

Développeur d'application - Android

String Projet 2 =
" Analysez les tests de l'application MagIC GitHub";

String P2 ="P2_PILI_Xavier";

SOMMAIRE

int Git	= 1 ;
int Tests initiaux qui Échouent	= 2 ;
int les Corrections de Codes	= 3 ;
int Tests finaux en Réussite	= 4 ;
int les Codes Corrigés	= 5 ;
int les Livrables	= 6 ;
int les Liens Url Png	= 7 ;
String merci = "MERCI"	= 8 ;



git pull



git push



git pull

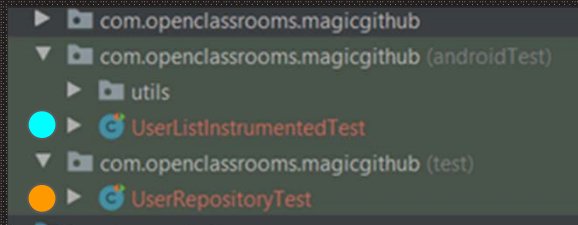


git push



int Git

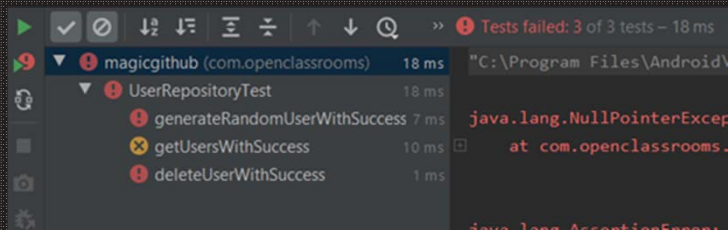
= 1;



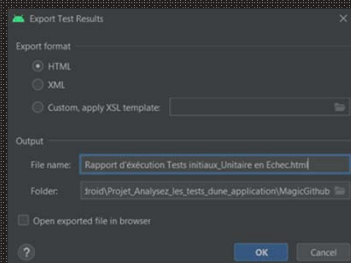
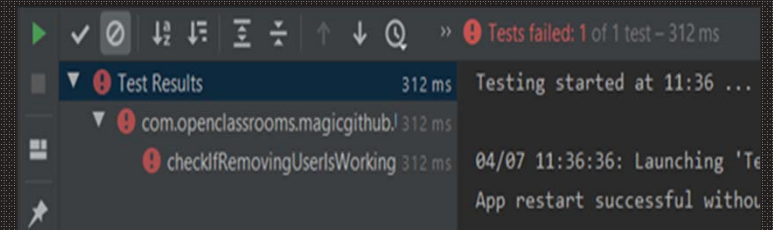
int Tests initiaux Échouent = 2 ;

Tests Initiaux ;

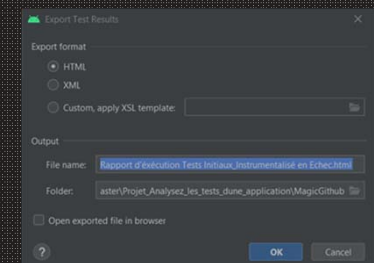
● Tests Unitaire ;

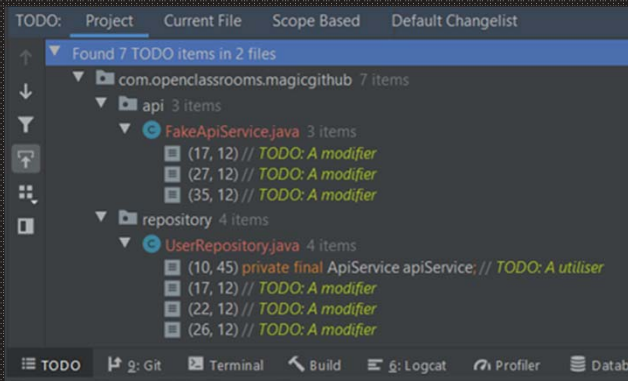


● Tests Instrumentalisés ;



Exports Rapports d'exécutions Initiaux ;





int les Corrections de Codes = 3 ;

// TODO: à modifier

public class FakeApiService implements ApiService {

```

    public List<User> getUsers() {
        // TODO: A modifier
        return null;
    }

    /**
     * Generate a random {@Link User} and add it {@Link FakeApiService#users} List.
     * This user must be get from the {@Link FakeApiServiceGenerator#FAKE_USERS_RANDOM} List.
     */
    @Override
    public void generateRandomUser() {
        // TODO: A modifier
    }

    /**
     * Delete a {@Link User} from the {@Link FakeApiService#users} List.
     */
    @Override
    public void deleteUser(User user) {
        // TODO: A modifier
    }
}

```

public class UserRepository {

```

    private final ApiService apiService; // TODO: A utiliser

    public UserRepository(ApiService apiService) { this.apiService = apiService; }

    public List<User> getUsers() {
        // TODO: A modifier
        return null;
    }

    public void generateRandomUser() {
        // TODO: A modifier
    }

    public void deleteUser(User user) {
        // TODO: A modifier
    }
}

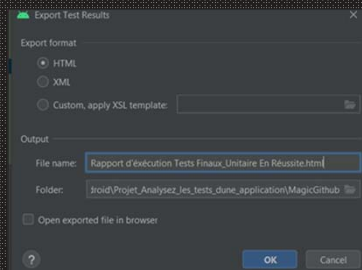
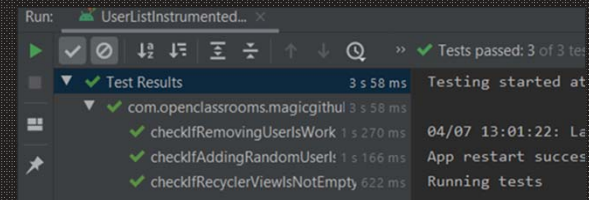
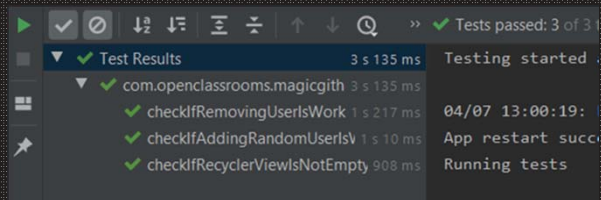
```


int Tests finaux en Réussite = 4 ;

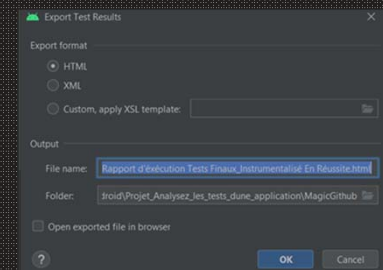
Tests Finaux ;

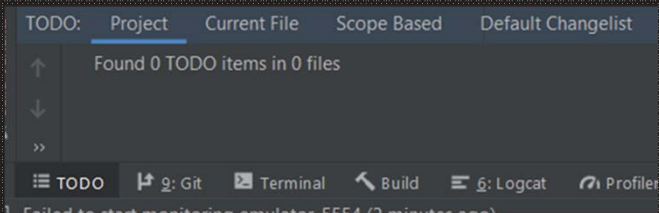
Tests Unitaire ;

Tests Instrumentalisés ;



Exports Rapports d'exécutions Finaux ;





int les Codes Corrigés

= 5 ;

Méthodes Complétées ;

Paramètres passés ;

// TODO: à modifier → DONE //

Code intended ;



```
@Override
public List<User> getUsers() {
    // DONE //
    return users;
}

/**
 * Generate a random {@Link User} and add it {@Link FakeApiService#users} List.
 * This user must be get from the {@Link FakeApiServiceGenerator#FAKE_USERS_RANDOM} List.
 */
@Override
public void generateRandomUser() {
    // DONE //
    this.getUsers().add(User.random());
}

/**
 * Delete a {@Link User} from the {@Link FakeApiService#users} List.
 */
@Override
public void deleteUser(User user) {
    // DONE //
    users.remove(user);
}
```

```
public class UserRepository {

    private final ApiService apiService; // A utiliser //

    public UserRepository(ApiService apiService) { this.apiService = apiService; }

    public List<User> getUsers() {
        // DONE//
        return apiService.getUsers();
    }

    public void generateRandomUser() {
        // DONE //
        apiService.generateRandomUser();
    }

    public void deleteUser(User user) {
        // DONE //
        apiService.deleteUser(user);
    }
}
```

int les Livrables

= 6

;

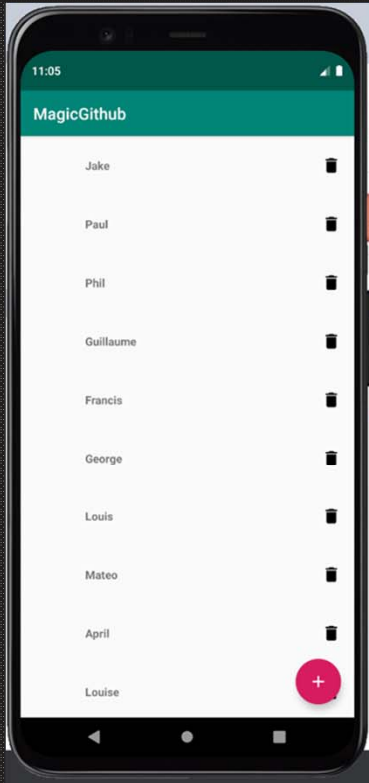
Rapports d'exécution (unitaire et instrumentalisé) des tests initiaux (qui doivent indiquer que les tests échouent) au format HTML ;



Rapports d'exécution (unitaire et instrumentalisé) des tests finaux (qui doivent indiquer que les tests réussissent) au format HTML ;



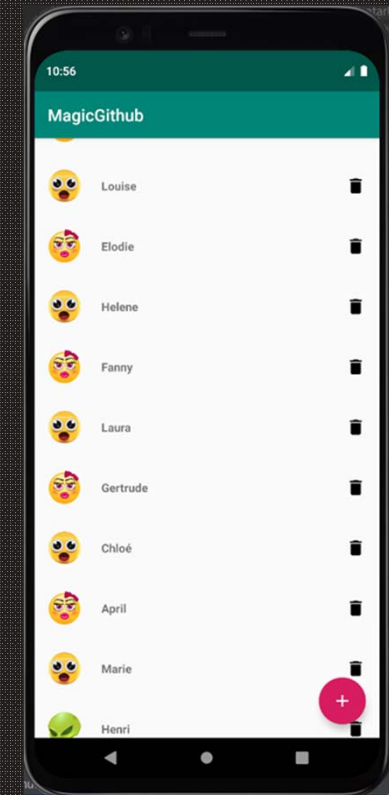
int les Liens Url Png = 7;



```
public abstract class FakeApiServiceGenerator {  
  
    static List<User> generateUsers() {  
        return new ArrayList<>{FAKE_USERS};  
    }  
  
    public static List<User> FAKE_USERS = Arrays.asList(  
        new User( id: "001", login: "Jake", avatarUrl: "https://cdn.pixabay.com/photo/2013/07/12/18/35/alien-153542_960_720.png"),  
        new User( id: "002", login: "Paul", avatarUrl: "https://cdn.pixabay.com/photo/2016/08/21/18/48/emoticon-1610518_1280.png"),  
        new User( id: "003", login: "Phil", avatarUrl: "https://cdn.pixabay.com/photo/2016/08/11/10/23/emoji-1585265_1280.png"),  
        new User( id: "004", login: "Guillaume", avatarUrl: "https://cdn.pixabay.com/photo/2016/08/11/10/23/emoji-1585265_1280.png"),  
    );  
}
```



Nouveaux liens url pour png



```
public class test {  
    public static void main(String[] args) {  
        String merci = "MERCI";  
        System.out.println(merci);  
    }  
}
```