

Objectifs :

- La gestion de versions (Versioning).
- Découvrir le langage TypeScript.
- Premier pas avec Angular.

I. La gestion de versions (Versioning).

Choix :GitLab

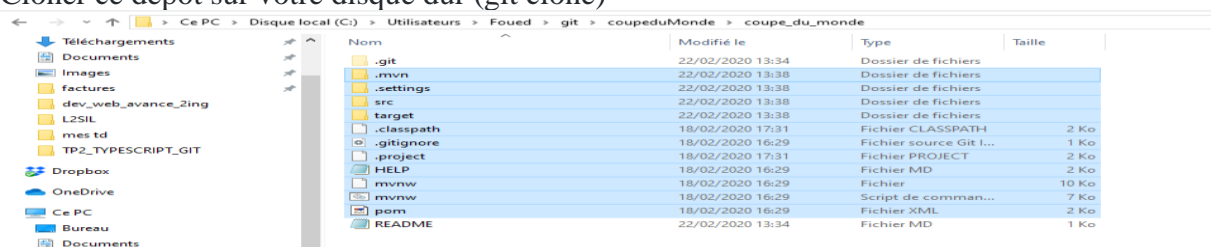
- a- Taper **git version** sur votre terminal pour vérifier que Git est bien installé sur votre machine. Télécharger depuis git-scm.com/downloads, sinon.
- b- Installer et/ou configurer git
Après l'installation de Git vous devez renseigner votre nom et votre adresse de courriel.

```
$ git config --global user.name "user name"
```

```
$ git config --global user.email votre_mail
```

- c- Créer un dépôt sur GitLab
 - Créer "Nouveau projet"
 - Le nom du projet sera aussi le nom du dépôt correspondant.
 - Copier l'adresse HTTPS fournie dans la page du dépôt

```
C:\Users\Foued\git\coupeduMonde>git config --global user.name "foued.oueslati"
C:\Users\Foued\git\coupeduMonde>git config --global user.email foued.oueslati@capit.fr
C:\Users\Foued\git\coupeduMonde>git clone https://gitlab.com/fwdwes/coupe_du_monde.git
Cloning into 'coupe_du_monde'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
C:\Users\Foued\git\coupeduMonde>
```

d- Cloner ce dépôt sur votre disque dur (git clone)

Nom	Modifié le	Type	Taille
.git	22/02/2020 13:34	Dossier de fichiers	
.mvn	22/02/2020 13:38	Dossier de fichiers	
.settings	22/02/2020 13:38	Dossier de fichiers	
src	22/02/2020 13:38	Dossier de fichiers	
target	22/02/2020 13:38	Dossier de fichiers	
.classpath	18/02/2020 17:31	Fichier CLASSPATH	2 Ko
.gitignore	18/02/2020 16:29	Fichier source Git I...	1 Ko
.project	18/02/2020 17:31	Fichier PROJECT	2 Ko
HELP	18/02/2020 16:29	Fichier MD	2 Ko
mvnw	18/02/2020 16:29	Fichier	10 Ko
mvnw	18/02/2020 16:29	Script de comman...	7 Ko
pom	18/02/2020 16:29	Fichier XML	2 Ko
README	22/02/2020 13:34	Fichier MD	1 Ko

- Créer un commit :
 - **\$ git add** .(rajouter tous les fichiers).
 - Copier votre projet dans le dépôt.
 - **\$ git status** (optionnel) pour constater qu'un fichier a été créé mais pas encore ajouté dans le dépôt(optionnel) pour afficher le contenu actuel de l'espace de staging (index)
 - **\$ git commit -m "lancement de projet"** pour créer un commit .
- e- **\$ git push** pour uploader votre commit sur votre dépôt distant, hébergé sur le serveur GitLab de l'EEMI.
- f- Le projet devrait maintenant être visible depuis la page web du dépôt, sur GitLab.

Maintenant, vous avez le choix de travailler avec :

- GIT CMD : commande
- GIT BASH : ligne de commande pour un syntaxe Unix.
- GIT GUI : en mode graphique

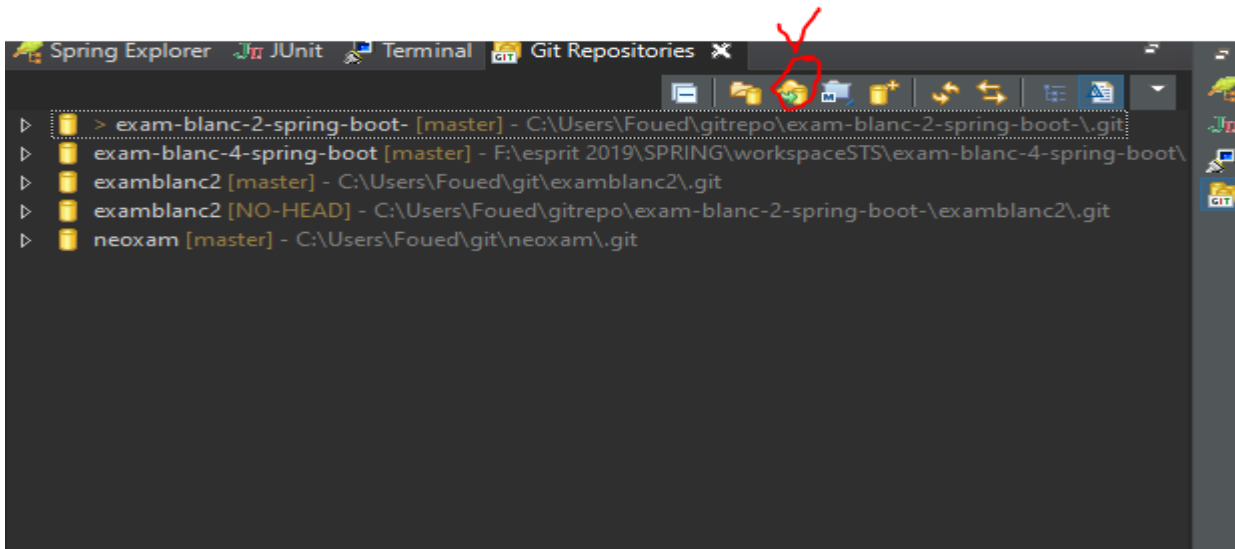
Dans ce TP, nous allons travailler avec EGIT ; Git intégré sous Eclipse afin de vous familiariser avec son utilisation.

Notez bien : Il s'agit bien d'une utilisation de base de Git. Pour un niveau plus avancé vous pouvez se référer a <https://www.atlassian.com/git/tutorials/>

EGIT :

Sous Eclipse :View -> Show View -> Other -> Git Repository

Choisir:

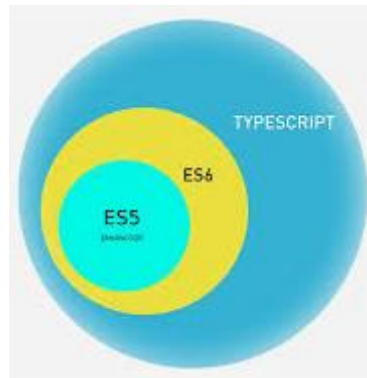


Pour importer votre repository(dépôt).

Vous pouvez également faire (commit ,push, pull, create new branch...) en cliquant sur le bouton droite sur le projet en choisissant **Team**.

II. TypeScript :

TypeScript est un sur-ensemble typé de JavaScript qui se compile en JavaScript simple. N'importe quel navigateur. Tout hôte. N'importe quel OS. Open source.



	JavaScript	TypeScript
Type	Langage interprété	Langage compilé
Développer par	Netscape Communication, ECMA international ,Mozilla Foundation	Anders Heilsberg chez Microsoft
Exécution	Client	Client
Extension	.tsx ou .ts	.js
Syntaxe	Module, fonction, variable, extension...	<script> //javascriptCode </script>

Via <https://www.typescriptlang.org//index.html> vous pouvez également télécharger le compilateur TS.

Dans ce TP nous allons installer TypeScript via NodeJs. Lien <https://nodejs.org/en/>

Ensuite taper la commande : **npm install -g typescript**

Comment compiler un code .ts ?

tsc Sample.ts

Sample.ts **—————>** **Sample.js**

Un dossier (TypeScriptDemos-master) est fourni avec ce TP, contenant des exemples de codes .ts.
Exécuter des exemples de votre choix.

III. Premier pas avec Angular :

1-Préparation de l'environnement :

-**NodeJs** est déjà installé.

-**Angular CLI** est un outil en ligne de commande puissant qui permet de créer un projet. Lancer la commande d'installation : **npm install -g @angular/cli**

-Ensuite créez votre première application : **ng new nom-projet**

Dans un projet Angular, nous avons plusieurs fichiers de configuration à savoir:

App : Le coeur de notre application c'est ce dossier qui contiendra tout le code de notre application.

Package.json : C'est un fichier de configuration NPM. Il contient des configurations Angular. Ces configurations permettent d'installer les autres packages en relation, en utilisant NPM.

Tsconfig.json : tous les fichiers TypeScript (*.ts) seront compilés et convertis vers des fichiers JavaScript (*.js). Ce qui permet, aux différents navigateurs, de lire ces fichiers. Pour assurer cette action on fait appel au fichier tsconfig.json.

Index.html : c'est notre page de démarrage. Elle contient les appels vers les composants Angular.

2-Compiler votre projet :

- Avec un éditeur de texte tel que **Visual Studio Code**, lancer le terminal et taper la commande **ng serve** pour compiler et pouvoir lancer votre projet.

Atelier 1 :

1-ng new examBlancFront

2- Créer un nouveau dossier **components** sous le répertoire **/app** (c'est une sorte d'organisation et de séparation du composant père (AppComponent) de ses composants fils.

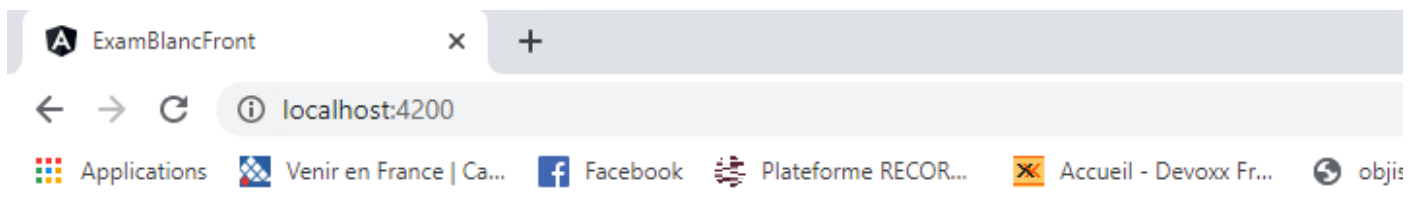
2- Les trois fichiers. **component.ts**, **.component.html** et **.component.css** sous le répertoire **/components** seront créés.

Exemple : On va créer un composant **UserComponent** :

`ng generate component /component/user`

3- Ajouter le sélecteur **<user></user>** au template parent **app.component.html** en supprimant le contenant par défaut.

4-Comiler et lancer le projet : **ng serve**, dans votre navigateur taper **localhost :4200**



Examen Blanc :Atelier Premier pas avec Angular

user works!

5- Créer le model user : `ng generate class /model/user`

```
exam_blanc3_front > examBlancFront > src > a
1  export class User {
2      id:number;
3      email:string;
4      pwd:string;
5      fname:string;
6      lname:string;
7  }
8  }
```

6-créer le service userservice: `ng g s /service/user`

7-créer les trois composants:

`ng generate component /component/listusers`

`ng generate component /component/formuser`

8-Modifier userService.ts:

```
@Injectable({
  providedIn: 'root'
})
export class UserService {
  private baseUrl:string;

  constructor(private http:HttpClient) {
    this.baseUrl='http://localhost:8090/api';
  }
  public getListUser():Observable<User[]>{
    return this.http.get<[User]>('http://localhost:8090/api/users');
  }
  public createtUser(User:User){
    return this.http.post('http://localhost:8090/api/addusert',User);
  }
}
```

9-Modifier ListuserComponent.ts:

```
@Component({
  selector: 'app-listusers',
  templateUrl: './listusers.component.html',
  styleUrls: ['./listusers.component.css']
})
export class ListusersComponent implements OnInit {
  users:User[];
  constructor(private userService:UserService) { }

  ngOnInit() {
    this.userService.getListUser().subscribe(data=>{
      this.users=data;
    })
  }
}
```

Modifier ListuserComponent.html

```
<p>listusers works!</p>
<div class="card my-5">
  <div class="card-body">
    <table class="table table-bordered table-striped">
      <thead class="thead-dark">
        <tr>
          <th scope="col">#</th>
          <th scope="col">FirstName</th>
          <th scope="col">LasttName</th>
```

```

        <th scope="col">Email</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let User of users">
        <td>{{ User.id }}</td>
        <td>{{ User.fname }}</td>
        <td>{{ User.lname }}</td>

        <td><a href="mailto:{{ User.email }}">{{ User.email }}</a></td>
      </tr>
    </tbody>
  </table>
</div>
</div>

```

11-Dans Formuser Component:

```

@Component({
  selector: 'app-formusers',
  templateUrl: './formusers.component.html',
  styleUrls: ['./formusers.component.css']
})
export class FormusersComponent {
  User: User;

  constructor(
    private route: ActivatedRoute,
    private router: Router,
    private userService: UserService) {
    this.User = new User();
  }

  onSubmit() {
    this.userService.createUser(this.User).
      subscribe(result => this.gotoUserList());
  }

  gotoUserList() {
    this.router.navigate(['/users']);
  }
}

```

Et dans FormuserComponent.html

```

<div class="card my-5">
  <div class="card-body">
    <form (ngSubmit)="onSubmit()" #userForm="ngForm">
      <div class="form-group">
        <label for="name">FName</label>
        <input type="text" [(ngModel)]="User.fname"

```

```

        class="form-control"
        id="fname"
        name="fname"
        placeholder="Enter your name"
        required #name="ngModel">
    </div>

    <div [hidden]="!name.pristine" class="alert alert-danger">Name is required</div>
    <div class="form-group">
        <label for="name">LName</label>
        <input type="text" [(ngModel)]="User.lname"
            class="form-control"
            id="lname"
            name="lname"
            placeholder="Enter your last name"
            required #name="ngModel">
    </div>

    <div [hidden]="!name.pristine" class="alert alert-
danger">Last Name is required</div>

    <div class="form-group">
        <label for="email">Email</label>
        <input type="text" [(ngModel)]="User.email"
            class="form-control"
            id="email"
            name="email"
            placeholder="Enter your email address"
            required #email="ngModel">
        <div [hidden]="!email.pristine" class="alert alert-
danger">Email is required</div>
    </div>
    <button type="submit" [disabled]="!userForm.form.valid"
        class="btn btn-info">Submit</button>
    </form>
</div>
</div>

```

12-app.component.html

```

<div class="container">
    <div class="row">
        <div class="col-md-12">
            <div class="card bg-dark my-5">
                <div class="card-body">
                    <h2 class="card-title text-center text-white py-3">{{ title }}</h2>
                    <ul class="text-center list-inline py-3">
                        <li class="list-inline-item">
                            <a routerLink="/users" class="btn btn-info">List Users</a>
                        </li>
                        <li class="list-inline-item">
                            <a routerLink="/addusert" class="btn btn-info">Add User</a>
                        </li>
                    </ul>
                </div>
            </div>
        </div>
    </div>
</div>

```



```

        </ul>
      </div>
    </div>
    <router-outlet></router-outlet>
  </div>
</div>
</div>

```

Et app.module.ts :

```

@NgModule({
  declarations: [
    AppComponent,
    ListusersComponent,
    FormusersComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    FormsModule
  ],
  providers: [UserService],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

13-Dans app.routing.module

```

const routes: Routes = [
  { path: 'users', component: ListusersComponent },
  { path: 'addusert', component: FormusersComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }

```

14-Bootstrap votre application :

Dans index.html

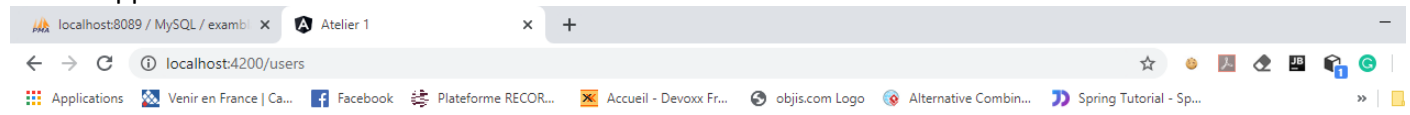
```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Atelier 1</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <link rel="stylesheet"

```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

15-run application :



listusers works!

#	FirstName	LasttName	Email
1			xxxx@yyyy.com
2			xxxx@yyyy.com
4			