

CGI

CPlayer Application

Presented By

Juhi Mandal

Harshitha P

Objectives

- About CPlayer Application
- Setup
 - Spring Boot
 - MySQL
- Angular
 - Component
 - Services
 - Launching Browser
- API Testing
 - Postman
- e2e Testing
 - Protractor
- Selenium
 - POM
 - Object Repository
 - Retrieving data through Excel Sheet
 - Launching Browser

About CPlayer Application

- CPlayerApp
- FavoriteService
 - FavouriteServiceApplication.java
 - FavouriteServiceSwagger.java
 - GlobalExceptionController.java
 - PlayerController.java
 - BattingMatchType.java
- UserService
 - UserServiceApplication.java
 - UserSwaggerConfig.java
 - UserAuthController.java
 - User.java
 - UserAuthRepository.java

Setups:

- Spring-tool-suite-4-4.3.2
- MySQL Workbench8.0 CE
- Angular 7
- Selenium

Visual Studio Code

- Visual Studio Code supports almost every major programming language. Common web-based languages like JavaScript, TypeScript, CSS, and HTML ship in the box, and rich extensions from the Visual Studio marketplace provide completions, linting, debugging, and refactoring support for hundreds more.

Open Visual Studio Code

- In Visual studio open terminal and type:
 - > npm install
 - > ng serve -o

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\CPlayerAppProjectVisualStudio\CplayerApp\CPPlayersUI>ng serve -o
Your global Angular CLI version (8.3.3) is greater than your local
version (7.0.7). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
** Angular Live Development Server is listening on localhost:4200, open your browser on http:
  17% building modules 65/75 modules 10 active ...date-player\update-player.component.cssBrows
  ommand `npm update caniuse-lite browserslist` 

Date: 2019-09-18T03:53:50.336Z
Hash: 75e3ce941906ed5da793
Time: 30814ms
chunk {main} main.js, main.js.map (main) 134 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 236 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 182 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 6.78 MB [initial] [rendered]
i [wdm]: Compiled successfully.
[]
```

CPlayer Application

Launching Browser before
creating userprofile component

The screenshot shows a web browser window with the URL `localhost:4201/login` in the address bar. The page has a red header bar with the text "Cricket Player" and navigation links for "Search Player", "Favourite List", and "Logout". Below the header, the main content area is titled "Login User". It contains two input fields: one for "User ID" with a placeholder "Rectangular Snip" and another for "Password". At the bottom are two buttons: a grey "Login User" button and a purple "Register" button.

Cricket Player

Search Player Favourite List Logout

Login User

User ID :*

>Password :*

Login User Register

Register Page (Before User Profile Component)

← → ⌛ ⓘ localhost:4201/register

Cricket Player

Search Player

Favourite List

Logout

Register User

First Name :

harshita

Last Name :

ps

User ID : *

harshi12

Password : *

harshi12

Register User

Clear

Login Page (Before User Profile Component)

← → ⌛ ⓘ localhost:4201/login

Cricket Player

Search Player

Favourite List

Logout

Login User

User ID : *

harshi12

• Rectangular Snip

Password : *

harshi12|

Login User

Register

Angular

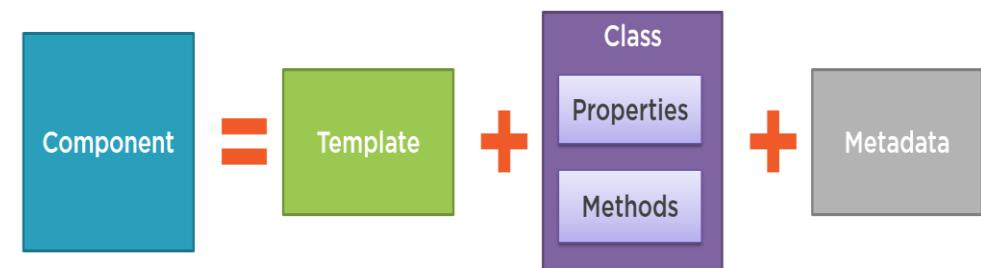
- Angular lets you extend HTML with HTML attributes called directives
- Angular directives offers functionality to HTML applications
- Angular provides built-in directives and user defined directives

History of Angular

- Angular JS- oct 20, 2010
- Version 2- Final released 2016, sept
- Version 4- Final released 2017 march
- Version 5- 2017 nov, progressive web apps
- Version 6 – May 2018
- Version 7- oct 2018
- Version 8 – may 28, 2019

Component

- Major part of the development with Angular is done in the components.
- Components are basically classes that interact with the .html file of the component, which gets displayed on the browser.

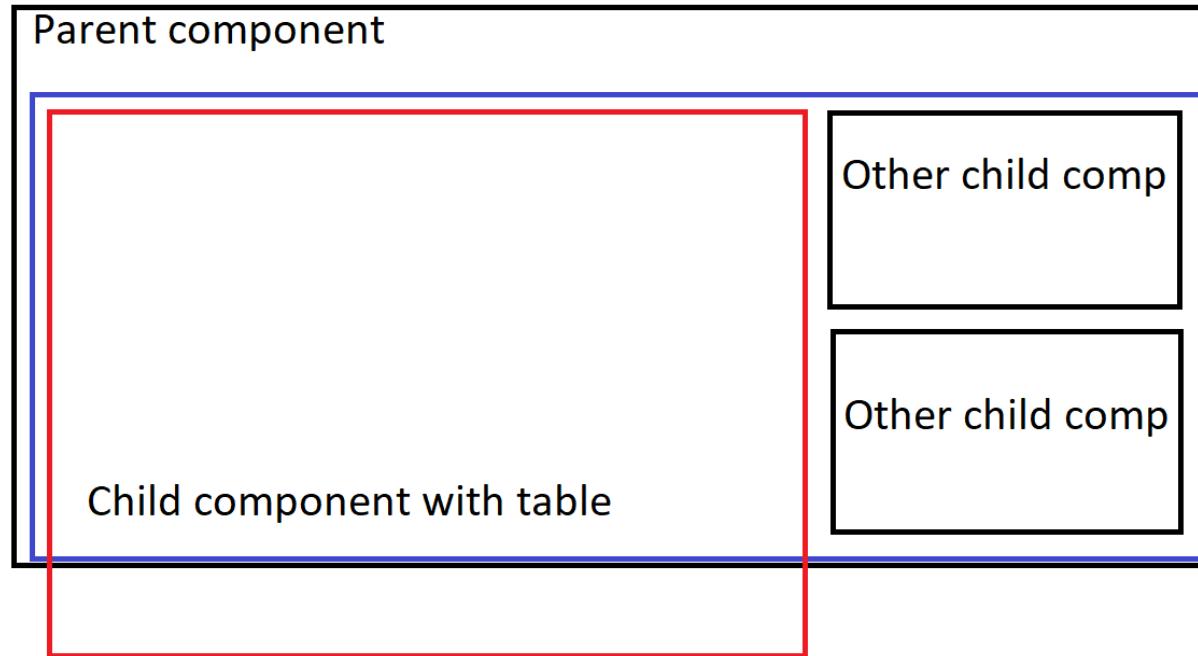


- View layout
- Created with HTML
- Includes binding and directives
- Code supporting the view
- Created with TypeScript
- Properties: data
- Methods: logic
- Extra data for Angular
- Defined with a decorator

File Structure of Component

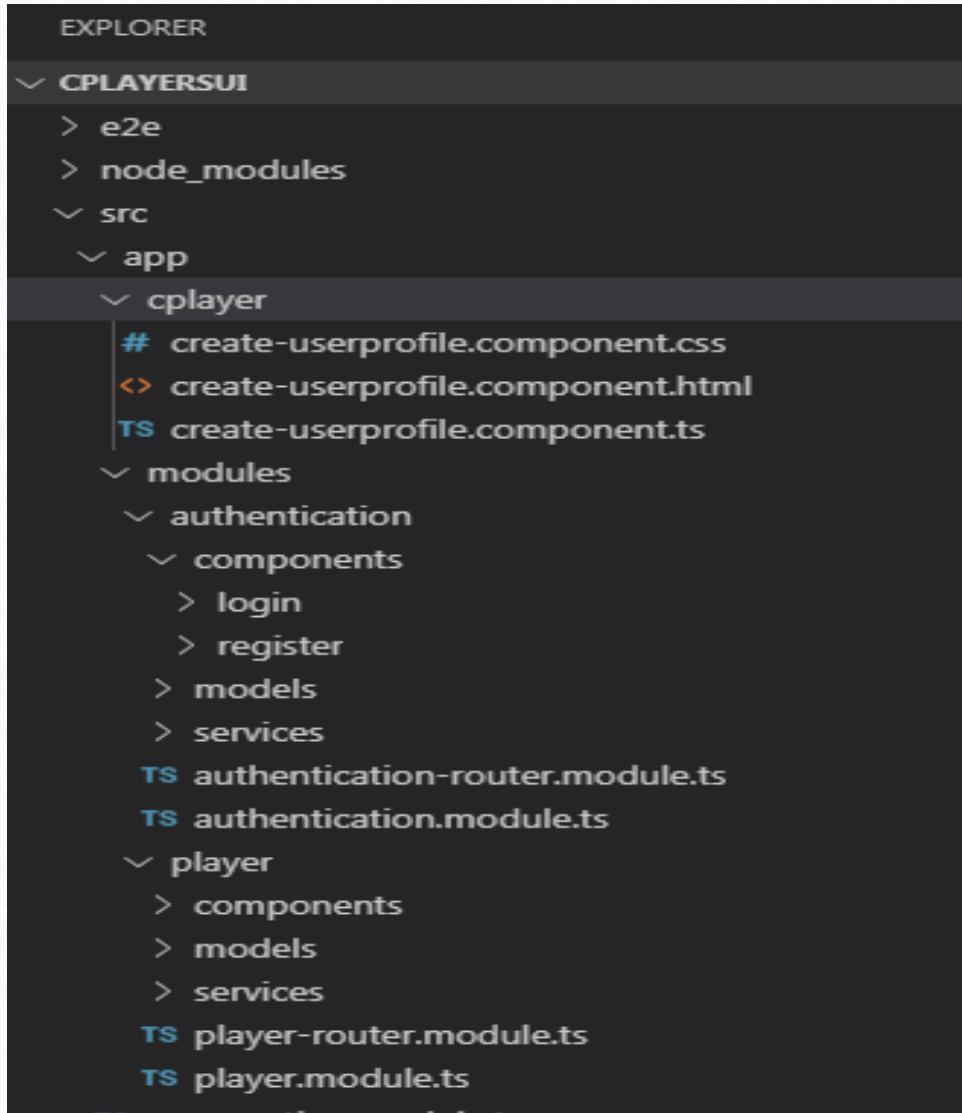
- ❑ app.component.css
- ❑ app.component.html
- ❑ app.component.spec.ts
- ❑ app.component.ts
- ❑ app.module.ts

Continue...



Creating Component

```
ng g c cplayer/ create-
userprofile --spec= false -- 
flat= true
```



Creating Registration Form

src > app > cplayer > create-userprofile.component.html > ...

```
1 <form class="register-form" #registerForm="ngForm" (ngSubmit)="registerUser()" [style.fontSize.px]="15">
2   <div class="register-block">
3     <div class="row">
4       <h3>User Profile </h3>
5     </div>
6     <mat-label class="label-width"><b>First Name</b> :</mat-label>
7     <mat-form-field class="register-input-full-width">
8       <input id="firstName" matInput [(ngModel)]="user.firstName" name="firstName" type="text" required>
9     </mat-form-field>
10    </mat-label>
11    <mat-label class="label-width"><b>Last Name</b> :</mat-label>
12    <mat-form-field class="register-input-full-width">
13      <input id="lastName" matInput [(ngModel)]="user.lastName" name="lastName" type="text" required>
14    </mat-form-field>
15    </mat-label>
16    <mat-label class="label-width"><b>User ID</b> :<span style="color:#red">*</span></mat-label>
17    <mat-form-field class="register-input-full-width">
18      <div class="form-group" [class.has-error]="userId.invalid && userId.touched" [class.has-success]="userId.valid">
19        <input id="userId" matInput [(ngModel)]="user.userId" name="userId" #userId="ngModel" required>
20        <span class="help-block" *ngIf="userId.errors?.required && userId.touched" style="color:#red">
21          User Id is required
22        </span>
23      </div>
24    </mat-form-field>
25  </mat-label>
```

src > app > cplayer > create-userprofile.component.html > ...

```
25   </mat-label>
26
27   <mat-label class="label-width"><b>Password</b> :<span style="color:#red">*</span></mat-label>
28   <mat-form-field class="register-input-full-width">
29     <div class="form-group" [class.has-error]="password.invalid && password.touched" [class.has-success]="password.valid">
30       <input id="password" matInput [(ngModel)]="user.password" name="password" #password="ngModel" required>
31       <span class="help-block" *ngIf="password.errors?.required && password.touched" style="color:#red">
32         Password is required
33       </span>
34     </div>
35   </mat-form-field>
36   </mat-label>
37
38   <br>
39   <div class="row">
40     <button class="register-user" mat-raised-button color="primary" type="submit" (click)="userprofile()" [disabled]="registerForm.invalid">Register UserProfile</button>&nbsp;
41     <button mat-raised-button color="primary" (click)="resetInput()">Clear</button>
42   </div>
43 </div>
44 </div>
45 </div>
46 </form>
47
```

create-userprofile.component.css



The screenshot shows a code editor interface with a dark theme. The top navigation bar includes File, Edit, Selection, View, Go, Debug, Terminal, and Help. The title bar indicates the current file is 'create-userprofile.component.css' in the 'CPlayersUI' workspace. The left sidebar, titled 'EXPLORER', displays the project structure under 'CPLAYERSUI'. Key components shown include 'e2e', 'node_modules', 'src' (with 'app' and 'cplayer' subfolders), and various files like 'create-userprofile.component.css', 'app-routing.module.ts', and 'app.module.ts'. The main editor area contains the CSS code for 'create-userprofile.component.css'.

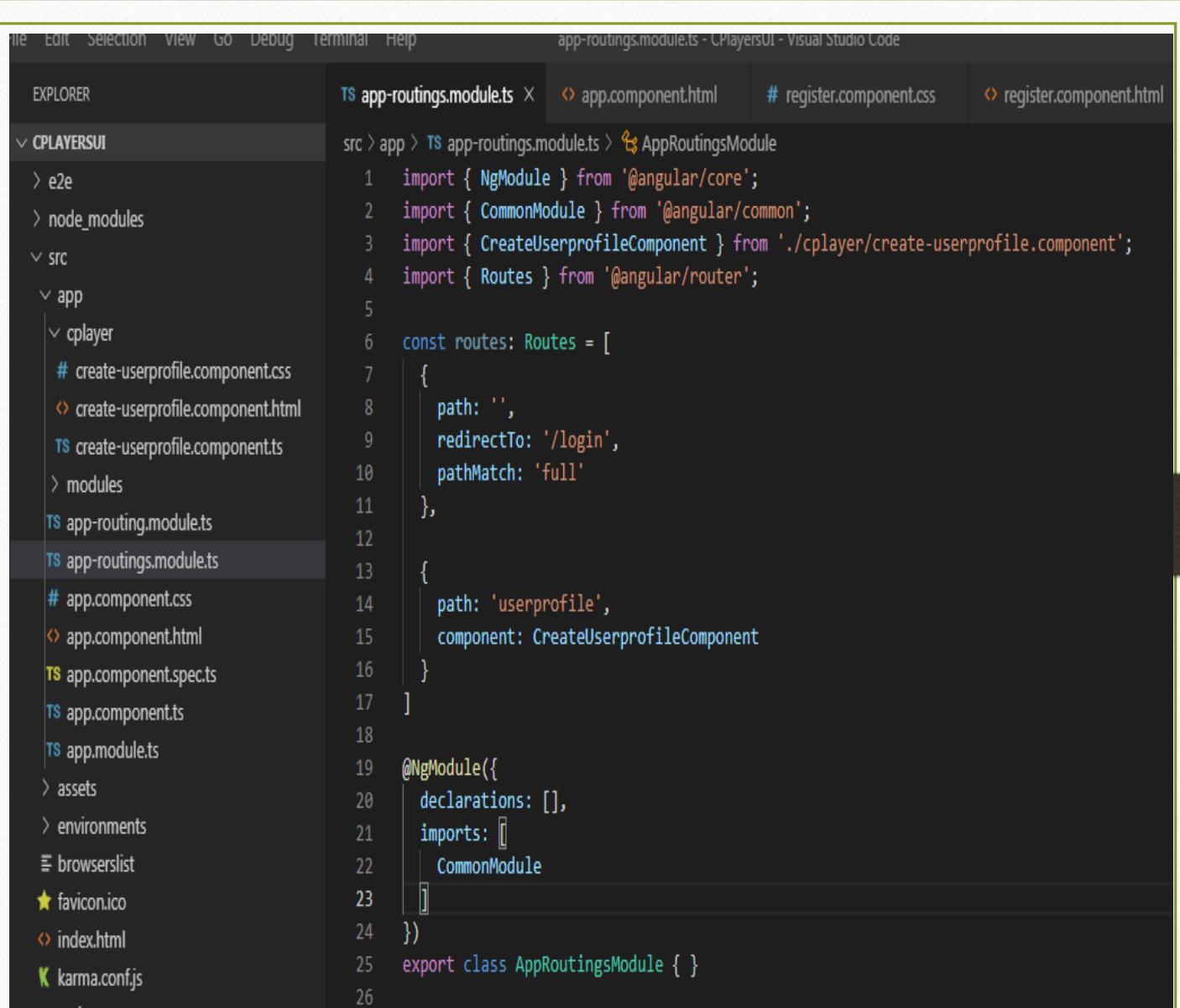
```
# create-userprofile.component.css
src > app > cplayer > # create-userprofile.component.css > .register-block
1  .register-block {
2    margin: auto;
3    width: 50%;
4    padding: 10px;
5    padding-top: 50px;
6    display: flex;
7    align-content: center;
8    justify-content: center;
9    flex-flow: column nowrap;
10   align-items: center;
11 }
12
13 .register-input-full-width {
14   width: 80%;
15 }
16
17 h3 {
18   text-align: center;
19 }
20
21 .label-width {
22   font-size: 14px;
23 }
```

create-userprofile.component.ts

```
ponent.css      register.component.html      TS authentication-router.module.ts      TS app-routing.module.ts      TS create-userpr
src > app > cplayer > TS create-userprofile.component.ts > CreateUserProfileComponent
  1  import { Component, OnInit } from '@angular/core';
  2
  3  import { Router } from '@angular/router';
  4  import { User } from '../modules/authentication/models/user';
  5  import { AuthenticationService } from '../modules/authentication/services/authentication.service';
  6
  7
  8  @Component({
  9    selector: 'app-create-userprofile',
 10    templateUrl: './create-userprofile.component.html',
 11    styleUrls: ['./create-userprofile.component.css']
 12  })
 13  export class CreateUserProfileComponent implements OnInit {
 14
 15    user: User;
 16
 17    constructor(private _authService: AuthenticationService, private _router: Router) {
 18      this.user = { userId: '', password: '', firstName: '', lastName: '' };
 19    }
 20
 21    ngOnInit() {
 22    }
 23
 24    userprofile() {
 25      // console.log('user object is ' ,this.user);
 26      this._authService.registerUser(this.user).subscribe((data) => {
 27        console.log('user data' + data);
 28        this._router.navigate(['/login']);
 29      })
 30    }
 31  }
 32 }
```

Creating Module

```
ng g m app-routings --flat=true --spec=false
```



The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File Edit Selection View Go Debug Terminal Help
- Title Bar:** app-routings.module.ts - CPlayersUI - Visual Studio Code
- Explorer:** Shows the project structure under CPLAYERSUI:
 - e2e
 - node_modules
 - src
 - app
 - cplayer
 - # create-userprofile.component.css
 - create-userprofile.component.html
 - create-userprofile.component.ts
 - modules
 - app-routing.module.ts
 - app-routings.module.ts
 - # app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - assets
 - environments
 - browserslist
 - favicon.ico
 - index.html
 - karma.conf.js
 - Editor:** The file app-routings.module.ts is open, showing the following code:

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { CreateUserProfileComponent } from './cplayer/create-userprofile.component';
import { Routes } from '@angular/router';

const routes: Routes = [
  {
    path: '',
    redirectTo: '/login',
    pathMatch: 'full'
  },
  {
    path: 'userprofile',
    component: CreateUserProfileComponent
  }
]

@NgModule({
  declarations: [],
  imports: [
    CommonModule
  ]
})
export class AppRoutingModule { }
```
 - Quick Access:** # register.component.css, register.component.html

app-routing-module.ts

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File Edit Selection View Go Debug Terminal Help
- Title Bar:** app-routing.module.ts - CPIlayersUI - Visual Studio Code
- Explorer:** Shows the project structure under CPILAYERSUI:
 - e2e
 - node_modules
 - src
 - app
 - cplayer
 - # create-userprofile.component.css
 - create-userprofile.component.html
 - TS create-userprofile.component.ts
 - modules
 - TS app-routing.module.ts
 - TS app-routings.module.ts
 - # app.component.css
 - app.component.html
 - TS app.component.spec.ts
 - TS app.component.ts
 - TS app.module.ts
 - assets
 - environments
 - browserslist
 - favicon.ico
 - index.html
- Code Editor:** Displays the content of app-routing.module.ts:

```
src > app > TS app-routing.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { Routes, RouterModule } from '@angular/router';
3 import { CreateUserProfileComponent } from './cplayer/create-userprofile.component';
4
5 const routes: Routes = [
6   {
7     path: '',
8     redirectTo: '/login',
9     pathMatch: 'full'
10 },
11 {
12   path: 'userprofile',
13   component: CreateUserProfileComponent
14 }
15 ]
16 ]
17
18 @NgModule({
19   imports: [RouterModule.forRoot(routes)],
20   exports: [RouterModule]
21 })
22 )
23 export class AppRoutingModule { }
```

app.component.html

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** app.component.html - CPlayersUI - Visual Studio Code.
- Explorer:** Shows the project structure under 'CPLAYERSUI'. The 'app' folder is expanded, showing 'cplayer', 'create-userprofile.component.css', 'create-userprofile.component.html', 'create-userprofile.component.ts', 'modules', 'app-routing.module.ts', 'app-routings.module.ts', 'app.component.css', 'app.component.html' (which is the active file), 'app.component.spec.ts', 'app.component.ts', and 'app.module.ts'. Other files like 'assets', 'environments', and 'browserlist' are also listed.
- Editor:** The 'app.component.html' file is open, displaying the following code:

```
src > app > app.component.html > mat-toolbar
1  <mat-toolbar color="warn">
2    <span><strong>Cricket Player</strong></span>&ampnbsp&ampnbsp&ampnbsp
3    <button class="search-button" mat-button [routerLink]="/players/search">Search Player</button>
4    <button mat-button [routerLink]="/players/favouritelist">Favourite List</button>
5
6    <button mat-button [routerLink]="/userprofile">User Profile</button>
7
8    <button mat-button (click)="logoutUser()">Logout</button>
9  </mat-toolbar>
10 <router-outlet></router-outlet>
11
```

app.module.ts

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar on the left containing project files like e2e, node_modules, src, app, cplayer, and various component and module files.
- Top Bar**: Tabs for `app.module.ts`, `index.html`, `create-userprofile.component.html`, and `app-routing.module.ts`.
- Code Area**: The `app.module.ts` file content:

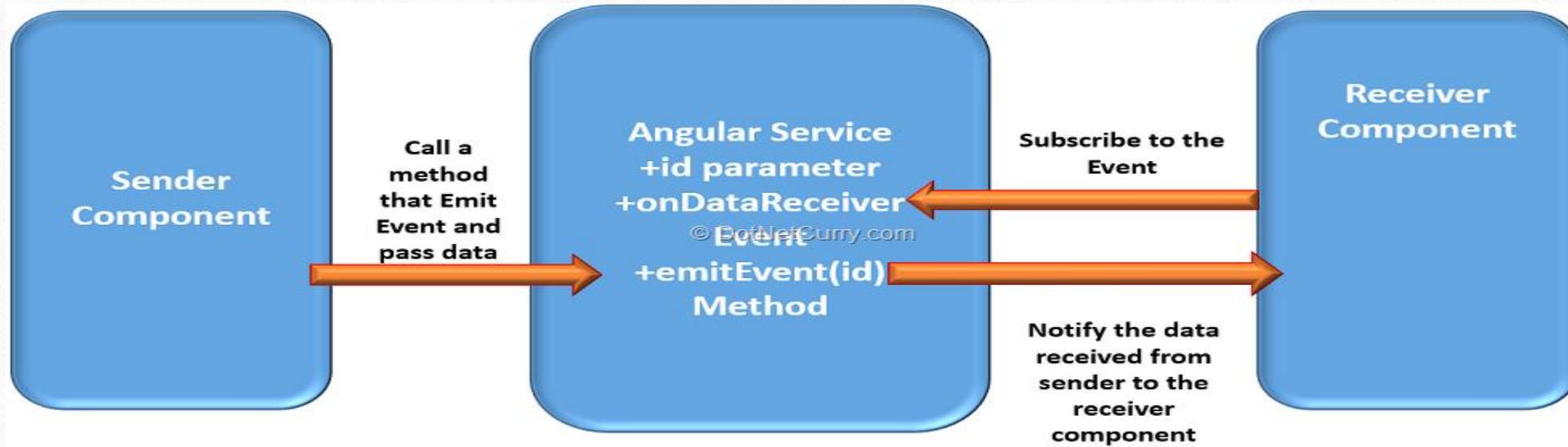
```
src > app > TS app.module.ts > AppModule
  6 import { FormsModule } from '@angular/forms';
  7 import { MatToolbarModule } from '@angular/material/toolbar';
  8 import { MatButtonModule, MatButton } from '@angular/material/button';
  9 import { MatInputModule } from '@angular/material/input';
 10 import { MatDialogModule } from '@angular/material/dialog';
 11 import { AuthenticationModule } from './modules/authentication/authentication.module';
 12 import { PlayerModule } from './modules/player/player.module';
 13 import { CreateUserProfileComponent } from './cplayer/create-userprofile.component';
 14 import { AppRoutingModule } from './app-routing.module';

 15 @NgModule({
 16   declarations: [
 17     AppComponent,
 18     CreateUserProfileComponent
 19   ],
 20   imports: [
 21     BrowserModule,
 22     AppRoutingModule,
 23     BrowserAnimationsModule,
 24     FormsModule,
 25     MatToolbarModule,
 26     MatButtonModule,
 27     MatInputModule,
 28     MatDialogModule,
 29     AuthenticationModule,
 30     PlayerModule,
 31     AppRoutingModule
 32   ],
 33   providers: [],
 34   bootstrap: [AppComponent]
 35 })
 36 export class AppModule { }
```

The code editor has a dark theme and includes a status bar at the bottom with icons for file, search, and other functions.

Services

- AngularJS services are substitutable objects that are wired together using dependency injection (DI)



Types of Services

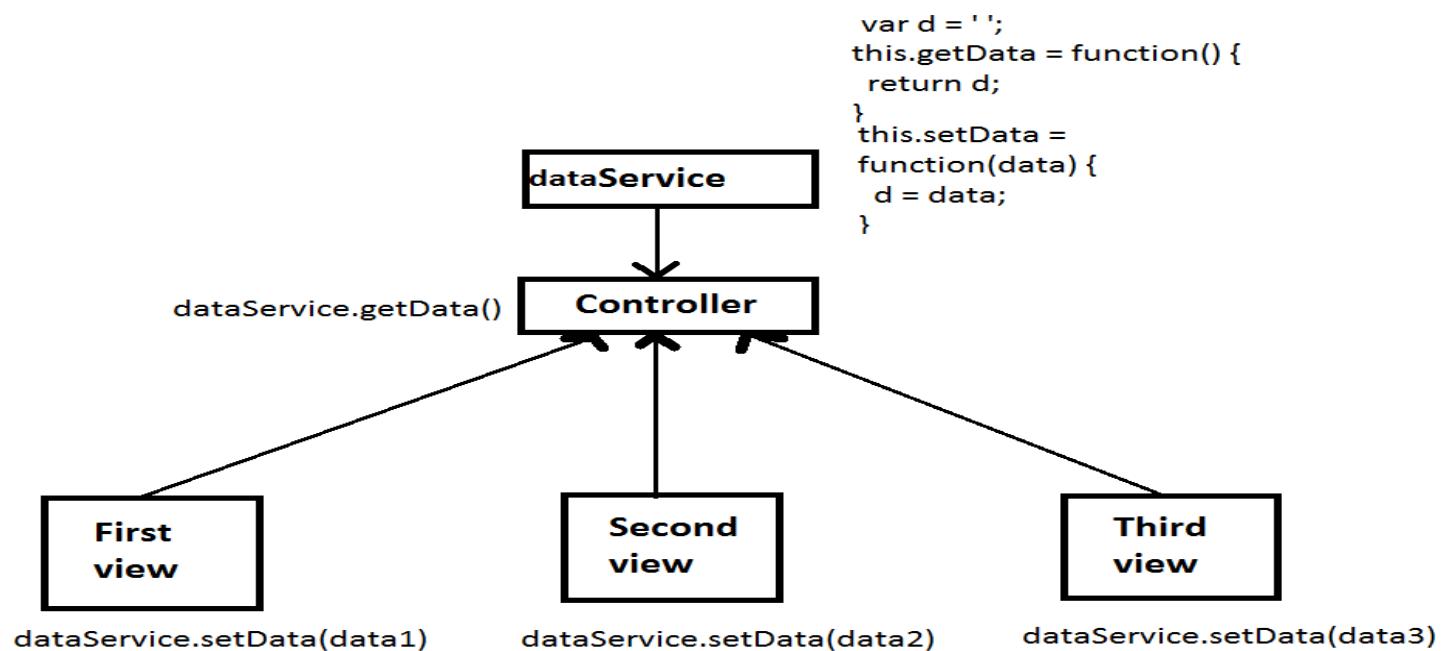
Factory

```
app.factory('helloService', function(){
  return {
    sayHello: function(text){
      return "Hello " + text;
    },
    sayGoodbye: function(text){
      return "Goodbye " + text;
    }
  };
});
```

Service

```
app.service('helloService', function(){
  this.sayHello= function(text){
    return "Hello " + text;
  };
  this.sayGoodbye = function(text){
    return "Goodbye " + text;
  };
});
```

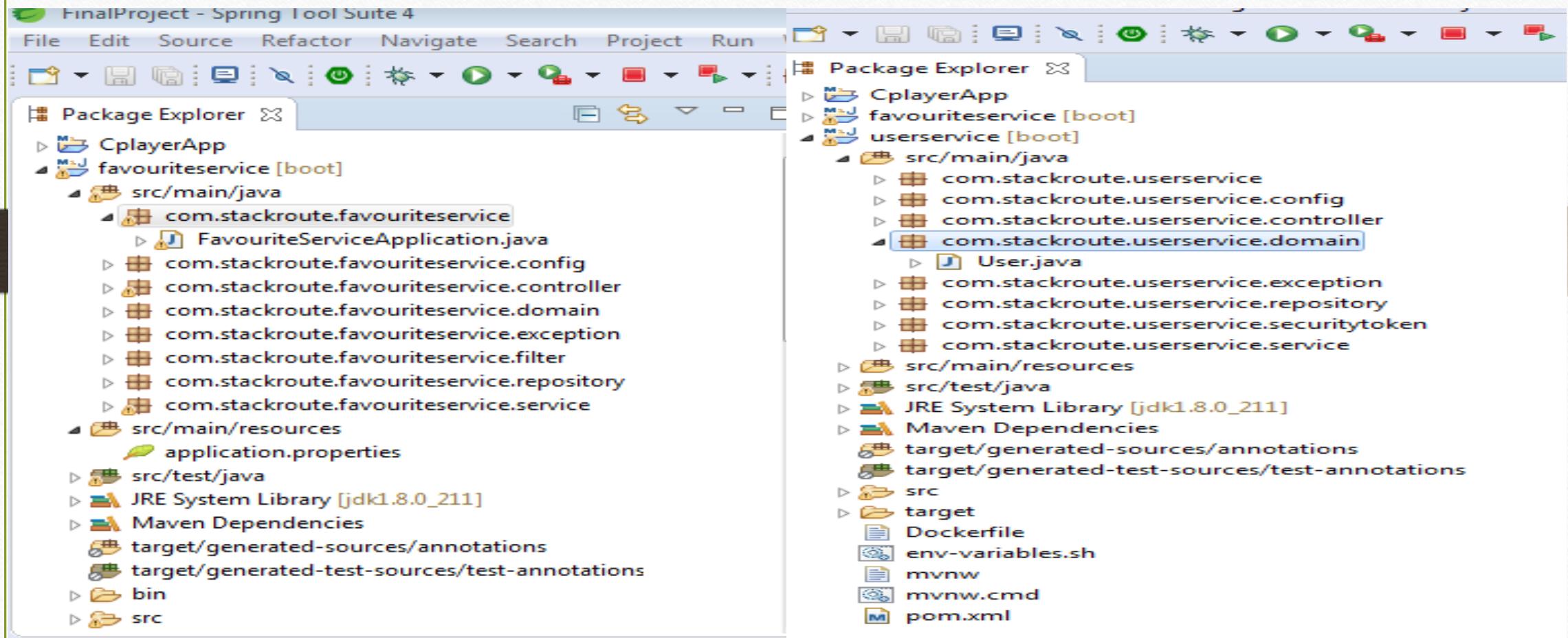
Angular Data Services



Spring Boot

- Spring Boot is an open source Java-based framework used to create a Micro Service.
- It is developed by Pivotal Team
- It is easy to create a stand-alone and production ready spring applications using Spring Boot.

Project Structure of Spring Boot



MySQL Database

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

The screenshot shows the MySQL Workbench interface. The left pane displays the Navigator with the 'playerdb' schema selected, showing tables like batting_match_types, batting_stats, and user. The right pane shows a query results grid for the 'user' table, with the following data:

userid	created	first_name	last_name	password	user_id	created_time_stamp
NULL	NULL	Juhi	Mandal	juhi123	juhi	2019-09-11 12:19:43
NULL	NULL	Juhi	Mandal	juhi123	juhi	2019-09-11 12:19:43
NULL	NULL	ismiti	mandal	ismiti123	ismiti	2019-09-11 12:48:54
NULL	NULL	juhi	mandal	juhi123	juhi123	2019-09-11 14:27:21
NULL	NULL	juhi	mandal	juhi123	juhi123	2019-09-11 14:27:23
NULL	NULL	Ismiti	Ghosh	ismi123	ismi	2019-09-11 14:46:19
NULL	NULL	juhi	krishna	juhi99	juhi99	2019-09-13 09:42:32
NULL	NULL	sunita	Ghotiya	Suni	Suni	2019-09-13 10:07:45
NULL	NULL	Juhi	Mandal	juhi123	jui123	2019-09-13 12:01:24
NULL	NULL	Juhi	Mandal	juhi123	jui123	2019-09-13 12:01:24
NULL	NULL	aaa	bbb	aaaa	aaaa	2019-09-13 12:06:09
NULL	NULL	aaa	bbb	aaaa	aaaa	2019-09-13 12:06:09
NULL	NULL	Juhi	Mandal	juhi1234	juhi1234	2019-09-13 12:07:26
NULL	NULL	Juhi	Mandal	juhi1234	juhi1234	2019-09-13 12:07:26
NULL	NULL	sunita	ghotiya	sunita123	sunita...	2019-09-13 16:41:42
NULL	NULL	shivali	chauhan	shivali123	shivali...	2019-09-13 16:42:58

Application Properties

The screenshot shows two Java projects in the Spring Tool Suite 4 interface:

- CplayerApp**: Contains the `application.properties` file with the following content:

```
1 spring.datasource.url=jdbc:mysql://localhost:3307/playerdb
2 spring.datasource.username=root
3 spring.datasource.password=juhi123
4 spring.jpa.hibernate.ddl-auto=update
5 spring.jpa.hibernate.show-sql=true
6 server.port=8081
7
```
- favouriteservice [boot]**: Contains the `application.properties` file with the following content:

```
1 spring.datasource.url=jdbc:mysql://localhost:3307/playerdb
2 spring.datasource.username=root
3 spring.datasource.password=juhi123
4 spring.jpa.hibernate.ddl-auto=update
5 spring.jpa.hibernate.show-sql=true
6 server.port=8082
7
```

The interface includes a toolbar, a menu bar (File, Edit, Navigate, Search, Project, Run, Window, Help), and several views like Package Explorer, Navigator, and Problems.

User.java Class

The screenshot shows an IDE interface with the following components:

- Package Explorer:** Shows the project structure with packages like CplayerApp, favouriteservice [boot], and userservice [boot]. The userservice package contains sub-packages com.stackroute.userservice, com.stackroute.userservice.config, com.stackroute.userservice.controller, and com.stackroute.userservice.domain.
- Code Editor:** Displays the User.java file content:

```
55
56     public String getFirstName() {
57         return firstName;
58     }
```
- Console:** Shows the application log for UserServiceApplication [Spring Boot App] running on Tomcat 9.0.14. The log output is as follows:

```
2019-09-18 11:15:04.526 INFO 7624 --- [           main] c.s.userservice.UserServiceApplication : Starting UserServiceApplication on admin-PC with
2019-09-18 11:15:04.528 INFO 7624 --- [           main] c.s.userservice.UserServiceApplication : No active profile set, falling back to default p
2019-09-18 11:15:05.520 INFO 7624 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repositories in DEFUALT
2019-09-18 11:15:05.579 INFO 7624 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 53ms
2019-09-18 11:15:05.899 INFO 7624 --- [           main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation
2019-09-18 11:15:06.191 INFO 7624 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8082 (http)
2019-09-18 11:15:06.214 INFO 7624 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2019-09-18 11:15:06.214 INFO 7624 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.14]
2019-09-18 11:15:06.220 INFO 7624 --- [           main] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which
2019-09-18 11:15:06.322 INFO 7624 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2019-09-18 11:15:06.322 INFO 7624 --- [           main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization compl
2019-09-18 11:15:06.456 INFO 7624 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2019-09-18 11:15:06.881 INFO 7624 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2019-09-18 11:15:06.920 INFO 7624 --- [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [
    name: default
    ...
]
2019-09-18 11:15:06.963 INFO 7624 --- [           main] org.hibernate.Version : HHH000412: Hibernate Core {5.3.7.Final}
2019-09-18 11:15:06.964 INFO 7624 --- [           main] org.hibernate.cfg.Environment : HHH000206: hibernate.properties not found
2019-09-18 11:15:07.067 INFO 7624 --- [           main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.0.
2019-09-18 11:15:07.185 INFO 7624 --- [           main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.
2019-09-18 11:15:07.676 INFO 7624 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persist
2019-09-18 11:15:08.128 WARN 7624 --- [           main] aWebConfiguration$JpaWebMvcConfiguration : spring.jpa.open-in-view is enabled by default. T
2019-09-18 11:15:08.495 INFO 7624 --- [           main] d.s.w.p.DocumentationPluginsBootstrapper : Context refreshed
2019-09-18 11:15:08.507 INFO 7624 --- [           main] d.s.w.p.DocumentationPluginsBootstrapper : Found 1 custom documentation plugin(s)
2019-09-18 11:15:08.518 INFO 7624 --- [           main] s.d.s.w.s.ApiListingReferenceScanner : Scanning for api listing references
2019-09-18 11:15:08.657 INFO 7624 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8082 (http) with cont
2019-09-18 11:15:08.659 INFO 7624 --- [           main] c.s.userservice.UserServiceApplication : Started UserServiceApplication in 4.652 seconds
```

FavouriteServiceApplication.java Class

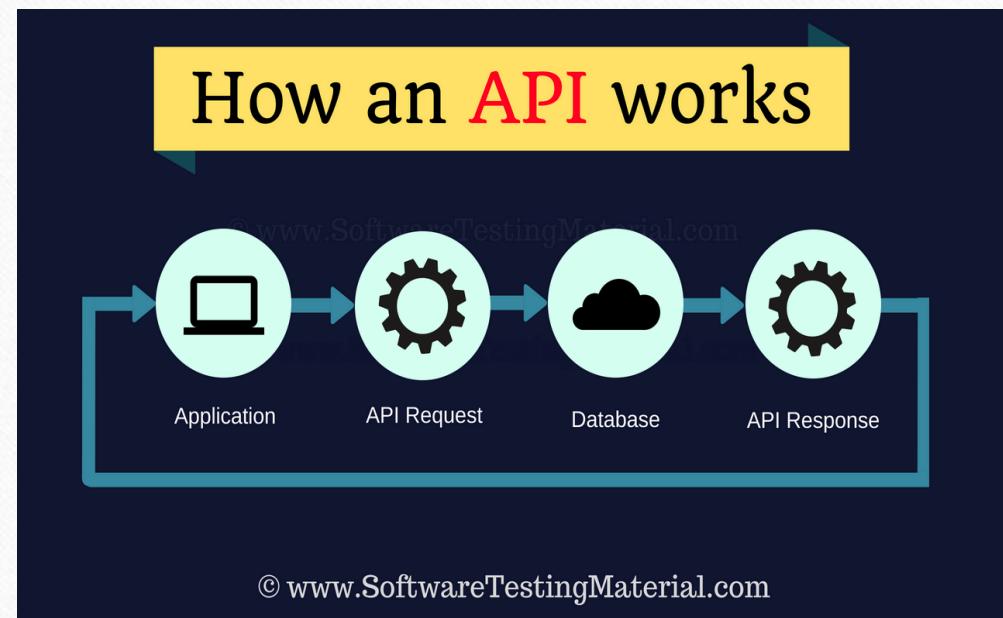
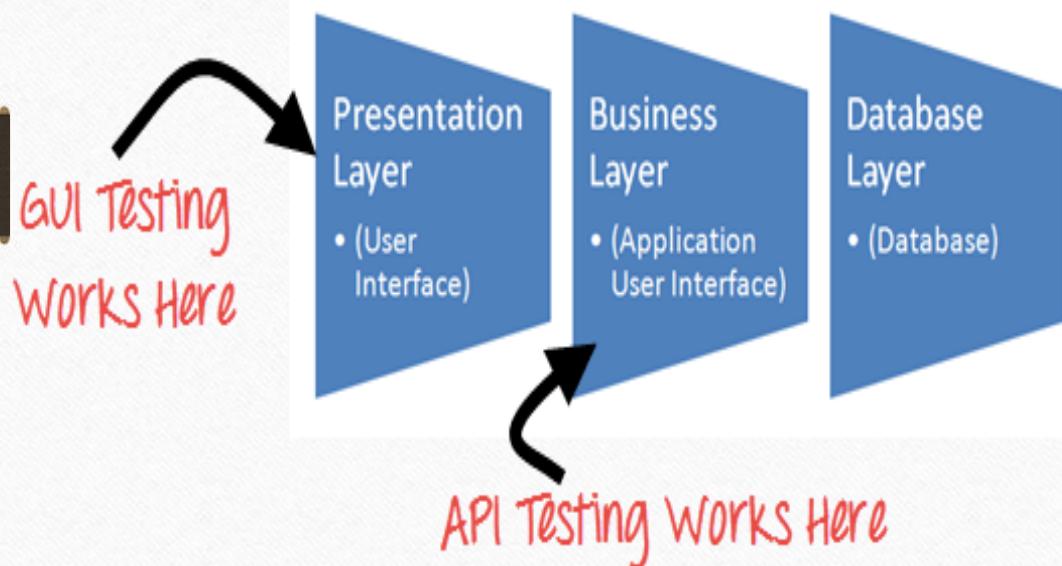
The screenshot shows the Spring Tool Suite 4 (STS 4) interface. The top navigation bar includes File, Edit, Source, Refactor, Navigate, Project, Run, Window, and Help. The left sidebar features the Package Explorer, Boot Dashboard, and a search bar for tags and working sets. The central workspace displays the code for `FavouriteServiceApplication.java` and the `JwtFilter.java` class. The `FavouriteServiceApplication.java` code is as follows:

```
package com.stackroute.favouriteservice.filter;
import java.io.IOException;
public class JwtFilter extends GenericFilterBean {
    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
```

The `JwtFilter.java` code is partially visible. Below the code editor is the Problems, Javadoc, Declaration, and Console tabs. The Console tab shows the application's startup logs:

```
favourite - FavouriteServiceApplication [Spring Boot App] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (Sep 18, 2019, 9:31:37 AM)
2019-09-18 09:31:40.240 INFO 7480 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2019-09-18 09:31:40.241 INFO 7480 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.20]
2019-09-18 09:31:40.247 INFO 7480 --- [           main] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal performance in production environments was loaded
2019-09-18 09:31:40.376 INFO 7480 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2019-09-18 09:31:40.376 INFO 7480 --- [           main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization started
2019-09-18 09:31:40.554 INFO 7480 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2019-09-18 09:31:40.972 INFO 7480 --- [           main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2019-09-18 09:31:41.013 INFO 7480 --- [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo ...
2019-09-18 09:31:41.013 INFO 7480 --- [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo ...
2019-09-18 09:31:41.057 INFO 7480 --- [           main] org.hibernate.Version : HHH000412: Hibernate Core {5.3.7.Final}
2019-09-18 09:31:41.057 INFO 7480 --- [           main] org.hibernate.cfg.Environment : HHH000206: hibernate.properties not found, using defaults
2019-09-18 09:31:41.195 INFO 7480 --- [           main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.0.Final}
2019-09-18 09:31:41.363 INFO 7480 --- [           main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect
Inside Player Class Constructor
inside skills constructor
2019-09-18 09:31:53.626 INFO 7480 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2019-09-18 09:31:54.224 WARN 7480 --- [           main] aWebConfiguration$JpaWebMvcConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering
2019-09-18 09:31:54.630 INFO 7480 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2019-09-18 09:31:54.942 INFO 7480 --- [           main] d.s.w.p.DocumentationPluginsBootstrapper : Context refreshed
2019-09-18 09:31:54.955 INFO 7480 --- [           main] d.s.w.p.DocumentationPluginsBootstrapper : Found 1 custom documentation plugin
2019-09-18 09:31:54.965 INFO 7480 --- [           main] s.d.s.w.s.ApiListingReferenceScanner : Scanning for api listing reference
2019-09-18 09:31:55.287 INFO 7480 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8081 (http)
2019-09-18 09:31:55.289 INFO 7480 --- [           main] c.s.f.FavouriteServiceApplication : Started FavouriteServiceApplication in 1.393 seconds (JVM running for 2.011)
```

API Testing



API Testing Tools

- Postman
- HP QTP(UFT)
- JMeter
- Katalon
- Tricentis

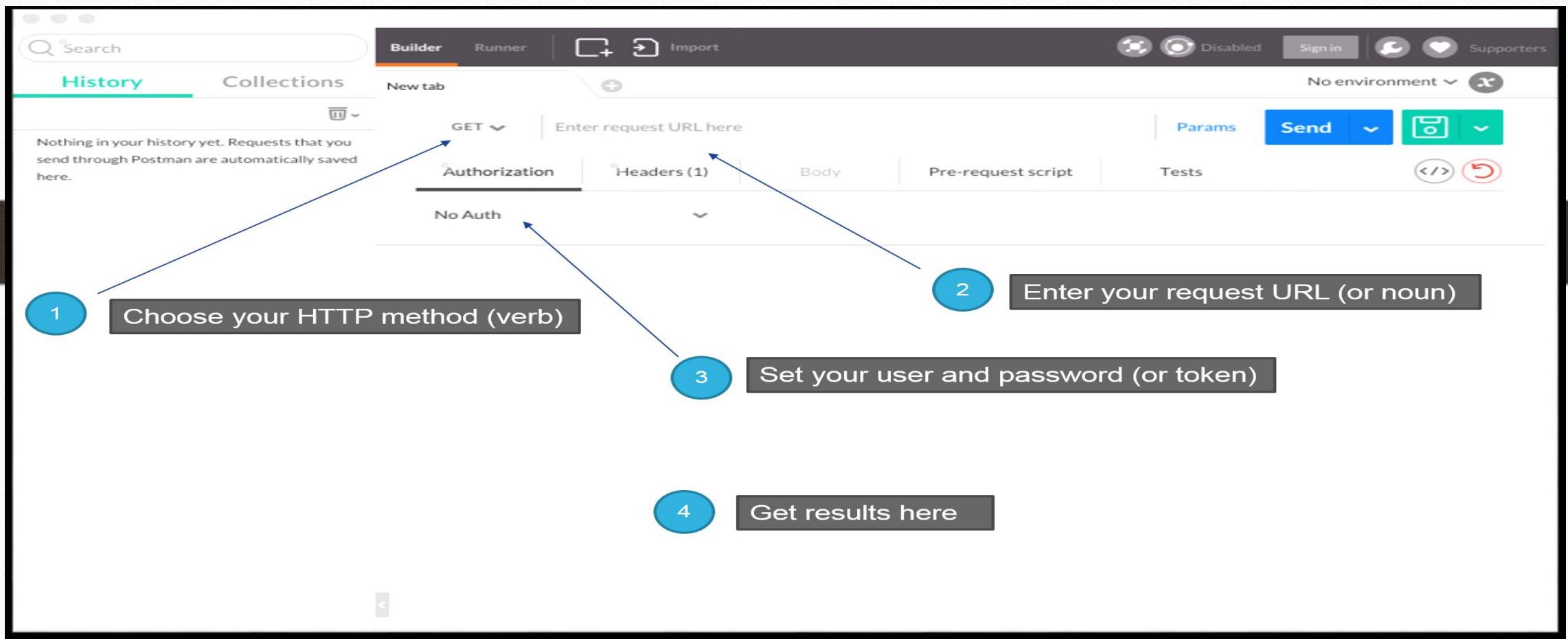
Postman

- Postman contains a powerful runtime based on Node.js that allows you to add dynamic behavior to requests and collections.
- Postman can be used to automate many types of tests including unit tests, functional tests, integration tests, end-to-end tests, regression tests, mock tests, etc.

Postman Request Method

- The most common methods are GET, POST, PUT, and DELETE. The names of the methods are self-explanatory. For example GET enables you to retrieve data from a server. POST enables you to add data to an existing file or resource in a server. PUT lets you replace an existing file or resource in a server. And DELETE lets you delete data from a server.

Continue...



Postman-Sending Request using Post

The screenshot shows the Postman application interface. A modal window is open, displaying a table of user data from a database. The table has columns: ID, Name, Username, Password, and DateCreated.

ID	Name	Username	Password	DateCreated
NULL	juni	mandoai	juni123	2019-09-11 14:27:21
NULL	juhi	mandal	juhi123	2019-09-11 14:27:23
NULL	Ismiti	Ghosh	ismi123	2019-09-11 14:46:19
NULL	juhi	krishna	juhi99	2019-09-13 09:42:32
NULL	sunita	Ghotiya	Suni	2019-09-13 10:07:45
NULL	Juhi	Mandal	juhi123	2019-09-13 12:01:24
NULL	Juhi	Mandal	juhi123	2019-09-13 12:01:24
NULL	aaa	bbb	aaaa	2019-09-13 12:06:09
NULL	aaa	bbb	aaaa	2019-09-13 12:06:09
NULL	Juhi	Mandal	juhi1234	2019-09-13 12:07:26
NULL	Juhi	Mandal	juhi1234	2019-09-13 12:07:26
NULL	sunita	ghotiya	sunita123	sunita...
NULL	shivali	chauhan	shivali123	shivali...
NULL	shivali	chauhan	shivali123	2019-09-13 16:41:42
NULL	shivali	chauhan	shivali123	2019-09-13 16:42:58

Postman Request Details:

- Method: POST
- URL: http://localhost:8082/user/login
- Headers: (9)
- Body (JSON):

```
1 {  
2   "userId": "sunita123",  
3   "password": "sunita123"  
4 }
```

Response Status: 200 OK

Response Body (Pretty JSON):

```
1 {  
2   "message": "User Successfully Logged In",  
3   "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWJ1Ii0iJzdw5pdGEExMjMiLCJpYXQiOjE1Njg3ODI0NDJ9.9xqmMpDpGqm7PzBdyUG7QmKAVOMR39s1JXIE9HRC6NA"  
4 }
```

Retrieving data through API-Postman

The screenshot shows the Postman application interface with two requests:

- Request 1 (Left):** POST <http://localhost:8081/api/v1/favouriteService/player>.
 - Body (JSON):**

```
1 {  
2   "id": "100",  
3   "pid": "1001",  
4   "name": "Sachin",  
5   "FullName": "Sachin Tendulkar"  
6 }  
7  
8 }
```
 - Authorization:** Bearer Token
 - Response Status:** 200 OK, Time: 30ms, Size: 303 B
- Request 2 (Right):** POST <http://localhost:8081/api/v1/favouriteService/player>.
 - Authorization:** Bearer Token (copied from Request 1)
 - Token:** eyJhbGciOiJIUzI1NiJ9.eyJzdWl... (long token value)
 - Response Status:** 200 OK, Time: 208ms, Size: 303 B

Postman-Update method

The screenshot shows the Postman application interface with the following details:

- Header Bar:** Shows various API endpoints (GET, POST, PUT) and a "No Environment" message.
- Title:** Jntitled Request
- Request Type and URL:** PUT http://localhost:8081/api/v1/favouriteService/player/106
- Body Tab:** Selected. Contains JSON data for updating a player record:

```
1 {  
2     "id": "100",  
3     "pid": "1001",  
4     "name": "Sachin",  
5     "FullName": "Sachin Tendulkar"  
6 }  
7  
8
```
- Response Status:** Status: 200 OK, Time: 33ms, Size: 303 B
- Pretty Tab:** Selected. Displays the JSON response:

```
1 {  
2     "errorCode": 500,  
3     "errorMessage": "Please Contact Your System Administrator"  
4 }
```

Postman-Delete method

DELETE ▾ http://localhost:8081/api/v1/favouriteService/player/100 **Send** ▾

Params Authorization ● Headers (10) **Body** ● Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA **JSON** ▾

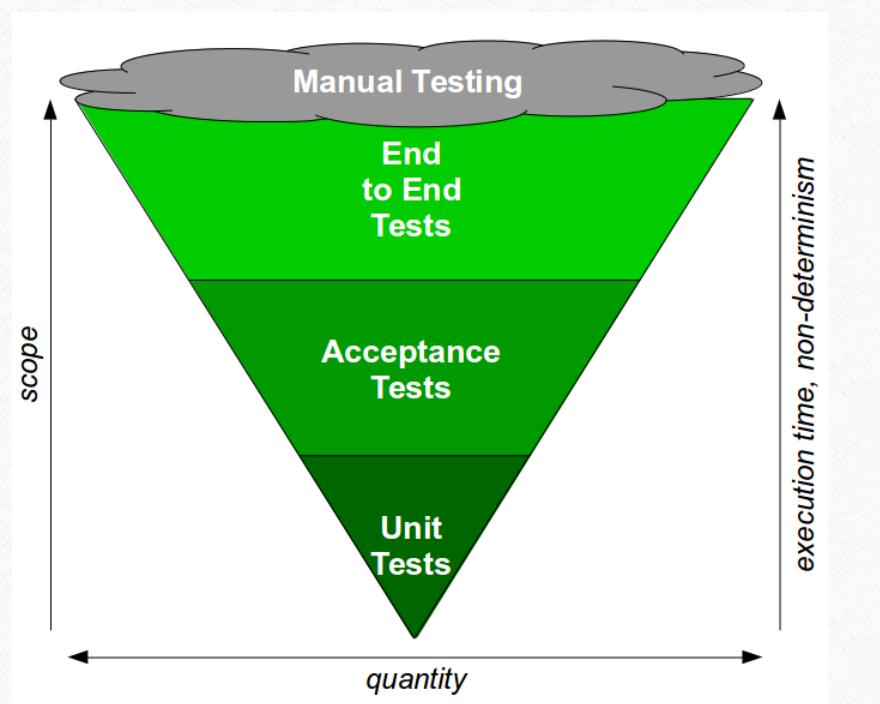
```
1 ▶ {  
2  
3   "id" : "100",  
4  
5   "pid": "1001",  
6  
7   "name": "Sachin",  
8  
9   "FullName": "Sachin Tendulkar"  
10  
11 }  
12
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 34ms Size: 303 B Sav

Pretty Raw Preview Visualize BETA JSON ▾

```
1 {  
2   "errorCode": 500,  
3   "errorMessage": "Please Contact Your System Administrator"  
4 }
```

E2E Testing

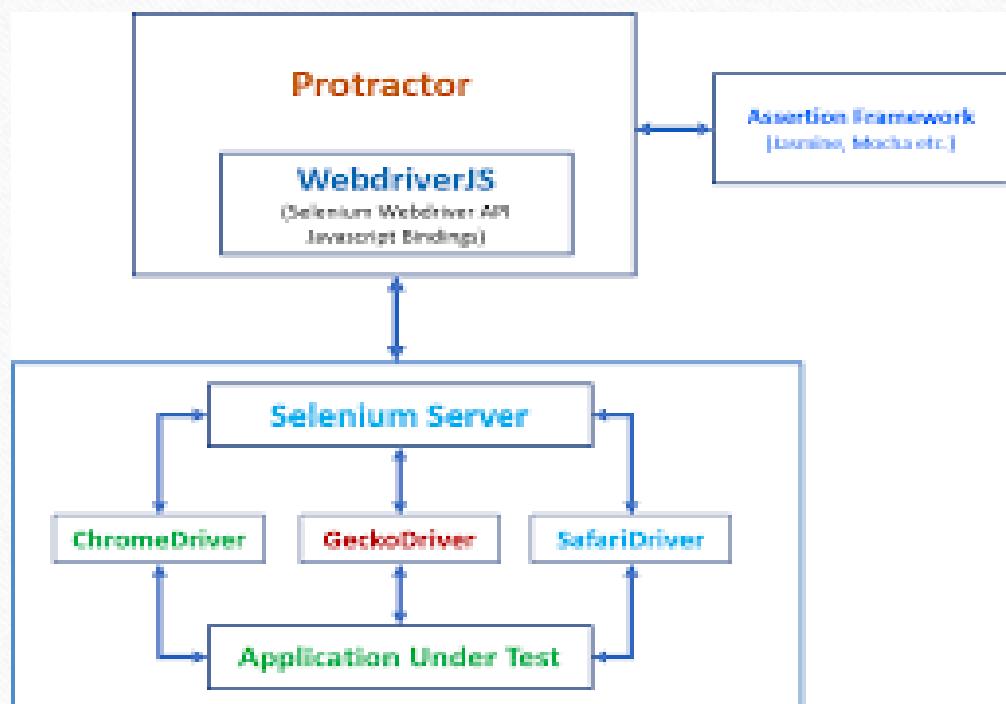


Unit vs E2E

Unit test	E2E test
APIs	UIs
Tiny	Large
White-box	Black-box
Isolated from the world	Make the pieces fit together

Browser automation!

Protractor

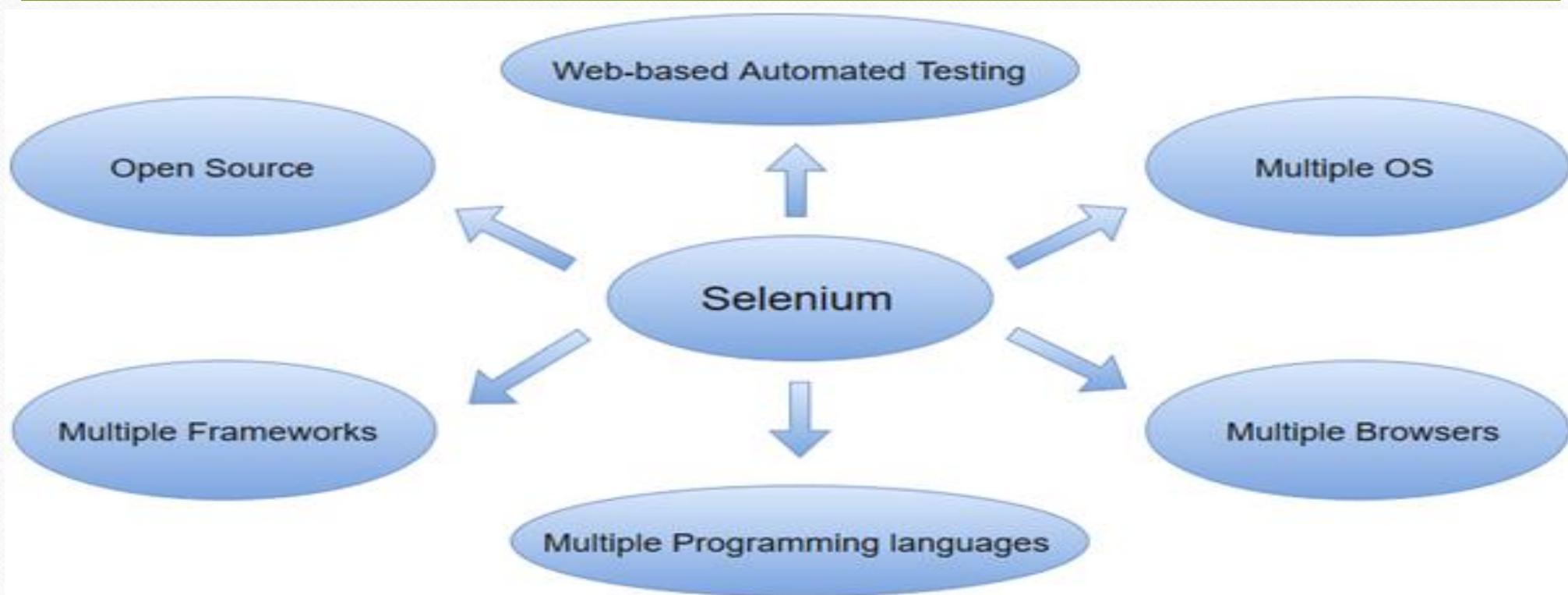


```
microsoft Windows [Version 6.1.7601]
pyright (c) 2020 Microsoft Corporation. All rights reserved.

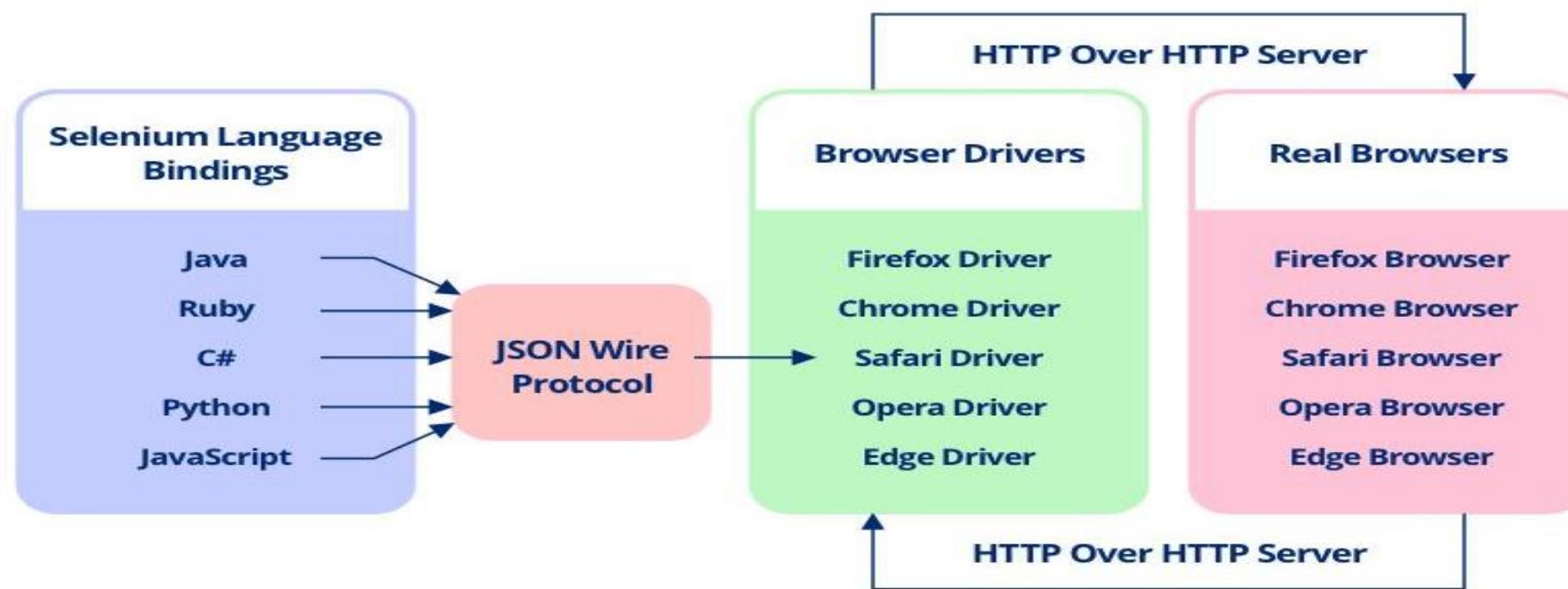
C:\Users\RE041943>npm install -g protractor
+utf-8-validate@1.2.1 install C:\Users\RE041943\AppData\Roaming\npm\node_modules\protractor\node_modules\selenium-webdriver\node_modules\ws\node_modules\utf-8-validate
node-gyp rebuild

C:\Users\RE041943\AppData\Roaming\npm\node_modules\protractor\node_modules\selenium-webdriver\node_modules\ws\node_modules\utf-8-validate> if not defined npm_config_node_gyp (node "C:\Program Files (x86)\nodejs\node_modules\npm\bin\node-gyp-bin\node-gyp.js" rebuild ) else (node "" rebuild)
```

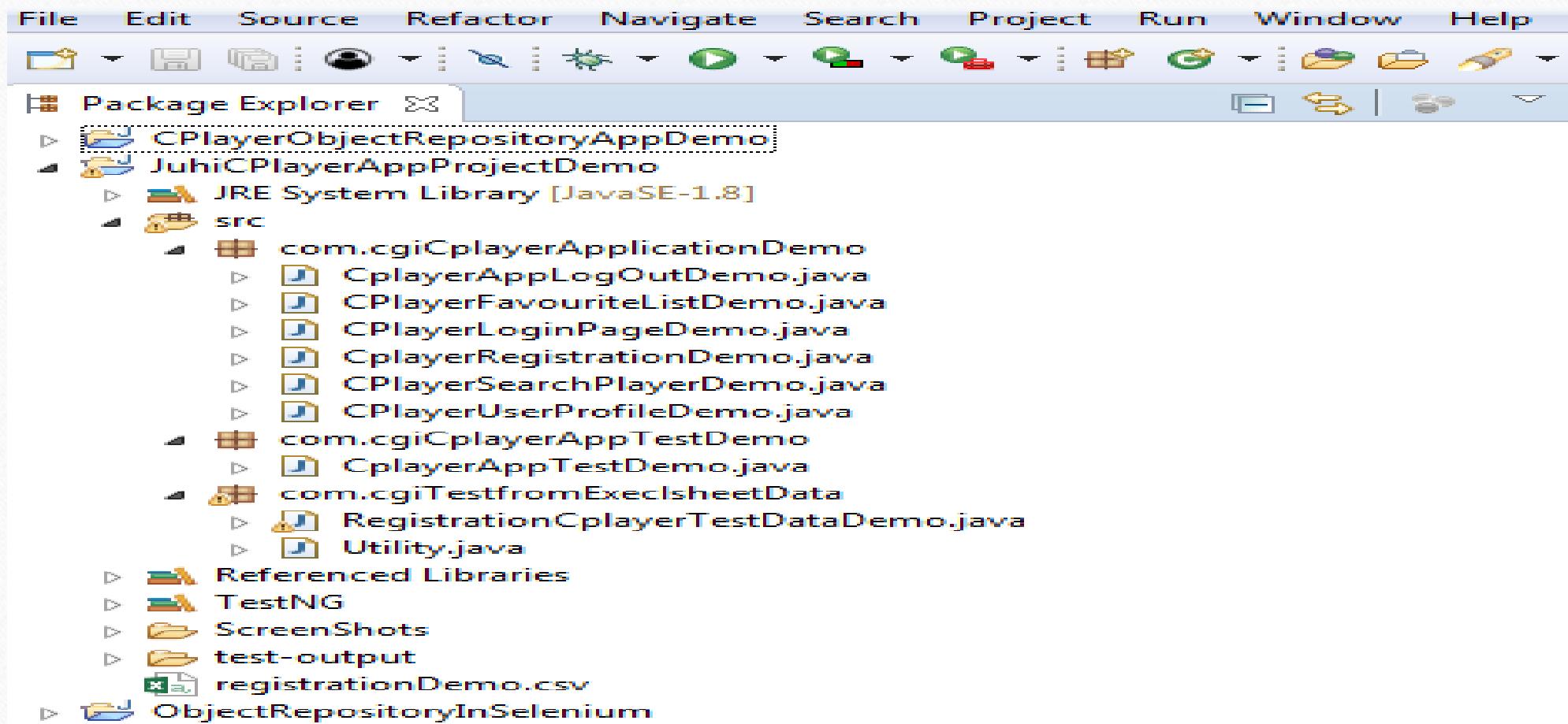
Selenium



Selenium WebDriver Architecture



Project Structure of Selenium



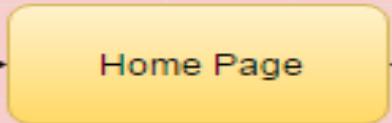
POM

Page Object Model

Web Pages



Page Objects

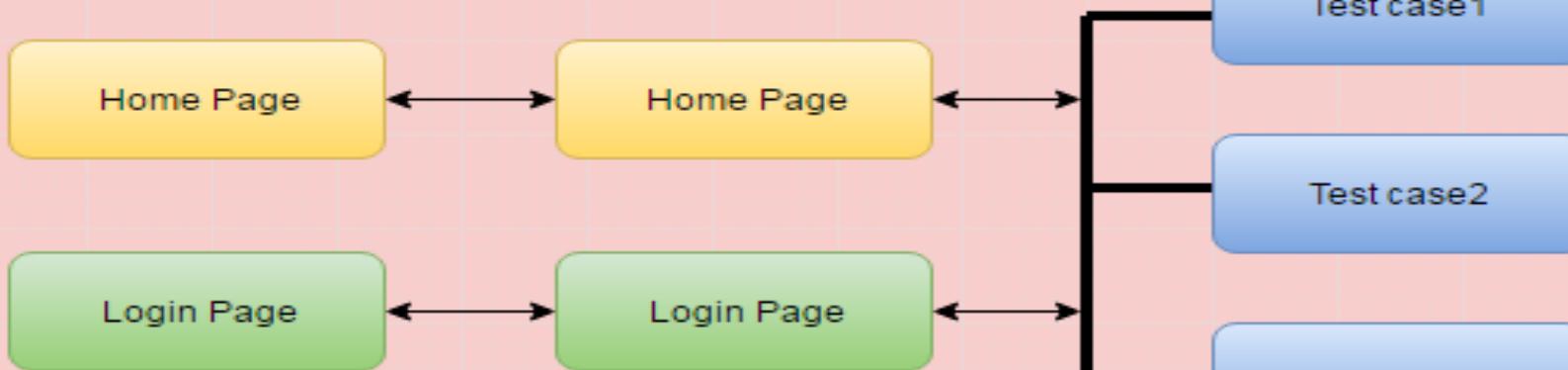


Test Scripts

Test case1

Test case2

Test case3



Retrieving data through Excel Sheet

```
package com.cgiTestfromExcelSheetData;

import java.io.File;

public class RegistrationCplayerTestDataDemo {

    WebDriver driver;
    String CSV_PATH = "D:\\\\CPlayerSeleniumPOM\\\\JuhiCPlayerAppProjectDemo\\\\registrationDemo.csv";

    @BeforeTest
    public void driverDemo() {

        System.setProperty("webdriver.chrome.driver",
                           "C:\\\\Selenium Examples Exercises\\\\Selenium java jar file\\\\chrome\\\\chromedriver.exe");

        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

    }

    @Test
    public void readData() throws Exception {
        CSVReader reader = new CSVReader(new FileReader(CSV_PATH));
        String[] csvCell;

        while ((csvCell = reader.readNext()) != null) {

            driver.get("http://localhost:4200/register");
            Utility.ScreenshotExample(driver, "Register Page");
            String fName = csvCell[0];
            String lName = csvCell[1];
            String usrid = csvCell[2];
            String Pwd = csvCell[3];
        }
    }
}
```

Continue..

```
//driver.findElement(By.xpath("//html/body/app-root/app-login/form/div/mat-card-actions/button[2]/span"));
driver.findElement(By.id("firstName")).sendKeys(fName);
Thread.sleep(3000);
driver.findElement(By.id("lastName")).sendKeys(lName);
Thread.sleep(3000);
driver.findElement(By.id("userId")).sendKeys(usrid);
Thread.sleep(3000);
driver.findElement(By.id("password")).sendKeys(Pwd);

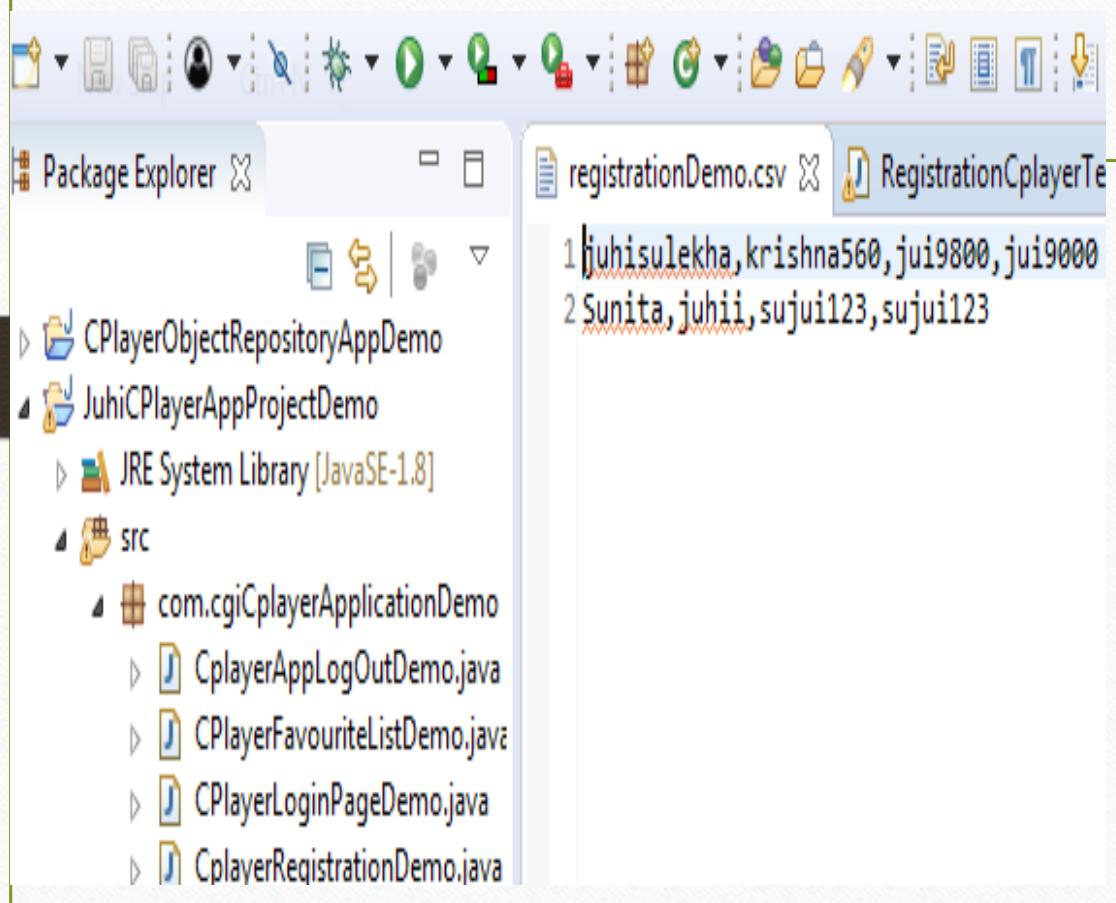
Thread.sleep(3000);
driver.findElement(By.cssSelector("body > app-root > app-register > mat-card-content > form > div:nth-child(7) > button.register-user.mat-raised-button")).click();
//driver.findElement(By.xpath("//html/body/app-root/app-login/form/div/mat-card-actions/button[2]/span")).click();
//driver.switchTo().alert().accept();
driver.findElement(By.xpath("//html/body/app-root/mat-toolbar/button[1]")).click();

driver.findElement(By.id("userId")).sendKeys(usrid);
Thread.sleep(3000);
driver.findElement(By.id("password")).sendKeys(Pwd);
Thread.sleep(3000);
driver.findElement(By.xpath("//html/body/app-root/app-login/form/div/mat-card-actions/button[1]/span"));
Thread.sleep(6000);
Utility.ScreenshotExample(driver, "Register Page1");
System.out.println("ScreenShot Taken Successfully");

}

reader.close();
driver.quit();
}
```

Result in Browser



User Profile Logout

User Profile

First Name :
Sunita

Last Name :
juhii

User ID : *
sujui123

Password : *
sujui123

Register UserProfile **Clear**

Screen Shots

Cricket Player Search Player Favourite List User Profile Logout

Cricket Player Search Player Favourite List User Profile Logout

Login User

User ID : *

suju123

Password : *

suju123

Login User

Register

Register User

First Name :

juhi

Last Name :

krishna

User ID : *

ju198

Password : *

ju190

Register User

Clear

Example of Object Repository

```
package com.cgiConfigurationCPlayer;

import java.io.File;

public class ConfigurationUtility {

    Properties proper;

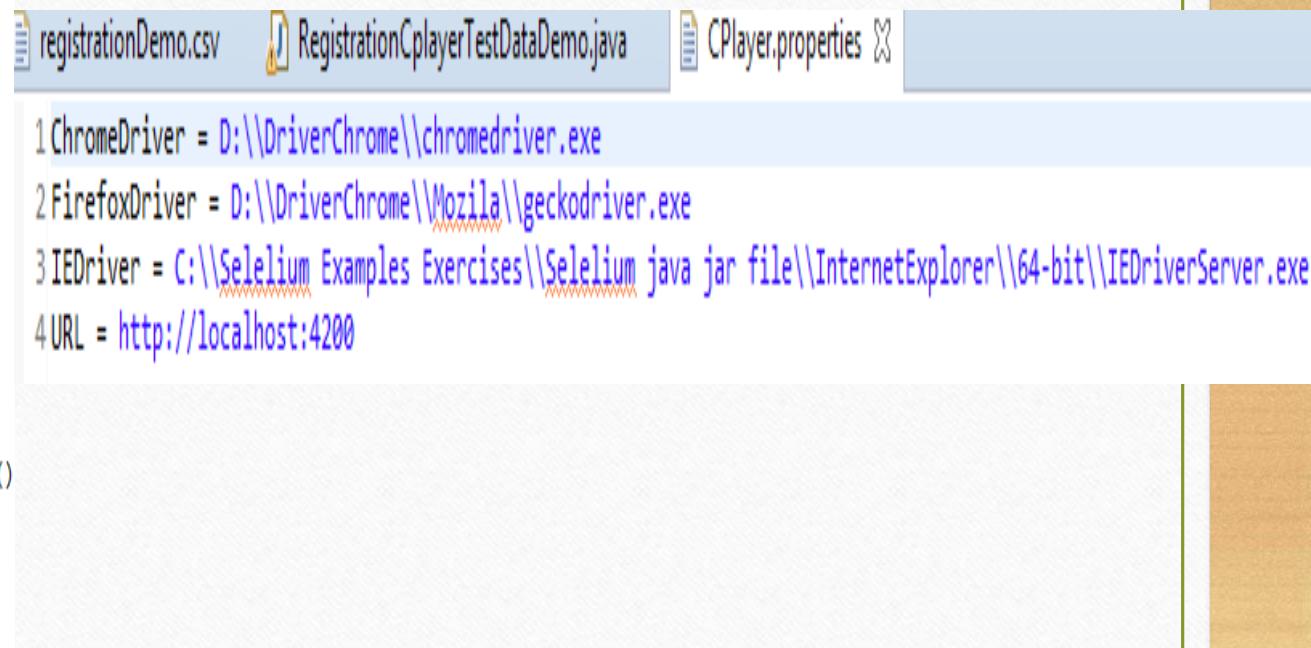
    public ConfigurationUtility() {
        try {
            File src = new File("D:\\CPlayerSeleniumPOM\\CPlayerObjectRepositoryAppDemo\\Configuration\\CPlayer.properties");
            FileInputStream fis = new FileInputStream(src);
            proper=new Properties();
            proper.load(fis);
        }
        catch(Exception e) {
            System.out.println("Exception:" +e.getMessage());
        }
    }

    public String demoGetChromePath() {
        String path = proper.getProperty("ChromeDriver");
        return path;
    }

    public String getAppUrl() {
        return proper.getProperty("URL");
    }
}
```

Continue..

```
package com.cgiCPlayerRepositoryDemo;  
  
import org.openqa.selenium.By;  
  
public class CPlayerLoginPageDemo {  
  
    @Test  
    public void tesChrome() throws Exception {  
        ConfigurationUtility cu = new ConfigurationUtility();  
        System.setProperty("webdriver.chrome.driver", cu.demoGetChromePath());  
        WebDriver driver = new ChromeDriver();  
  
        driver.manage().window().maximize();  
        driver.get(cu.getAppUrl());  
        driver.findElement(By.id("userId")).sendKeys("juhi123");  
        Thread.sleep(3000);  
        driver.findElement(By.id("password")).sendKeys("juhi123");  
        Thread.sleep(3000);  
        driver.findElement(By.xpath("//html/body/app-root/app-login/form/div/mat-card-actions/button[1]/span")).click()  
        Thread.sleep(4000);  
        driver.quit();  
    }  
}
```



Result in Chrome Browser

A screenshot of a web browser window titled "localhost:4200/players/search". The browser's address bar shows the URL. The page has a red header bar with the text "Cricket Player" and several navigation links: "Search Player", "Favourite List", "User Profile", and "Logout". Below the header is a search input field with a magnifying glass icon. The main content area is currently empty, showing only the placeholder text "Search player name...".

← → ⌛ ⓘ localhost:4200/players/search

apps Google Bookmarks Application Express... C Programming La... GeeksforGeeks | A c... Aptitude Questions... JavaScript Tutorial

Cricket Player Search Player Favourite List User Profile Logout

Search player name...

Functionality to be Tested

- Launching the Chrome Browser
- Opening Registration Page
- Login Page
- Search Player
- Add to Fav List
- Update the Content
- User Profile
- Logout the Application

Step1: Launching the Browser

```
3④ import org.openqa.selenium.WebDriver;
4
5 public class CplayerAppTestDemo {
6
7     // Global Declaration
8     WebDriver driver;
9
10    // Test annotation for testing
11    @Test
12
13    // Method for CPlayer Application Launch (Registration Page)
14    public void testCPlayerAppDemo() throws Exception {
15
16        // Setting ChromeBrowser driver path
17        System.setProperty("webdriver.chrome.driver",
18            "C:\\\\Selenium Examples Exercises\\\\Selenium java jar file\\\\chrome\\\\chromedriver.exe");
19        System.setProperty("webdriver.chrome.bin", "C:\\\\Program Files (x86)\\\\Google\\\\Chrome\\\\Application\\\\chrome.exe");
20        driver = new ChromeDriver();
21        // Application URL
22        driver.get("http://localhost:4200");
23
24        // Maximizing the Window Size
25        driver.manage().window().maximize();
26
27        CplayerRegistrationDemo obj1_crd = new CplayerRegistrationDemo(driver);
28        obj1_crd.clickRegisterButton();
29        obj1_crd.typeFirstName();
30        obj1_crd.typeLastName();
31        obj1_crd.typeUserId();
32        obj1_crd.typePassword();
33        obj1_crd.clickRegisterUserButton();
34
35        // Closing the Driver
36        driver.quit();
37    }
38}
```

Result in Chrome Browser

← → ⌛ ⓘ localhost:4200/login

🔍 ☆ SC

Apps Google Bookmarks Application Express... C Programming La... GeeksforGeeks | A c... Aptitude Questions... JavaScript Tutorial Citrix Applications

Cricket Player Search Player Favourite List User Profile Logout

Login User

User ID : *

Password : *

Step2 : Registration Page

```
19     // Test annotation for testing
20 @Test(priority = 1)
21
22 // Method for CPlayer Application Launch (Registration Page)
23 public void testCPlayerAppDemo() throws Exception {
24
25     // Setting ChromeBrowser driver path
26     System.setProperty("webdriver.chrome.driver",
27         "C:\\\\Selenium Examples Exercises\\\\Selenium java jar file\\\\chrome\\\\chromedriver.exe");
28     System.setProperty("webdriver.chrome.bin", "C:\\\\Program Files (x86)\\\\Google\\\\Chrome\\\\Application\\\\chrome.exe");
29     driver = new ChromeDriver();
30     // Application URL
31     driver.get("http://localhost:4200");
32
33     // Maximizing the Window Size
34     driver.manage().window().maximize();
35
36     CplayerRegistrationDemo obj1_crd = new CplayerRegistrationDemo(driver);
37     obj1_crd.clickRegisterButton();
38     obj1_crd.typeFirstName();
39     obj1_crd.typeLastName();
40     obj1_crd.typeUserId();
41     obj1_crd.typePassword();
42     obj1_crd.clickRegisterUserButton();
43
44     // Closing the Driver
45     driver.quit();
46 }
47 }
```

CPlayer Registration Page

localhost:4200/register

Cricket Player Search Player Favourite List User Profile Logout

Register User

First Name :
harshi

Last Name :
juhi

User ID : *
juhiharshi

Password : *
juhiharshi

Register User **Clear**

Step-3: Login Page

```
@Test(priority = 2)
// Method for Launching Login Page(CPlayer)
public void testCPlayerAppLoginDemo() throws Exception {

    // Setting ChromeBrowser driver path
    System.setProperty("webdriver.chrome.driver",
        "C:\\Selenium Examples Exercises\\Selenium java jar file\\chrome\\chromedriver.exe");
    System.setProperty("webdriver.chrome.bin", "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe");
    driver = new ChromeDriver();
    driver.get("http://localhost:4200");

    // Maximizing the Window Size
    driver.manage().window().maximize();

    Thread.sleep(5000);
    // Creating object of CPlayerLoginPageDemo class and impotring
    CPlayerLoginPageDemo obj2_clp = new CPlayerLoginPageDemo(driver);
    Thread.sleep(5000);
    obj2_clp.typeUserLoginId();
    Thread.sleep(4000);
    obj2_clp.typeUserLoginPassword();
    obj2_clp.clickLoginUserButton();
    Thread.sleep(5000);
    driver.quit();

}
```

Cplayer Login Page

← → ⌂ ⓘ localhost:4200/login

_apps Google Bookmarks Application Express... C Programming La... GeeksforGeeks | A c... Aptitude Questions... JavaScript Tutorial Citrix A

Cricket Player

Search Player

Favourite List

User Profile

Logout

Login User

User ID : *

juhiharshi

Password : *

juhiharshi

Login User

Register

CPlayer Search List Bar

localhost:4200/players/search

Apps Google Bookmarks Application Express... C Programming La... GeeksforGeeks | A c... Aptitude Questions... JavaScript Tutorial Citrix Applications

rohit sharma

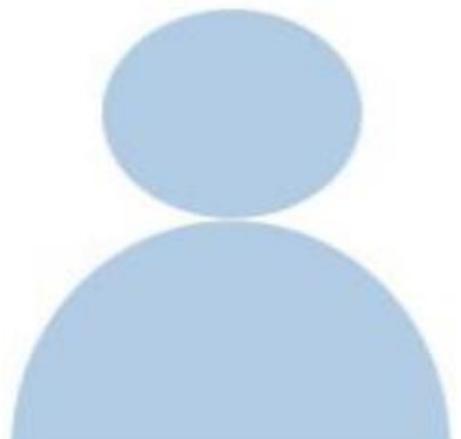
Rohit Sharma | Rohit Gurunath Sharma



Rohit | Rohit G. Sharma



Rohit | Rohit Sharma



Rohit Sharma | Rohit Parmod Sharma



[View Details](#) [View Details](#) [View Details](#) [View Details](#)

CPlayer Add to Fav List

localhost:4200/players/search

Apps Google Bookmarks Application Express... C Programming La... GeeksforGeeks | A c... Aptitude Questions... JavaScript Tutorial Citrix Applications

	League	10w	5w	4w	Econ	Ave	BBM	BBI	Wkts	Runs	Balls	Inns	Mat
Rohit S	ListA	0	0	1	5.10	38.73	4/28	4/28	30	1162	1367	70	239
Rohit Guru	First Class	0	0	1	3.22	47.16	5/85	4/41	24	1132	2104	64	81
	T20I's	0	0	0	9.97	113.00	1/22	1/22	1	113	68	9	65
	ODI's	0	0	0	5.21	64.37	2/27	2/27	8	515	593	38	170
	Test's	0	0	0	3.62	101.00	1/35	1/26	2	202	334	10	21

Add to FavouriteList Cancel

[View Details](#) [View Details](#) [View Details](#) [View Details](#)

Step4: Search Players

```
75@  @Test(priority = 3)
76    // Method for Searching Player by name add to favorite list
77    public void testCPlayerAppSearchDemo() throws Exception {
78
79        // Setting ChromeBrowser driver path
80        System.setProperty("webdriver.chrome.driver",
81                            "C:\\Selenium Examples Exercises\\Selenium java jar file\\chrome\\chromedriver.exe");
82        System.setProperty("webdriver.chrome.bin", "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe");
83        driver = new ChromeDriver();
84        driver.get("http://localhost:4200");
85        // Maximizing the Window Size
86        driver.manage().window().maximize();
87
88        CPlayerSearchPlayerDemo obj3_cspd = new CPlayerSearchPlayerDemo(driver);
89
90        obj3_cspd.typeUserLoginId();
91        Thread.sleep(4000);
92        obj3_cspd.typeUserLoginPassword();
93        obj3_cspd.clickLoginUserButton();
94        Thread.sleep(4000);
95        obj3_cspd.clickSearchPlayer();
96        Thread.sleep(4000);
97        obj3_cspd.typeSearchName();
98        Thread.sleep(5000);
99        obj3_cspd.clickSearchBar();
00        Thread.sleep(5000);
01        obj3_cspd.clickOnViewDetails();
02        Thread.sleep(5000);
03        obj3_cspd.clickOnAddToFavList();
04        Thread.sleep(5000);
05        // obj3_cspd.clickOnCancelButton();
06        driver.quit();
07    }
08}
```

CPlayer App Update Content

localhost:4200/players/favouritelist

	0	0	1	5.10	38.73	4/28	4/28	30	1162	1367	70	239
ListA	0	0	1	5.10	38.73	4/28	4/28	30	1162	1367	70	239
First Class	0	0	1	3.22	47.16	5/85	4/41	24	1132	2104	64	81
T20I's	0	0	0	9.97	11							
ODI's	0	0	0	5.21	64							
Test's	0	0	0	3.62	10							

Rohit Gurunath Sharma

Profile:

right-handed batsman and an occasional right-arm off break bowler. Rohit Gurunath Sharma is an Indian international cricketer who plays for Mumbai in domestic cricket and captains Mumbai Indians in the Indian Premier League as a right-handed batsman and an occasional right-arm off break bowler.

Update Cancel

Mahendra Singh Dhoni

Step-5: Update Content

```
// Method for Showing Favorite list here we can see our fav player details and
// also we can update
@Test(priority = 4)
public void testCPlayerAppFavListDemo() throws Exception {

    // Setting ChromeBrowser driver path
    System.setProperty("webdriver.chrome.driver",
        "C:\\Selenium Examples Exercises\\Selenium java jar file\\chrome\\chromedriver.exe");
    System.setProperty("webdriver.chrome.bin", "C:\\Program Files (x86)\\Google\\Application\\chrome.exe");
    driver = new ChromeDriver();
    driver.navigate().to("http://localhost:4200");
    // Maximizing the Window Size
    driver.manage().window().maximize();

    // Creating Object
    CPlayerFavouriteListDemo obj4_cfld = new CPlayerFavouriteListDemo(driver);

    // Calling Method
    obj4_cfld.typeUserLoginId();
    Thread.sleep(4000);
    obj4_cfld.typeUserLoginPassword();
    obj4_cfld.clickLoginUserButton();
    Thread.sleep(3000);
    obj4_cfld.clickOnFavouriteListButton();
    Thread.sleep(3000);
    obj4_cfld.clickonUpdateButton();
    Thread.sleep(3000);
    obj4_cfld.typeupdateContent();
    Thread.sleep(3000);
    obj4_cfld.clickOnUpdate();
    Thread.sleep(4000);
    driver.quit();
}
```

Step-6: User Profile

- Registration Page
- Login Page

User Profile-Registration Page

localhost:4200/userprofile

Apps Google Bookmarks Application Express... C Programming La... GeeksforGeeks | A c... Aptitude Questions... JavaScript Tutorial Citrix Applications

Cricket Player Search Player Favourite List User Profile Logout

User Profile

First Name :
Harshitha

Last Name :
P

User ID : *
harshi123

Password : *
harshi123

Register UserProfile **Clear**

User Profile-Login Page

The screenshot shows a web browser window with the following details:

- Address Bar:** localhost:4200/login
- Toolbar:** Back, Forward, Refresh, Home, and several bookmarks including "Google", "Bookmarks", "Application Express...", "C Programming La...", "GeeksforGeeks | A c...", "Aptitude Questions...", "JavaScript Tutorial", and "Citrix Applications".
- Header Bar:** A red navigation bar with the following items: "Cricket Player" (highlighted in white), "Search Player", "Favourite List", "User Profile", and "Logout".
- Content Area:** A white form titled "Login User". It contains two input fields:
 - "User ID : *" followed by the value "harshi123".
 - "Password : *" followed by the value "harshi123".
- Buttons:** Two purple buttons at the bottom: "Login User" and "Register".

User Profile-Registration & Login

```
143 @Test(priority = 5)
144 public void testUserProfile() throws InterruptedException {
145     // Setting ChromeBrowser driver path
146     System.setProperty("webdriver.chrome.driver",
147         "C:\\Selenium Examples Exercises\\Selenium java jar file\\chrome\\chromedriver.exe");
148     System.setProperty("webdriver.chrome.bin", "C:\\Program Files (x86)\\Google\\Chrome\\Application\\chrome.exe");
149     driver = new ChromeDriver();
150     driver.navigate().to("http://localhost:4200");
151     // Maximizing the Window Size
152     driver.manage().window().maximize();
153
154     // Creating object
155     CPlayerUserProfileDemo obj5_cupd = new CPlayerUserProfileDemo(driver);
156     // Calling Methods
157     obj5_cupd.typeUserLoginId();
158     Thread.sleep(4000);
159     obj5_cupd.typeUserLoginPassword();
160     obj5_cupd.clickLoginUserButton();
161     Thread.sleep(3000);
162     obj5_cupd.clickOnUserProfileHeader();
163     Thread.sleep(3000);
164     obj5_cupd.typeFirstName();
165     Thread.sleep(3000);
166     obj5_cupd.typeLastName();
167     Thread.sleep(3000);
168     obj5_cupd.typeUserId();
169     Thread.sleep(3000);
170     obj5_cupd.typePassword();
171     Thread.sleep(3000);
172     obj5_cupd.clickRegisterUserButton();
173     Thread.sleep(7000);
174     driver.quit();
175 }
176 }
```

Step-7: Logout Page

```
// Method for Logout the Application
@Test(priority = 6)
public void testLogout() throws InterruptedException {
    System.setProperty("webdriver.chrome.driver",
        "C:\\\\Selenium Examples Exercises\\\\Selenium java jar file\\\\chrome\\\\chromedriver.exe");
    System.setProperty("webdriver.chrome.bin", "C:\\\\Program Files (x86)\\\\Google\\\\Chrome\\\\Application\\\\chrome.exe");
    driver = new ChromeDriver();
    driver.navigate().to("http://localhost:4200");

    // Maximizing the Window Size
    driver.manage().window().maximize();

    // Creating Object
    CplayerAppLogOutDemo obj6_cald = new CplayerAppLogOutDemo(driver);
    // Method Calling

    Thread.sleep(5000);
    obj6_cald.typeUserLoginId();
    obj6_cald.typeUserLoginPassword();
    Thread.sleep(5000);
    obj6_cald.clickLoginUserButton();
    Thread.sleep(5000);

    obj6_cald.clickOnLogOutButton();
    Thread.sleep(5000);
    driver.quit();
}
```

CPlayer Logout Page

localhost:4200/login

Apps Google Bookmarks Application Express... C Programming La... GeeksforGeeks | A c... Aptitude Questions... JavaScript Tutori...

Cricket Player Search Player Favourite List User Profile Logout

Login User

User ID : *

Password : *

Login User

Register

