Documentazione Sprint 2

Sprint 2 Nebulawatches

Tom Schillerwein, Alexandru Ciobanu, Gioele Chiodoni

2024

Sommario

[Implementazione 2](#_Toc159267295)

[Login 2](#_Toc159267296)

[Login JWT 2](#_Toc159267297)

[Login Google 2](#_Toc159267298)

[Test 2](#_Toc159267299)

[Test Case 2](#_Toc159267300)

[Risultati test 7](#_Toc159267301)

# Implementazione

## Login

Nel nostro applicativo il login si può fare con due modalità. La prima tipologia di login è basata su JWT (JSON web token), la seconda è un login con Google.

### Login JWT

Per registrare un nuovo utente, l’applicazione necessita di un email, username e password. Se la richiesta è valida, l’applicativo ritorna un token che serve a dire al server che questo utente è autenticato.

Per autenticare un utente già esistente, l’applicazione necessita di email e password dell’utente. Anche in questo caso, se la richiesta è valida, ritorna un token che segnala al server che questo utente è autenticato.

L’utente invia il JWT ricevuto in ogni richiesta successiva e l’applicazione verifica il JWT per autenticare l’utente.

Se il token sta per scadere, l’applicazione può decidere di rinnovarlo e inviare un nuovo JWT all’utente.

### Login Google

Quando un utente vuole accedere, l’applicazione lo reindirizza alla pagina di login di Google. L’utente inserisce poi le proprie credenziali di Google e viene autenticato se questo è andato a buon fine.

# Test

## Test Case

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Registrazione utente |
| **Descrizione:** | Passando username, email, password ci si può registrare. | | |
| **Prerequisiti:** | Accedere a | | |
| **Procedura:** | 1. Accendere Postman 2. Creare collezione “NebulaWatches” 3. Creare richiesta POST e mettere come URL “localhost:64321/api/v1/auth/register” poi andare nella tab “Body/raw”: {  “username” : “gino”,  “email”: “gino@gmail.com”,  “password”: “1234” } 4. Schiacciare Send | | |
| **Risultati attesi:** | Nella response un token di autorizzazione | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-002 | **Nome:** | Login utente |
| **Descrizione:** | Passando email, password, viene autenticato l’utente e viene generato un token di autenticazione | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL avere un utente registrato. | | |
| **Procedura:** | 1. Accendere Postman 2. Creare richiesta POST e mettere come URL “localhost:64321/api/v1/auth/authenticate” poi andare nella tab “Body/raw”: {  “email”: “gino@gmail.com”,  “password”: “1234” } 3. Schiacciare Send | | |
| **Risultati attesi:** | Nella response un token di autorizzazione | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-007 | **Nome:** | Endpoint watch/reference |
| **Descrizione:** | Ritorna le informazioni sull’orologio | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/watches/101.035”, questo andrà a cercare l’orologio con reference a “101.035” 3. Aggiungere il token di autenticazione in Authorization/Bearer Token 4. Schiacciare Send. | | |
| **Risultati attesi:** | Nella response le informazioni sull’orologio | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-007 | **Nome:** | Endpoint watch/reference/image |
| **Descrizione:** | Ritorna l’array di byte dell’immagine. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman. 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/watches/101.035/image”, questo andrà a cercare l’orologio con reference a “101.035” e ne prenderà l’immagine. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare Send. | | |
| **Risultati attesi:** | Nella response l’array di byte dell’immagine | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-007 | **Nome:** | Endpoint watches |
| **Descrizione:** | Ritorna le informazioni su tutti gli orologi nel database. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/watches”, questo andrà a prendere tutti gli orologi. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare Send. | | |
| **Risultati attesi:** | Nella response le informazioni su tutti gli orologi nel database. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-007 | **Nome:** | Endpoint hands |
| **Descrizione:** | Ritorna le informazioni su tutti i hands nel database. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman. 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/hands”, questo andrà a prendere tutti i hands. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare Send. | | |
| **Risultati attesi:** | Nella response le informazioni su tutti i hands nel database. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-007 | **Nome:** | Endpoint dial finishes |
| **Descrizione:** | Ritorna le informazioni su tutti i dial finishes nel database. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman. 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/dial\_finishes”, questo andrà a prendere tutti i dial finishes. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare Send. | | |
| **Risultati attesi:** | Nella response le informazioni su tutti i dial finishes nel database | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-007 | **Nome:** | Endpoint dial colors |
| **Descrizione:** | Ritorna le informazioni su tutti i dial colors nel database. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman. 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/dial\_colors”, questo andrà a prendere tutti i dial colors. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare Send. | | |
| **Risultati attesi:** | Nella response le informazioni su tutti i dial colors nel database | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-009  REQ-007 | **Nome:** | Endpoint watch shape |
| **Descrizione:** | Ritorna le informazioni su tutti i watch shape nel database. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman. 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/watch\_shapes”, questo andrà a prendere tutti i watch shape. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare Send. | | |
| **Risultati attesi:** | Nella response le informazioni su tutti i watch shape nel database | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-010  REQ-007 | **Nome:** | Endpoint materials |
| **Descrizione:** | Ritorna le informazioni su tutti i materials nel database. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman. 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/materials”, questo andrà a prendere tutti i materials. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare materials. | | |
| **Risultati attesi:** | Nella response le informazioni su tutti i materials nel database | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-011  REQ-007 | **Nome:** | Endpoint brands |
| **Descrizione:** | Ritorna le informazioni su tutti i brands nel database. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman. 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/brands”, questo andrà a prendere tutti i materials. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare brands. | | |
| **Risultati attesi:** | Nella response le informazioni su tutti i brands nel database | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-012  REQ-007 | **Nome:** | Endpoint families |
| **Descrizione:** | Ritorna le informazioni su tutti le families nel database. | | |
| **Prerequisiti:** | Installare Postman, accendere server Spring, accendere Server MySQL aver eseguito il login con un utente. | | |
| **Procedura:** | 1. Accendere Postman. 2. Creare richiesta GET e mettere come URL “localhost:64321/v1/families”, questo andrà a prendere tutti le families. 3. Aggiungere il token di autenticazione in Authorization/Bearer Token. 4. Schiacciare families. | | |
| **Risultati attesi:** | Nella response le informazioni su tutti le families nel database | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-013  REQ-002 | **Nome:** | Google login |
| **Descrizione:** | L’utente viene mandato al login di Google per accedere | | |
| **Prerequisiti:** | Server Spring e Server MySQL accesi | | |
| **Procedura:** | 1. Accedere al dominio http://localhost:64321/hello 2. Inserire le proprie credenziali dopo essere stati ridiretti sul login di Google | | |
| **Risultati attesi:** | Il login su Google va a buon fine | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-014  REQ-002 | **Nome:** | Google returns email |
| **Descrizione:** | Dopo il login con Google viene stampato a schermo la email con cui è stato eseguito l’accesso. | | |
| **Prerequisiti:** | Login con Google andato a buon fine. | | |
| **Procedura:** | 1. Fare login con Google, vedi TC-13 | | |
| **Risultati attesi:** | Email stampata a schermo. | | |

## Risultati test

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Risultato | Data | Stato |
| TC-001 | Il token di autorizzazione è stato creato e passato. | 02.02.2024  Sprint 1 | Passato |
| TC-002 | Il token di autorizzazione è stato creato e passato. | 02.02.2024  Sprint 1 | Passato |
| TC-003 | Le informazioni sugli orologi sono state passate. | 02.02.2024  Sprint 1 | Passato |
| TC-004 | L’array di byte dell’immagine è stato ritornato. | 02.02.2024  Sprint 1 | Passato |
| TC-005 | Tutte le informazioni degli orologi nel database sono state ritornate. | 02.02.2024  Sprint 1 | Passato |
| TC-006 | Tutte le informazioni sulle hands nel database sono state ritornate. | 02.02.2024  Sprint 1 | Passato |
| TC-007 | Tutte le informazioni sui dial finishes sono state ritornate. | 02.02.2024  Sprint 1 | Passato |
| TC-008 | Tutte le informazioni sui dial colors sono state ritornate. | 02.02.2024  Sprint 1 | Passato |
| TC-009 | Tutte le informazioni sui watch shapes sono state ritornate. | 02.02.2024  Sprint 1 | Passato |
| TC-010 | Tutte le informazioni sui materials sono state ritornate. | 02.02.2024  Sprint 1 | Passato |
| TC-011 | Tutte le informazioni sui brands sono state ritornate. | 02.02.2024  Sprint 1 | Passato |
| TC-012 | Tutte le informazioni sulle famiglie sono state ritornate | 02.02.2024  Sprint 1 | Passato |
| TC-013 | Non viene aperto il login, ma viene ritornata utente anonimo: | 02.02.2024  Sprint 1 | Fallito |
| TC-014 | Non viene aperto il login, ma viene ritornata utente anonimo: | 02.02.2024  Sprint 1 | Fallito |