

Rapport Complet des Tests de l'application Kanban

Réalisé Par:

Mouad Nadzi

Asmaa El Bouazzaoui

Mohamed AlMahdy Baâkrim

Abderrahmane Bensalek

Aya El Feddani

2024/2025

Sous la direction de :

M. Driss ESSABAR

Contents

1	Tests WebDriver	3
1.1	Objectifs des Tests	3
1.2	Plan de Test	3
1.2.1	1. Inscription	3
1.2.2	2. Connexion	4
1.2.3	3. Création de Board	5
1.2.4	4. Ajout de Sprints	7
1.2.5	5. Ajout de Tâches	8
1.3	Résultats des Tests	8
1.3.1	Inscription	8
1.3.2	Connexion	13
1.3.3	Création de Board	14
1.3.4	Ajout de Sprints	15
1.3.5	Ajout de Tâches	15
1.4	Conclusions	15
2	Tests JMeter	15
2.1	Objectifs	15
2.2	Groupe de Threads	15
2.3	Résultats :	16
2.3.1	Add User	16
2.3.2	Fetch Users	17
2.3.3	Authenticate	19
2.3.4	Add Board	20
2.3.5	Fetch Boards	22
2.3.6	Add Sprint	24
2.3.7	Fetch Sprints	25
2.3.8	Add Task	27
2.3.9	Fetch Tasks	28
2.4	Conclusion	30

Introduction

Ce rapport documente les résultats des tests fonctionnels et de performance réalisés sur l'application Kanban. Les tests ont été effectués à l'aide de Selenium WebDriver pour les scénarios fonctionnels et JMeter pour les scénarios de charge et de performance.

1 Tests WebDriver

Les tests WebDriver visent à valider les fonctionnalités critiques de l'application Kanban telles que l'inscription, la connexion, la création de tableaux, et l'ajout de sprints et de tâches.

1.1 Objectifs des Tests

- Inscription d'un nouvel utilisateur : Vérifier que l'application permet aux nouveaux utilisateurs de s'inscrire correctement.
- Connexion : Confirmer que les utilisateurs peuvent se connecter avec des informations valides.
- Création de Board : Tester la fonctionnalité de création de tableaux pour la gestion des projets.
- Ajout de Sprints : Valider que les utilisateurs peuvent ajouter des sprints avec des informations correctes.
- Ajout de Tâches : Tester la création de tâches avec la sélection d'utilisateurs et de sprints.

1.2 Plan de Test

1.2.1 1. Inscription

Scénarios Testés :

- Champs vides : Vérification qu'une erreur est affichée si l'utilisateur tente de s'inscrire sans remplir tous les champs.
- Utilisateur existant : Validation qu'un message d'erreur est affiché si le nom d'utilisateur ou l'email existe déjà.
- Email invalide : S'assurer qu'une adresse email incorrecte est rejetée.

- Mot de passe court : Confirmation qu'un mot de passe inférieur à 6 caractères est refusé.
- Succès de l'inscription : Vérification que l'utilisateur est redirigé vers la page d'accueil après une inscription réussie.

1.2.2 2. Connexion

Scénarios Testés :

- Champs vides : Validation qu'une erreur s'affiche si les champs de connexion ne sont pas remplis.
- Identifiants invalides : Vérification qu'un message d'erreur est affiché si les informations de connexion sont incorrectes.
- Succès de la connexion : S'assurer que l'utilisateur est redirigé vers la page des boards après une connexion réussie.

```

1 package web;
2
3 import org.junit.jupiter.api.AfterEach;
4
5
6
7
8 public class LoginTest {
9     private WebDriver driver;
10    private WebDriverWait wait;
11
12    @BeforeEach
13    public void setup() {
14        System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");
15        driver = new ChromeDriver();
16        wait = new WebDriverWait(driver, Duration.ofSeconds(10));
17    }
18
19    @AfterEach
20    public void teardown() {
21        if (driver != null) {
22            driver.quit();
23        }
24    }
25
26    @Test
27    public void testSignupProcess() {
28        // Open the home page
29        driver.get("http://localhost:4200/home");
30
31        // Click the "Let's explore ..." button
32        WebElement exploreButton = wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector(".btn.sign-in")));
33        exploreButton.click();
34
35        // Wait for the login/signup page to load
36        wait.until(ExpectedConditions.urlContains("/login"));
37
38        // Test Case 1: Empty Fields
39        registerUser("", "", "");
40        validateErrorMessageById("signupErrorMessage", "All fields are required.");
41    }
42
43
44
45
46
47
48
49
50
51

```

```

52 // Test Case 2: Duplicate username/email
53 registerUser("existingUser", "existing@example.com", "password123");
54 validateErrorMessageById("signupErrorMessage", "Username or email already exists. Please try again.");
55
56 // Test Case 3: Invalid email
57 fieldFiller("newUser", "invalid-email", "password123");
58 validateErrorMessageById("emailError", "Please enter a valid email address.");
59
60 // Test Case 4: Invalid password
61 fieldFiller("newUser", "valid@example.com", "short");
62 validateErrorMessageById("passwordError", "Password must be at least 6 characters long.");
63
64 // Test Case 5: Successful registration
65 registerUser("uniqueUser", "unique@example.com", "password123");
66 handleAlert(); // Automatically handle the alert
67 wait.until(ExpectedConditions.urlContains("/home"));
68 assertTrue(driver.getCurrentUrl().contains("/home"));
69
70 WebElement exploreButton2 = wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector(".btn.sign-in")));
71 exploreButton2.click();
72
73 WebElement loginButton = wait.until(ExpectedConditions.elementToBeClickable(By.id("loginLabel")));
74 loginButton.click();
75
76 login("", "");
77 validateErrorMessageById("loginErrorMessage", "All fields are required.");
78
79 login("nonexistent", "wrong");
80 validateErrorMessageById("loginErrorMessage", "Username or password incorrect. Please try again.");
81
82 login("uniqueUser", "password123");
83 wait.until(ExpectedConditions.urlContains("/boards"));
84 assertTrue(driver.getCurrentUrl().contains("/boards"));
85 }
86
87 private void registerUser(String username, String email, String password) {
88 // locate fields and fill the form using IDs

```

```

89 // locate fields and fill the form using IDs
90 WebElement usernameField = wait.until(ExpectedConditions.elementToBeClickable(By.id("signupUsername")));
91 WebElement emailField = driver.findElement(By.id("signupEmail"));
92 WebElement passwordField = driver.findElement(By.id("signupPassword"));
93 WebElement signUpButton = driver.findElement(By.id("signUpButton"));
94
95 // Clear and enter values
96 usernameField.clear();
97 usernameField.sendKeys(username);
98 emailField.clear();
99 emailField.sendKeys(email);
100 passwordField.clear();
101 passwordField.sendKeys(password);
102
103 // Click sign up
104 signUpButton.click();
105
106 private void login(String username, String password) {
107 // locate fields and fill the form using IDs
108 WebElement usernameField = wait.until(ExpectedConditions.elementToBeClickable(By.id("loginUsername")));
109 WebElement passwordField = driver.findElement(By.id("loginPassword"));
110 WebElement login = driver.findElement(By.id("loginButton"));
111
112 // Clear and enter values
113 usernameField.clear();
114 usernameField.sendKeys(username);
115 passwordField.clear();
116 passwordField.sendKeys(password);
117
118 // Click sign up
119 login.click();
120
121 private void handleAlert(String username, String email, String password) {
122 // locate fields and fill the form using IDs
123 WebElement usernameField = wait.until(ExpectedConditions.elementToBeClickable(By.id("signupUsername")));
124 WebElement emailField = driver.findElement(By.id("signupEmail"));

```

```

125 // locate fields and fill the form using IDs
126 WebElement passwordField = driver.findElement(By.id("signupPassword"));
127
128 // Clear and enter values
129 usernameField.clear();
130 usernameField.sendKeys(username);
131 emailField.clear();
132 emailField.sendKeys(email);
133 passwordField.clear();
134 passwordField.sendKeys(password);
135
136 private void validateErrorMessage(String errorId, String expectedMessage) {
137 WebElement errorMessage = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id(errorId)));
138 assertTrue(errorMessage.getText().contains(expectedMessage), "Expected error message not found");
139
140 private void handleAlert() {
141 // wait for the alert to appear and then accept it
142 Alert alert = wait.until(ExpectedConditions.alertIsPresent());
143 alert.accept();
144 }

```

Figure 1: Voici le code .

1.2.3 3. Création de Board

Processus :

- Naviguer vers la page des boards après connexion.
- Cliquer sur le bouton Ajouter un Board .
- Remplir le formulaire avec un nom de board et valider.

- Vérifier que le nouveau board est visible dans la liste.

```

1 package web;
2
3 import org.junit.jupiter.api.AfterEach;
4
5 public class BoardTest {
6
7     private WebDriver driver;
8     private WebDriverWait wait;
9
10    @BeforeEach
11    public void setup() {
12        System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");
13        driver = new ChromeDriver();
14        wait = new WebDriverWait(driver, Duration.ofSeconds(10));
15    }
16
17    @AfterEach
18    public void teardown() {
19        if (driver != null) {
20            driver.quit();
21        }
22    }
23
24    @Test
25    public void testBoardCreate() {
26        // Open the home page
27        driver.get("http://localhost:4200/home");
28
29        // Click the "let's explore ..." button
30        WebElement exploreButton = wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector(".btn.sign-in")));
31        exploreButton.click();
32
33        // Wait for the login/signin page to load
34        wait.until(ExpectedConditions.urlContains("/login"));
35
36        WebElement loginButton = wait.until(ExpectedConditions.elementToBeClickable(By.id("loginlabel")));
37
38        loginButton.click();
39
40        login("reguser", "password123");
41        wait.until(ExpectedConditions.urlContains("/boards"));
42        assertTrue(driver.getCurrentUrl().contains("/boards"));
43
44        WebElement boardAddButton = wait.until(ExpectedConditions.elementToBeClickable(By.id("boardAddButton")));
45        boardAddButton.click();
46        wait.until(ExpectedConditions.urlContains("/addboard"));
47        assertTrue(driver.getCurrentUrl().contains("/addboard"));
48
49        addBoard("test board", "description of test board");
50        wait.until(ExpectedConditions.urlContains("/boards"));
51        assertTrue(driver.getCurrentUrl().contains("/boards"));
52    }
53
54    private void login(String username, String password) {
55        // Locate fields and fill the form using IDs
56        WebElement usernameField = wait.until(ExpectedConditions.elementToBeClickable(By.id("loginUsername")));
57        WebElement passwordField = driver.findElement(By.id("loginPassword"));
58        WebElement login = driver.findElement(By.id("loginbutton"));
59
60        // Clear and enter values
61        usernameField.clear();
62        usernameField.sendKeys(username);
63        passwordField.clear();
64        passwordField.sendKeys(password);
65
66        // Click sign up
67        login.click();
68    }
69
70    private void addBoard(String nameVal, String descVal) {
71        // Locate fields and fill the form using IDs
72        WebElement name = wait.until(ExpectedConditions.elementToBeClickable(By.id("name")));
73        WebElement desc = driver.findElement(By.id("description"));
74
75        // Clear and enter values
76        name.clear();
77        name.sendKeys(nameVal);
78        desc.clear();
79        desc.sendKeys(descVal);
80
81        // Click sign up
82        add.click();
83    }
84 }
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102

```

1.2.4 4. Ajout de Sprints

Procesus :

- Naviguer vers la page des sprints depuis la barre latérale.
- Cliquer sur le bouton `Ajouter un Sprint`.
- Remplir le formulaire avec les informations suivantes :
 - Nom du sprint.
 - Date de début (format : yyyy-MM-dd).
 - Date de fin (format : yyyy-MM-dd).
- Soumettre le formulaire et recharger la page pour vérifier que le sprint est ajouté.

```

1 package web;
2
3 import org.junit.Jupiter.api.AfterEach;
4
5 public class SprintTest {
6
7     private WebDriver driver;
8     private WebDriverWait wait;
9
10    @BeforeEach
11    public void setup() {
12        System.setProperty("webdriver.chrome.driver", "C:\\\\chromedriver.exe");
13        driver = new Chromedriver();
14        wait = new WebDriverWait(driver, Duration.ofSeconds(10));
15    }
16
17    @AfterEach
18    public void teardown() {
19        if (driver != null) {
20            driver.quit();
21        }
22    }
23
24    @Test
25    public void testSprintCreate() {
26        // Open the home page
27        driver.get("http://localhost:4200/home");
28
29        // Click the "let's explore ..." button
30        WebElement exploration = wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector("#btn_signin")));
31        exploration.click();
32
33        // Wait for the login/signup page to load
34        wait.until(ExpectedConditions.urlContains("/login"));
35
36        // Perform login
37        WebElement loginLabel = wait.until(ExpectedConditions.elementToBeClickable(By.id("loginLabel")));
38        loginLabel.click();
39    }
40 }

```

```

144  input("Enter a command: ")
145
146  # wait for navigation to the boards page
147  wait_until(page.url().contains('boards'), timeout=10)
148  mouse.click(page.get_by_text('boards'))
149
150  # navigate to the sprint section
151  wait_until(page.url().contains('boards/sprint'), timeout=10)
152  mouse.click(page.get_by_text('sprint'))
153
154  # wait until a sprint is created
155  wait_until(page.url().contains('boards/sprint/1'), timeout=10)
156  mouse.click(page.get_by_text('1'))
157
158  # wait for the sprint page
159  wait_until(page.url().contains('boards/sprint/1'), timeout=10)
160  mouse.click(page.get_by_text('1'))
161
162  # add a new sprint
163  add_sprint = wait_until(page.url().contains('boards/sprint/1'), timeout=10)
164  add_sprint.click()
165
166  # add a new sprint
167  wait_until(page.url().contains('boards/sprint/1'), timeout=10)
168  mouse.click(page.get_by_text('add new sprint'))
169
170  # Add a new sprint
171  add_sprint_text = "01.12.2020", "01.12.2020"
172  add_sprint_text.click()
173
174  # verify selection by the sprint page
175  wait_until(page.url().contains('boards/sprint/1'), timeout=10)
176  mouse.click(page.get_by_text('1'))
177
178  # Add the page title, changes are reflected
179  driver.navigate().refresh()
180
181
182  @pytest.mark.parametrize('page_number', [
183      1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 74
```

```

80 WebElement passwordField = driver.findElement(By.id("loginPassword"));
81 WebElement login = driver.findElement(By.id("loginbutton"));
82 // Clear and enter values
83 usernameField.clear();
84 usernameField.sendKeys(username);
85 passwordField.clear();
86 passwordField.sendKeys(password);
87
88 // Click login
89 login.click();
90
91 }
92
93 private void addTask(String nameVal, String startDate, String endDate) {
94     // Locate fields and fill the form using JS
95     WebElement name = wait.until(ExpectedConditions.elementToBeClickable(By.id("name")));
96     WebElement startDate = driver.findElement(By.id("startdate"));
97     WebElement endDate = driver.findElement(By.id("enddate"));
98     WebElement add = driver.findElement(By.id("addTaskbutton"));
99
100     // Clear and enter values
101     name.clear();
102     name.sendKeys(nameVal);
103     startDate.clear();
104     startDate.sendKeys(startDate);
105     endDate.clear();
106     endDate.sendKeys(endDate);
107
108     // Click add sprint button
109     add.click();
110 }

```

1.2.5 5. Ajout de Tâches

Processus :

- Naviguer vers la page des tâches.
- Cliquer sur le bouton Ajouter une Tâche .
- Remplir le formulaire avec :
 - Nom de la tâche.
 - Description.
 - Statut (TODO, INPROGRESS, DONE). Estimation en heures.
- Utilisateurs assignés.
- Sprints associés.
 - Valider et confirmer que la tâche apparaît dans la liste des tâches.

1.3 Résultats des Tests

1.3.1 Inscription

- Champs vides : L'application a affiché un message All fields are required. .
- Utilisateur existant : Un message Username or email already exists. Please try again. a été affiché.


```

1 package web;
2
3 import org.junit.jupiter.api.AfterEach;
4
5 public class TaskTest {
6
7     private WebDriver driver;
8     private WebDriverWait wait;
9
10    @BeforeEach
11    public void setup() {
12        System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");
13        driver = new ChromeDriver();
14        wait = new WebDriverWait(driver, Duration.ofSeconds(10));
15    }
16
17    @AfterEach
18    public void teardown() {
19        if (driver != null) {
20            driver.quit();
21        }
22    }
23
24    @Test
25    public void testTaskCreate() {
26        // Open the home page
27        driver.get("http://localhost:4200/home");
28
29        // Click the "let's explore ..." button
30        WebElement exploreButton = wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector(".btn, a[id='explore']")));
31        exploreButton.click();
32
33        // Wait for the login/register page to load
34        wait.until(ExpectedConditions.urlContains("/login"));
35
36        // Perform login
37        WebElement loginButton = wait.until(ExpectedConditions.elementToBeClickable(By.id("loginBtn")));
38        loginButton.click();
39
40        login("register", "password123");
41
42        // Wait for navigation to the boards page
43        wait.until(ExpectedConditions.urlContains("/boards"));
44        assertTrue(driver.getCurrentUrl().contains("/boards"));
45
46        // Navigate to the sprints section
47        WebElement boardButton = wait.until(ExpectedConditions.elementToBeClickable(By.id("boardContent")));
48        boardButton.click();
49        wait.until(ExpectedConditions.urlContains("/tasks/board"));
50        assertTrue(driver.getCurrentUrl().contains("/tasks/board"));
51        WebElement addTaskButton = wait.until(ExpectedConditions.elementToBeClickable(By.id("addTask")));
52        addTaskButton.click();
53        wait.until(ExpectedConditions.urlContains("/addTask"));
54        assertTrue(driver.getCurrentUrl().contains("/addTask"));
55
56        addTask(
57            "Develop login feature", // Task Name
58            "1000", // Status
59            "Implement the login functionality", // Description
60            "8" // Estimation
61        );
62
63        wait.until(ExpectedConditions.urlContains("/tasks/board"));
64        assertTrue(driver.getCurrentUrl().contains("/tasks/board"));
65    }
66
67    private void login(String username, String password) {
68        // Locate fields and fill the form using IDs
69        WebElement usernameField = wait.until(ExpectedConditions.elementToBeClickable(By.id("loginUsername")));
70        WebElement passwordField = driver.findElement(By.id("loginPassword"));
71        WebElement login = driver.findElement(By.id("loginButton"));
72
73        // Clear and enter values.
74        usernameField.clear();
75        usernameField.sendKeys(username);
76        passwordField.clear();
77    }
78
79 }

```

```

91     passwordField.sendKeys(password);
92
93     // Click login
94     login.click();
95 }
96
97 * private void addTask(String taskName, String status, String description, String estimation) {
98     // Locate fields and fill the form using the
99     WebElement nameField = wait.until(ExpectedConditions.elementToBeClickable(By.id("name")));
100     WebElement statusField = driver.findElement(By.id("status"));
101     WebElement descriptionField = driver.findElement(By.id("description"));
102     WebElement estimationField = driver.findElement(By.id("estimation"));
103     WebElement addButton = driver.findElement(By.cssSelector("button[type='submit']"));
104
105     // Clear and enter values
106     nameField.clear();
107     nameField.sendKeys(taskName);
108
109     // Select the status
110     statusField.click();
111     WebElement statusOption = wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//option[@value='" + status + "']")));
112     statusOption.click();
113
114     descriptionField.clear();
115     descriptionField.sendKeys(description);
116
117     estimationField.clear();
118     estimationField.sendKeys(estimation);
119
120     // Select users (represented as chips)
121     try {
122         WebElement userChip = wait.until(ExpectedConditions.elementToBeClickable(
123             By.xpath("/html/body/app-root/html/body/app-addTask/div/div[2]/form/div[2]/div/div[3]/div/div[1]"));
124         userChip.click();
125     } catch (Exception e) {
126         System.out.println("User chip not found.");
127     }
128
129
130     // Select sprints
131     try {
132         WebElement sprintOption = wait.until(ExpectedConditions.elementToBeClickable(
133             By.xpath("/html/body/app-root/html/body/app-addTask/div/div[2]/form/div[2]/div/div[4]/select/option[1]"));
134         sprintOption.click();
135     } catch (Exception e) {
136         System.out.println("Sprint option not found.");
137     }
138
139
140     // Click the Add Task button
141     addButton.click();
142 }
143
144 }
145
146 }
147

```

Figure 2: Voici le code.

Sign up

All fields are required.

Sign Up

Login

Sign up

Enter your username

Enter your email

Enter your password

Username or email already exists. Please try again.

Sign Up

Login

- Email invalide : Le message Please enter a valid email address. a été affiché.

Sign up

reguser

invalid-email

Please enter a valid email address.

Sign Up

Login

- Mot de passe court : Message Password must be at least 6 characters long. .

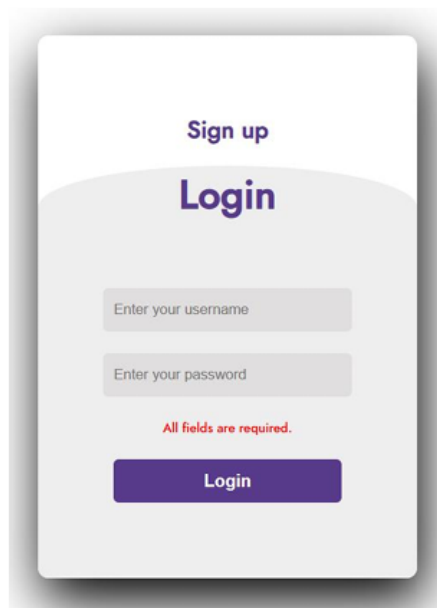
A mobile app interface for a sign-up form. The title "Sign up" is at the top in purple. Below it are three input fields: the first contains "reguser", the second contains "mahdybaakrim@gmail.com", and the third is a password field with three dots. A red error message "Password must be at least 6 characters long." is displayed below the password field. At the bottom, there is a purple "Sign Up" button and a grey "Login" button.

- Succès : L'utilisateur a été redirigé vers la page d'accueil avec une alerte de confirmation.

The same mobile app interface as above, but with a dark grey alert overlay at the top. The alert text reads "localhost:4200 says" and "Your account has been created successfully." with an "OK" button. The form fields below are now empty and have placeholder text: "Enter your username", "Enter your email", and "Enter your password". The "Sign Up" and "Login" buttons remain at the bottom.

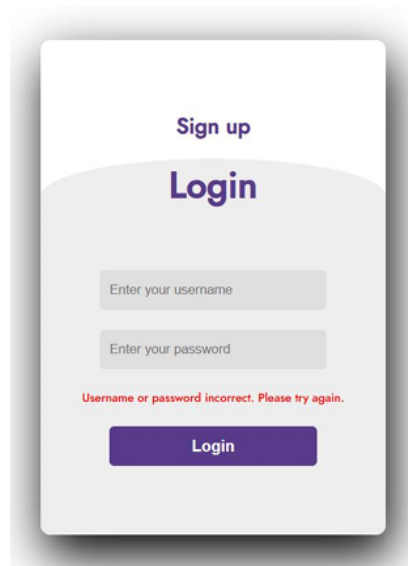
1.3.2 Connexion

- Champs vides : Message "All fields are required." affiché.



A login form with a white background and a light gray shadow. At the top, it says "Sign up" in a small font and "Login" in a larger, bold font. Below the title, there are two input fields: "Enter your username" and "Enter your password". Below these fields, a red error message reads "All fields are required.". At the bottom, there is a purple button labeled "Login".

- Identifiants invalides : Message Username or password incorrect. Please try again. .



A login form with a white background and a light gray shadow. At the top, it says "Sign up" in a small font and "Login" in a larger, bold font. Below the title, there are two input fields: "Enter your username" and "Enter your password". Below these fields, a red error message reads "Username or password incorrect. Please try again.". At the bottom, there is a purple button labeled "Login".

- Succès : Redirection vers la page des boards.

1.3.3 Création de Board

- Le formulaire a été correctement soumis.

- Le board a été ajouté avec succès et visible dans la liste.

1.3.4 Ajout de Sprints

- Les sprints ont été ajoutés avec des informations valides.
- La page a été rechargée et les sprints apparaissent dans la liste.

1.3.5 Ajout de Tâches

- Toutes les tâches ont été ajoutées avec des champs valides.
- Les utilisateurs et les sprints ont été correctement sélectionnés.

1.4 Conclusions

Les tests ont été concluants et confirment que les principales fonctionnalités de l'application fonctionnent comme prévu. Les cas d'erreur sont gérés correctement avec des messages d'erreur pertinents. L'ajout de boards, de sprints et de tâches est fonctionnel et intuitif.

2 Tests JMeter

Les tests JMeter visent à évaluer les performances de l'application sous différentes charges. Neuf groupes de threads représentant différents scénarios API ont été configurés.

2.1 Objectifs

- Mesurer le temps de réponse moyen pour chaque requête.
- Évaluer le débit (requêtes par seconde).
- Identifier les éventuels taux d'erreur sous charge.

2.2 Groupes de Threads

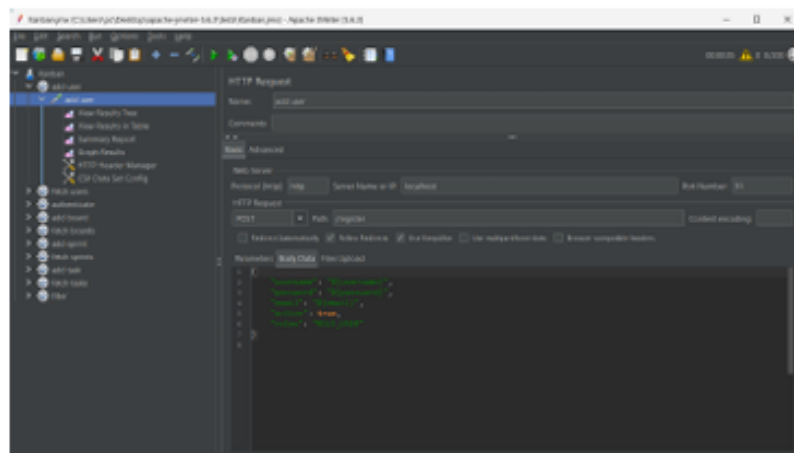
Les groupes de threads incluent les scénarios suivants :

- Ajout d'utilisateurs.
- Récupération d'utilisateurs.

- Authentification.
- Création de boards.
- Récupération de boards.
- Ajout de sprints.
- Récupération de sprints.
- Ajout de tâches.
- Récupération de tâches.

2.3 Résultats :

2.3.1 Add User



Name: View Results in Table

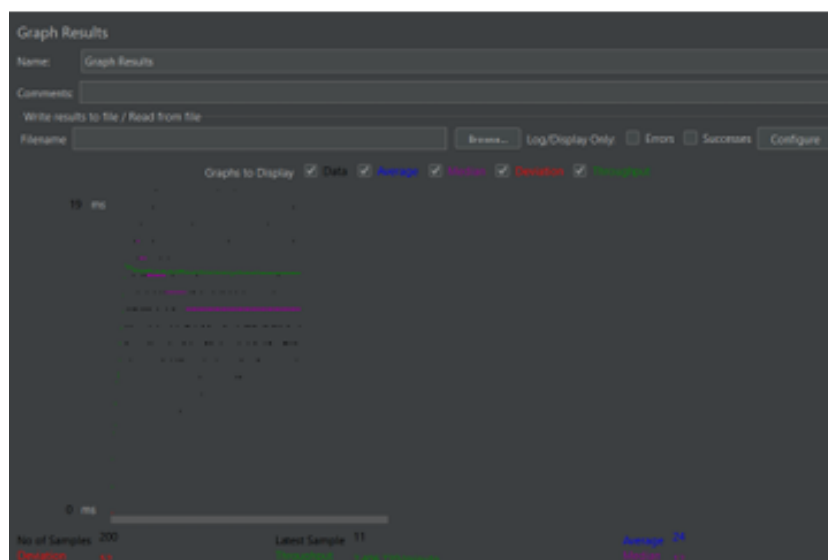
Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	16:08:21.629	add user 1-11	add user	99	✓	496	324	96	0
2	16:08:21.363	add user 1-8	add user	175	✓	496	321	172	0
3	16:08:21.412	add user 1-10	add user	126	✓	496	321	123	0
4	16:08:21.265	add user 1-4	add user	273	✓	496	321	270	0
5	16:08:21.316	add user 1-6	add user	222	✓	496	321	219	0
6	16:08:21.387	add user 1-9	add user	151	✓	496	321	148	0
7	16:08:21.238	add user 1-3	add user	300	✓	496	321	297	0
8	16:08:21.188	add user 1-1	add user	351	✓	496	311	347	1
9	16:08:21.288	add user 1-5	add user	251	✓	496	321	247	1
10	16:08:21.530	add user 1-7	add user	201	✓	496	321	197	0
11	16:08:21.488	add user 1-13	add user	56	✓	496	324	56	1
12	16:08:21.463	add user 1-12	add user	82	✓	496	324	82	0
13	16:08:21.512	add user 1-14	add user	33	✓	496	324	33	1
14	16:08:21.213	add user 1-2	add user	332	✓	496	321	332	1
15	16:08:21.539	add user 1-15	add user	10	✓	496	324	10	0
16	16:08:21.563	add user 1-16	add user	11	✓	496	324	11	1
17	16:08:21.589	add user 1-17	add user	12	✓	496	324	11	1
18	16:08:21.616	add user 1-18	add user	10	✓	496	324	10	1
19	16:08:21.638	add user 1-19	add user	11	✓	496	324	11	0
20	16:08:21.664	add user 1-20	add user	9	✓	496	324	8	0

☐ Scroll automatically? ☐ Child samples? No of Samples: 200 Latest Sample: 11 Average: 24 Deviation: 52



- Temps de réponse moyen : 24 ms
- Débit (Throughput) : 2,406 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.3.2 Fetch Users

HTTP Request

Name:fetch users

Comments:

Basic

Advanced

Web Server

Protocol (Http):httpServer Name or IP:localhostPort Number:91

HTTP Request

GETPath:/AllUsersContent encoding:

☐ Redirect Automatically

☒ Follow Redirects

☒ Use Keep-Alive

☐ Use multipart/form-data

☐ Browser-compatible headers

Parameters

Body Data

Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
-------	-------	-------------	--------------	-----------------

Detail

Add

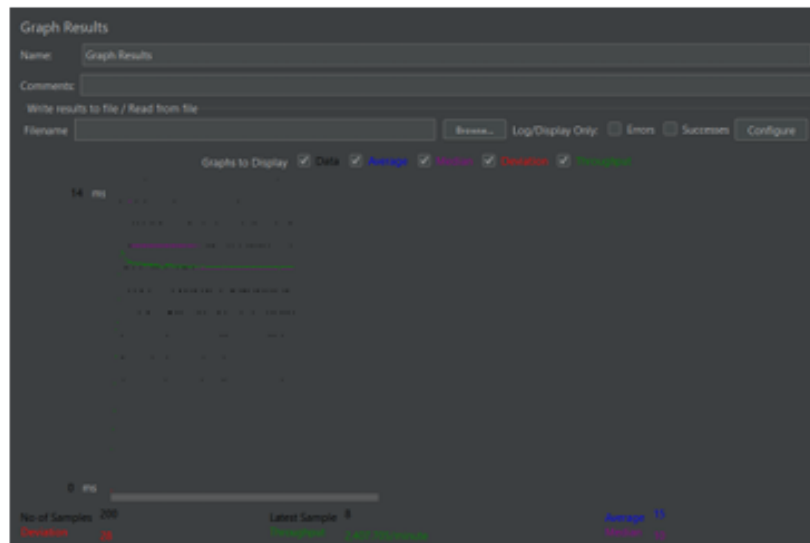
Add from Clipboard

Delete

Up

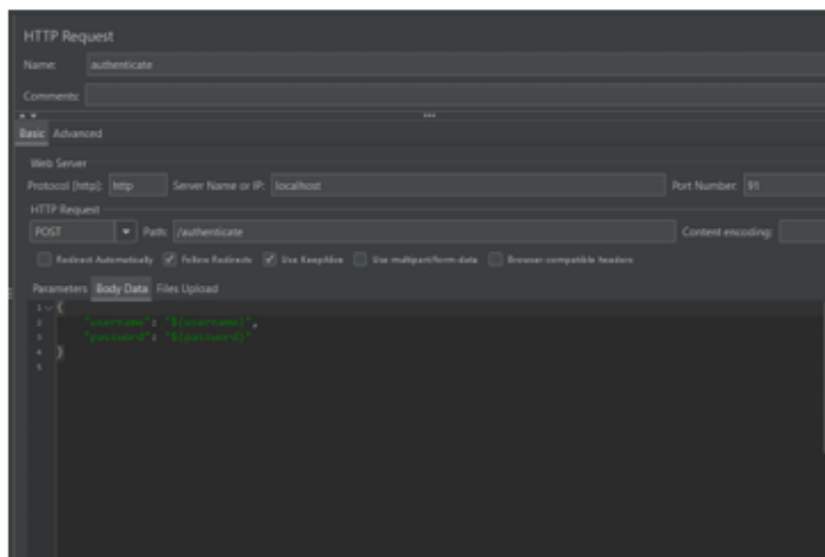
Down

Sample #	Start Time	Thread Name	Label	Sample Time(L)	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	16:52:41.746	fetch users 1 ...	fetch users	131		24677	155	126	1
2	16:52:41.697	fetch users 1 ...	fetch users	180		24677	155	176	0
3	16:52:41.670	fetch users 1 ...	fetch users	207		24677	155	204	1
4	16:52:41.848	fetch users 1 ...	fetch users	29		24677	155	24	1
5	16:52:41.772	fetch users 1 ...	fetch users	105		24677	155	100	0
6	16:52:41.721	fetch users 1 ...	fetch users	156		24677	155	151	1
7	16:52:41.646	fetch users 1 ...	fetch users	232		24677	155	227	1
8	16:52:41.797	fetch users 1 ...	fetch users	81		24677	155	75	1
9	16:52:41.821	fetch users 1 ...	fetch users	57		24677	155	54	1
10	16:52:41.871	fetch users 1 ...	fetch users	13		24677	155	12	0
11	16:52:41.895	fetch users 1 ...	fetch users	7		24677	155	7	0
12	16:52:41.920	fetch users 1 ...	fetch users	6		24677	155	6	1
13	16:52:41.945	fetch users 1 ...	fetch users	6		24677	155	6	1
14	16:52:41.971	fetch users 1 ...	fetch users	5		24677	155	5	0
15	16:52:41.996	fetch users 1 ...	fetch users	10		24677	155	10	0
16	16:52:42.023	fetch users 1 ...	fetch users	15		24677	155	15	0
17	16:52:42.045	fetch users 1 ...	fetch users	10		24677	155	9	1
18	16:52:42.072	fetch users 1 ...	fetch users	10		24677	155	10	1
19	16:52:42.099	fetch users 1 ...	fetch users	11		24677	155	9	1
20	16:52:42.124	fetch users 1 ...	fetch users	9		24677	155	9	1



- Temps de réponse moyen : 15 ms
- Débit (Throughput) : 2,407 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.3.3 Authenticate



Sample #	Start Time	Thread Name	Label	Sample Time(L)	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	161527.236	authenticate ..	authenticate	14	🟢	611	241	14	1
2	161527.210	authenticate ..	authenticate	41	🟢	609	243	40	1
3	161527.262	authenticate ..	authenticate	2	🟢	611	241	2	0
4	161527.287	authenticate ..	authenticate	5	🟢	611	241	5	0
5	161527.313	authenticate ..	authenticate	7	🟢	611	241	7	1
6	161527.336	authenticate ..	authenticate	5	🟢	611	241	5	1
7	161527.362	authenticate ..	authenticate	6	🟢	610	241	6	1
8	161527.388	authenticate ..	authenticate	6	🟢	611	241	6	1
9	161527.413	authenticate ..	authenticate	4	🟢	611	241	4	0
10	161527.438	authenticate ..	authenticate	5	🟢	610	241	5	1
11	161527.461	authenticate ..	authenticate	6	🟢	614	243	5	1
12	161527.485	authenticate ..	authenticate	3	🟢	614	243	3	0
13	161527.510	authenticate ..	authenticate	6	🟢	614	243	6	1
14	161527.536	authenticate ..	authenticate	4	🟢	614	243	4	0
15	161527.560	authenticate ..	authenticate	4	🟢	614	243	4	0
16	161527.587	authenticate ..	authenticate	5	🟢	614	243	5	0
17	161527.611	authenticate ..	authenticate	9	🟢	614	243	9	0
18	161527.639	authenticate ..	authenticate	4	🟢	614	243	4	0
19	161527.662	authenticate ..	authenticate	6	🟢	614	243	6	1
20	161527.686	authenticate ..	authenticate	5	🟢	614	243	5	1

☐ Scroll automatically?
 ☐ Cuid samples?
 No of Samples: 200
 Latest Sample: 3
 Average: 5
 Deviation: 2



- Temps de réponse moyen : 5 ms
- Débit (Throughput) : 2,411 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.3.4 Add Board

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

Protocol (http): Server Name or IP: Port Number:

HTTP Request

Method: Path: Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters **Body Data** Files Upload

```

1 {
2   "name": "TestBoard",
3   "description": "TestBoard description",
4   "score": 200
5 }
6

```

Sample #	Start Time	Thread Name	Label	Sample Time(s)	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	16/2/10.326	add board 1...	add board	39		656	253	39	1
2	16/2/10.339	add board 1...	add board	8		664	261	8	0
3	16/2/10.334	add board 1...	add board	33		664	261	33	1
4	16/2/10.384	add board 1...	add board	10		664	261	10	0
5	16/2/10.409	add board 1...	add board	10		664	261	9	0
6	16/2/10.433	add board 1...	add board	10		664	261	9	1
7	16/2/10.439	add board 1...	add board	9		664	261	8	0
8	16/2/10.483	add board 1...	add board	9		664	261	9	1
9	16/2/10.509	add board 1...	add board	7		665	261	7	1
10	16/2/10.523	add board 1...	add board	13		665	261	13	0
11	16/2/10.560	add board 1...	add board	10		667	263	10	0
12	16/2/10.584	add board 1...	add board	12		667	263	12	1
13	16/2/10.613	add board 1...	add board	12		667	263	12	1
14	16/2/10.634	add board 1...	add board	11		667	263	11	1
15	16/2/10.660	add board 1...	add board	12		667	263	12	2
16	16/2/10.684	add board 1...	add board	11		667	263	11	0
17	16/2/10.711	add board 1...	add board	13		667	263	13	1
18	16/2/10.735	add board 1...	add board	10		667	263	10	1
19	16/2/10.760	add board 1...	add board	11		667	263	11	1
20	16/2/10.785	add board 1...	add board	15		667	263	15	1

☐ Scroll automatically? ☐ Child samples? No of Samples: 200 Latest Sample: 12 Average: 11 Deviation: 6





- Temps de réponse moyen : 11 ms
- Débit (Throughput) : 2,403 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.3.5 Fetch Boards

HTTP Request

Name:

Comments:

Basic **Advanced**

Web Server

Protocol (http) Server Name or IP: Port Number:

HTTP Request

Method: Path: Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use Keepalive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Uploaded

Send Parameters With the Request:

Name	Value	URL Encode?	Content-Type	Include Equals?
------	-------	-------------	--------------	-----------------

Cancel Add Add from Clipboard Delete Up Down

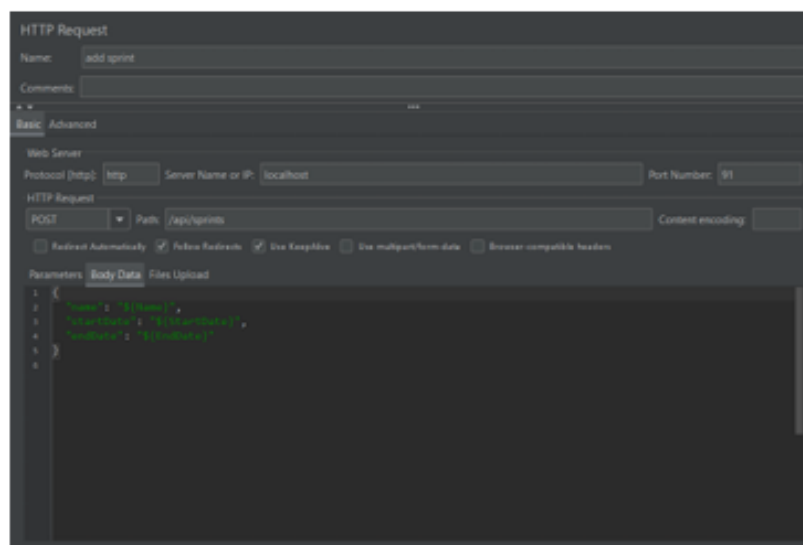
Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Send Bytes	Latency	Connect Tim...
1	16:22:29.414	fetch boards ..	fetch boards	36		43180	157	35	1
2	16:22:29.450	fetch boards ..	fetch boards	19		43180	157	18	1
3	16:22:29.466	fetch boards ..	fetch boards	12		43180	157	11	1
4	16:22:29.496	fetch boards ..	fetch boards	13		43180	157	12	1
5	16:22:29.526	fetch boards ..	fetch boards	9		43180	157	9	1
6	16:22:29.542	fetch boards ..	fetch boards	9		43180	157	9	0
7	16:22:29.573	fetch boards ..	fetch boards	11		43180	157	10	1
8	16:22:29.604	fetch boards ..	fetch boards	12		43180	157	11	1
9	16:22:29.619	fetch boards ..	fetch boards	9		43180	157	8	0
10	16:22:29.650	fetch boards ..	fetch boards	11		43180	157	10	1
11	16:22:29.681	fetch boards ..	fetch boards	10		43180	157	9	1
12	16:22:29.695	fetch boards ..	fetch boards	10		43180	157	10	1
13	16:22:29.726	fetch boards ..	fetch boards	10		43180	157	9	1
14	16:22:29.757	fetch boards ..	fetch boards	11		43180	157	11	1
15	16:22:29.772	fetch boards ..	fetch boards	10		43180	157	9	0
16	16:22:29.803	fetch boards ..	fetch boards	9		43180	157	8	1
17	16:22:29.834	fetch boards ..	fetch boards	11		43180	157	10	1
18	16:22:29.849	fetch boards ..	fetch boards	9		43180	157	8	0
19	16:22:29.881	fetch boards ..	fetch boards	11		43180	157	10	1
20	16:22:29.897	fetch boards ..	fetch boards	9		43180	157	9	0

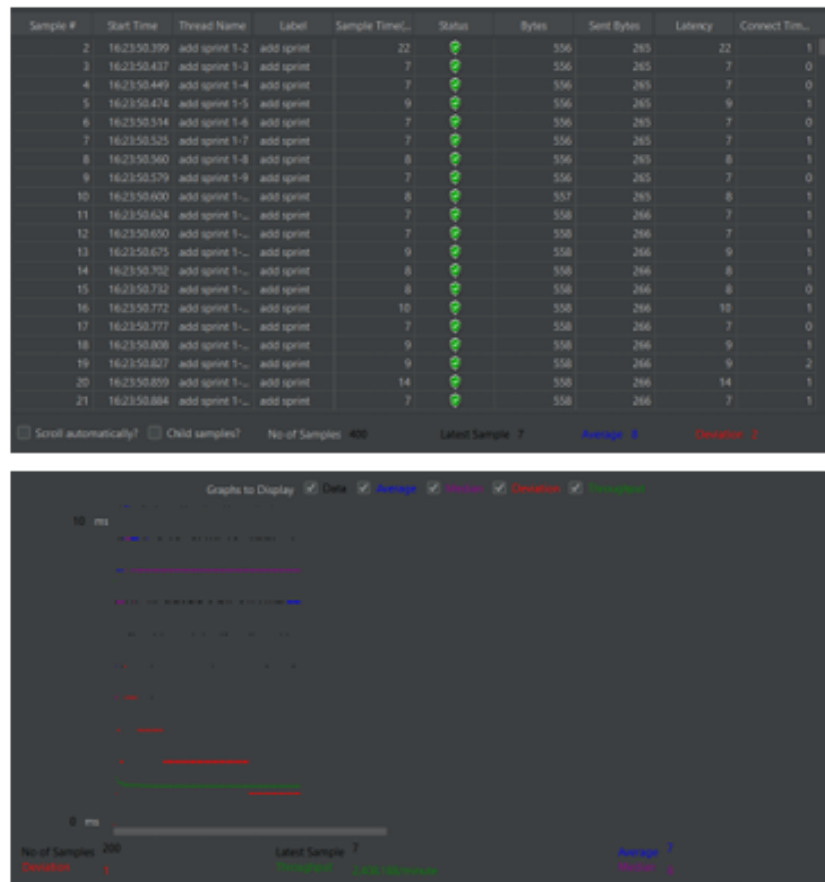
☐ Scroll automatically? ☐ Hide samples? No. of Samples: 200 Latest Sample: 8 Average: 10 Deviation: 1



- Temps de réponse moyen : 10 ms
- Débit (Throughput) : 2,405 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.3.6 Add Sprint





- Temps de réponse moyen : 8 ms
- Débit (Throughput) : 2,408 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.3.7 Fetch Sprints

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

Protocol (http): Server Name or IP: Port Number:

HTTP Request

GET Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use Keepalive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request

Name	Value	URL Encode?	Content Type	Include Equals?
------	-------	-------------	--------------	-----------------

Cancel Add Add from Clipboard Delete Up Down

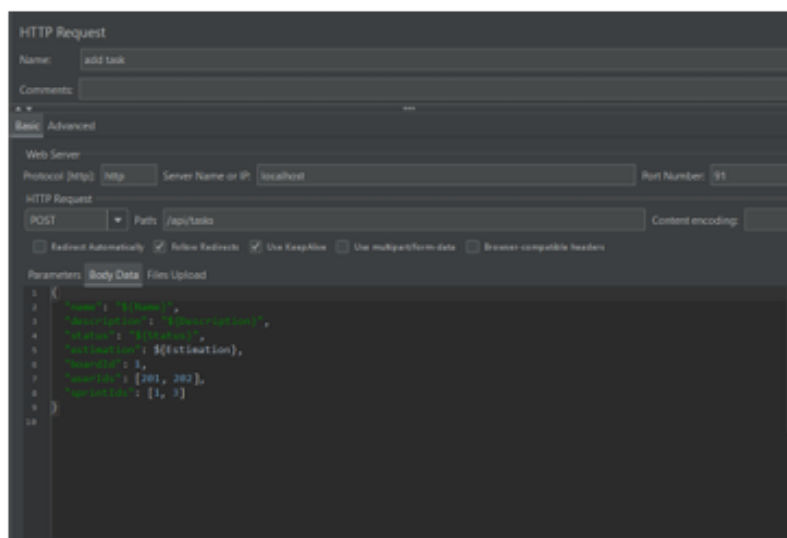
Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	16:26:03.695	fetch-sprints...	fetch-sprints	36		46047	158	34	2
2	16:26:03.724	fetch-sprints...	fetch-sprints	17		46047	158	15	1
3	16:26:03.755	fetch-sprints...	fetch-sprints	16		46047	158	14	1
4	16:26:03.786	fetch-sprints...	fetch-sprints	15		46047	158	14	1
5	16:26:03.802	fetch-sprints...	fetch-sprints	14		46047	158	13	0
6	16:26:03.833	fetch-sprints...	fetch-sprints	13		46047	158	11	0
7	16:26:03.849	fetch-sprints...	fetch-sprints	9		46047	158	8	0
8	16:26:03.879	fetch-sprints...	fetch-sprints	9		46047	158	8	1
9	16:26:03.911	fetch-sprints...	fetch-sprints	12		46047	158	11	0
10	16:26:03.927	fetch-sprints...	fetch-sprints	9		46047	158	8	0
11	16:26:03.958	fetch-sprints...	fetch-sprints	14		46047	158	12	1
12	16:26:03.974	fetch-sprints...	fetch-sprints	12		46047	158	11	0
13	16:26:04.005	fetch-sprints...	fetch-sprints	15		46047	158	14	0
14	16:26:04.035	fetch-sprints...	fetch-sprints	12		46047	158	11	0
15	16:26:04.050	fetch-sprints...	fetch-sprints	12		46047	158	10	1
16	16:26:04.081	fetch-sprints...	fetch-sprints	14		46047	158	12	1
17	16:26:04.097	fetch-sprints...	fetch-sprints	11		46047	158	10	0
18	16:26:04.128	fetch-sprints...	fetch-sprints	14		46047	158	13	1
19	16:26:04.159	fetch-sprints...	fetch-sprints	16		46047	158	14	1
20	16:26:04.174	fetch-sprints...	fetch-sprints	15		46047	158	13	2

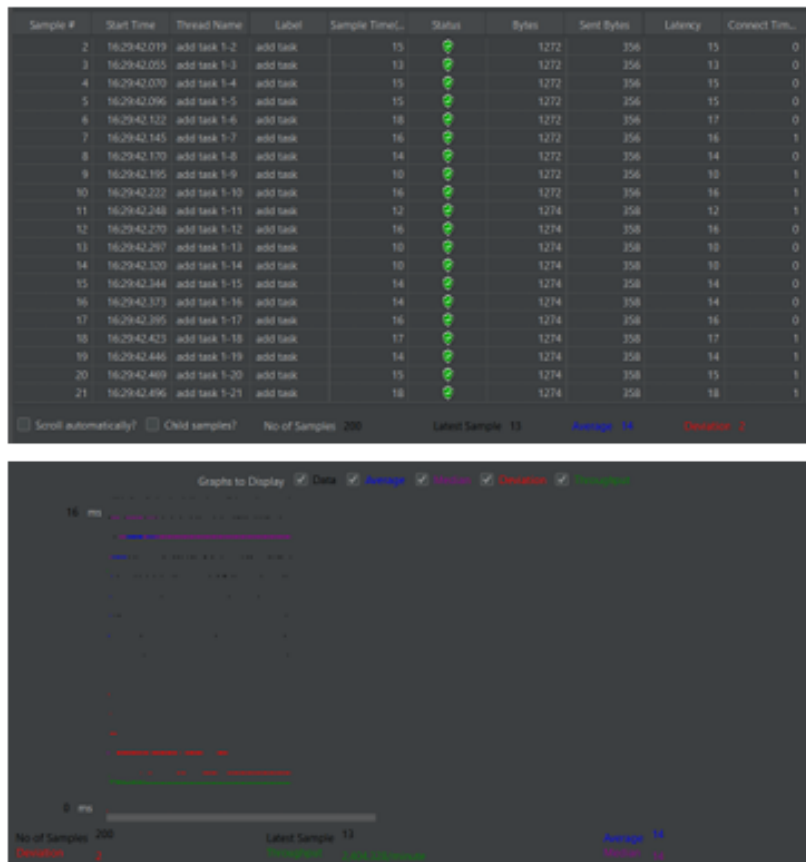
☐ Scroll automatically? ☐ Child samples? No of Samples: 200 Latest Sample: 12 Average: 12 Deviation: 2



- Temps de réponse moyen : 12 ms
- Débit (Throughput) : 2,402 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.3.8 Add Task





- Temps de réponse moyen : 14 ms
- Débit (Throughput) : 2,404 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.3.9 Fetch Tasks

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

Protocol (Default): Server Name or IP: Port Number:

HTTP Request

Method: Path: Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name	Value	URL Encode?	Content-Type	Include Equals?
------	-------	-------------	--------------	-----------------

Default Add Add from Clipboard Delete Up Down

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Tim...
1	16:31:07.520	fetch tasks 1-1	fetch tasks	118		331985	156	47	0
2	16:31:07.547	fetch tasks 1-2	fetch tasks	93		331985	156	20	1
3	16:31:07.572	fetch tasks 1-3	fetch tasks	76		331985	156	12	1
4	16:31:07.596	fetch tasks 1-4	fetch tasks	72		331985	156	15	0
5	16:31:07.626	fetch tasks 1-5	fetch tasks	62		331985	156	14	0
6	16:31:07.645	fetch tasks 1-6	fetch tasks	55		331985	156	8	0
7	16:31:07.671	fetch tasks 1-7	fetch tasks	59		331985	156	8	1
8	16:31:07.696	fetch tasks 1-8	fetch tasks	72		331985	156	10	0
9	16:31:07.721	fetch tasks 1-9	fetch tasks	72		331985	156	16	0
10	16:31:07.752	fetch tasks 1-...	fetch tasks	58		331985	156	12	1
11	16:31:07.770	fetch tasks 1-...	fetch tasks	53		331985	156	8	0
12	16:31:07.798	fetch tasks 1-...	fetch tasks	54		331985	156	8	0
13	16:31:07.823	fetch tasks 1-...	fetch tasks	53		331985	156	8	0
14	16:31:07.846	fetch tasks 1-...	fetch tasks	54		331985	156	9	1
15	16:31:07.871	fetch tasks 1-...	fetch tasks	54		331985	156	8	1
16	16:31:07.896	fetch tasks 1-...	fetch tasks	53		331985	156	8	1
17	16:31:07.922	fetch tasks 1-...	fetch tasks	53		331985	156	8	0
18	16:31:07.946	fetch tasks 1-...	fetch tasks	50		331985	156	8	0
19	16:31:07.972	fetch tasks 1-...	fetch tasks	54		331985	156	8	0
20	16:31:07.995	fetch tasks 1-...	fetch tasks	56		331985	156	8	1

☐ Scroll automatically? ☐ Child samples? No of Samples: 200 Latent Sample: 52 Average: 75 Deviation: 15



- Temps de réponse moyen : 59 ms
- Débit (Throughput) : 2,380 requêtes/min
- Taux d'erreur : 0%
- Succès vs Échecs : 200 succès, 0 échec

2.4 Conclusion

Les tests combinés WebDriver et JMeter montrent que l'application Kanban est fonctionnelle et performante. Toutes les fonctionnalités clés ont été validées, et les performances restent optimales même sous une charge utilisateur élevée.