookstore) F11
Clean and Build Project (dukes-bookstore) Shift-F11
Set Project Configuration
Set Project Browser
Set Main Project
Open Java Shell for Project (dukes-bookstore)
Generate Javadoc (dukes-bookstore)
Run File Shift-F6
Test File Ctrl-F6
Compile File F9
Check File Alt-F9
Validate File Alt+Shift-F9
Repeat Build/Run: DebuggingExercise (clean) Ctrl-F11
Stop Build/Run

Right-click on the project and **RUN**

Click on **Run** tab and **Run Project**

Working with Payara Server

Web applications require a server to run. In exercise sessions, we will use the **Payara Server.**

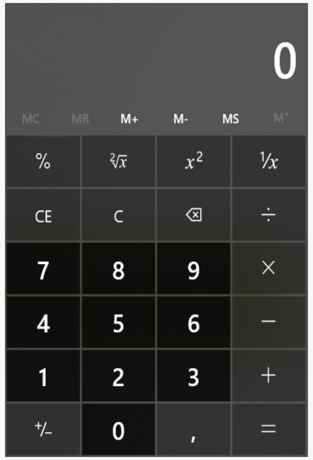


If you don’t have the **Payara Server** on NetBeans, please refer to this document.

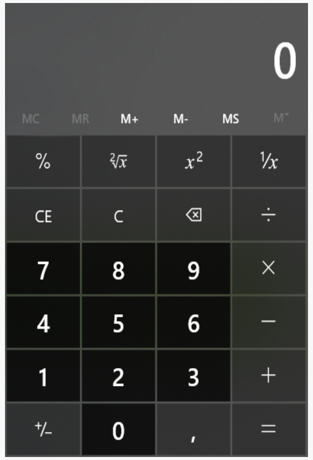
EJB
Enterprise JavaBeans

Stateless Session Bean [7]

1. When a client invokes the methods of a stateless bean, the bean’s instance variables may contain a state specific to that client but only for the duration of the invocation.
2. Except during method invocation, all instances of a stateless bean are equivalent, allowing the EJB container to assign an instance to any client.

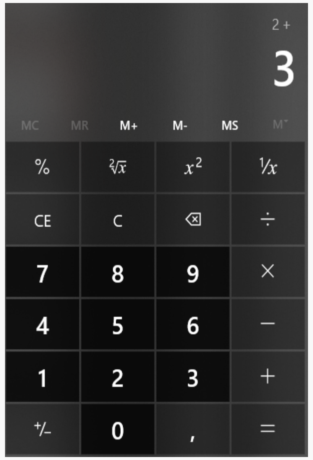


Client 1

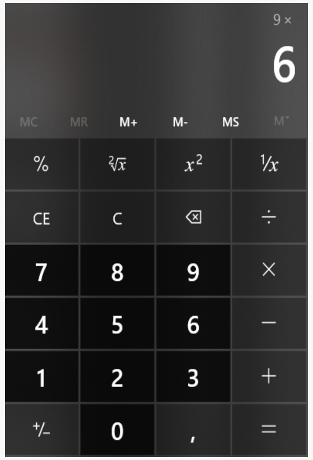


Client 2

Initial states



Client 1



Client 2

During method invocation

[7] https://docs.oracle.com/javaee/6/tutorial/doc/gipjg.html

Stateless Session Bean [7]

```
@Stateless
public class CalculatorBean {
    …
    public double add(double n1, double n2) {…}
    public double subtraction(double n1, double n2) {…}
    public double multiplication(double n1, double n2) {…}
    public double division(double n1, double n2) {…}
    …
}
```

Stateful Session Bean [7]

1. In a stateful session bean, the instance variables represent the state of a unique client/bean session.
2. A session bean is not shared; it can have only one client, in the same way that an interactive session can have only one user.

Client 1

Shopping cart #1
{}

Client 2

Shopping cart #2
{}

Initial states

Client 1

Shopping cart #1
{ book1, book2}

Client 2

Shopping cart #2
{book3, movie1,
movie2}

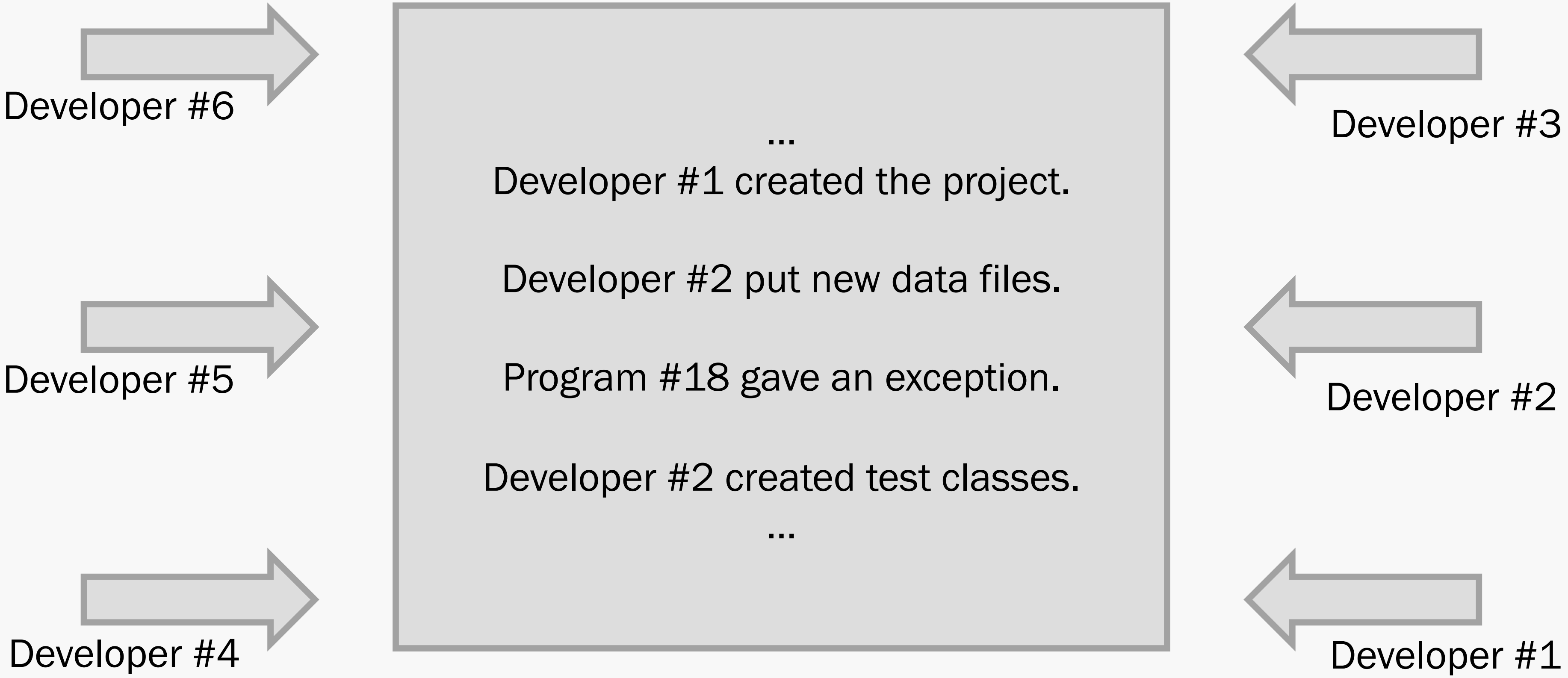
After few transactions

Stateful Session Bean [7]

```
@Stateful
public class CartBean {
    String customerId;
    String customerName;
    List<String> contents;
    …
    public void addBook(String title) {…}
    public void removeBook(String title) {…}
    public List<String> getContents() {…}
    …
}
```

Singleton Session Bean [7]

1. A singleton session bean is instantiated once per application and exists for the lifecycle of the application.
2. Singleton session beans are designed for circumstances in which a single enterprise bean instance is shared across and concurrently accessed by clients.



…
Developer #1 created the project.

Developer #2 put new data files.

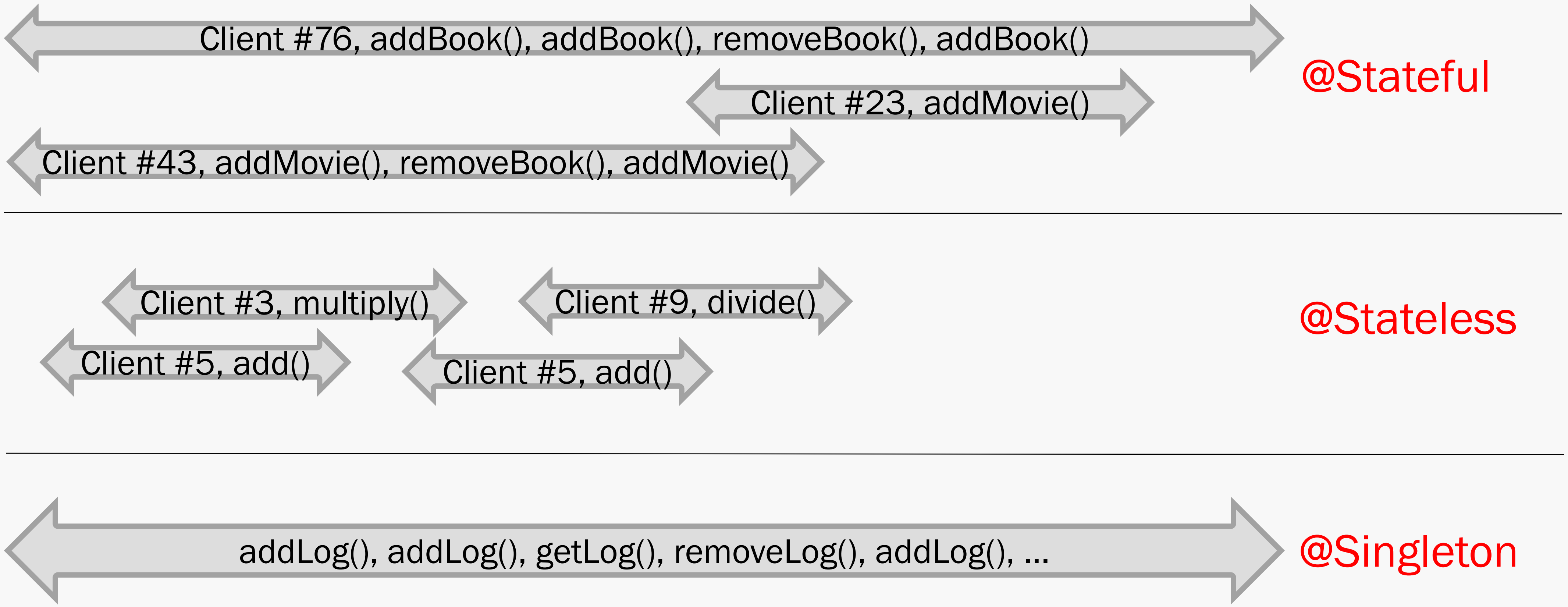
Program #18 gave an exception.

Developer #2 created test classes.
…

Singleton Session Bean [7]

```
@Singleton
public class LogBean {
    String logFileName;
    …
    public void addLog (String logMessage) {…}
    public void removeLog(int logID) {…}
    public String getLog(int logID) {…}
    …
}
```

The Lifecycles of Enterprise Beans



Client #76, addBook(), addBook(), removeBook(), addBook()

Client #23, addMovie()

Client #43, addMovie(), removeBook(), addMovie()

@Stateful

Client #3, multiply()

Client #9, divide()

Client #5, add()

Client #5, add()

@Stateless

addLog(), addLog(), getLog(), removeLog(), addLog(), ...

@Singleton