



Unit Test Report

HardShipV2

Versione 1.1



Sommario

Introduzione	3
Details	3
Overview dei risultati dei test	3
Dettagli dei risultati degli unit test HardShip (1 versione)	3
GestioneFeedbackControllerTest	3
GestioneBranoControllerTest	3
GestioneNewsControllerTest	3
GestioneAlbumControllerTest	4
GestioneAlbumServiceTest	4
GestioneNewsServiceTest	4
GestioneFeedbackServiceTest	4
Coverage dei test eseguiti	4
Dettagli dei risultati degli unit test HardShip (2 versione)	5
GestioneFeedbackControllerTest	5
GestioneBranoControllerTest	5
GestioneNewsControllerTest	5
GestioneAlbumControllerTest	5
GestioneAlbumServiceTest	5
GestioneNewsServiceTest	6
GestioneFeedbackServiceTest	6
GestioneUtenteServiceTest	6
GestioneUtenteControllerTest	6
Coverage dei test eseguiti	6



1. Introduzione

Il testing di unità rappresenta la fase di testing in cui si assicura che le componenti sviluppate funzionino in isolamento. Questo documento ha il compito di identificare la strategia di testing di unità per il sistema HardShipV2. In particolare, saranno specificate le componenti da testare e il modo in cui il testing dovrà essere eseguito. Lo scopo del documento è quello di riportare l'esecuzione del testing unitario e di verificare la presenza o meno di fault.

2. Details

2.1. Overview dei risultati dei test

Non sono stati riscontrati fault nelle componenti del sistema HardShipV2. Per informazioni dettagliate sugli item testati e sui risultati si faccia riferimento al paragrafo successivo.

2.2. Dettagli dei risultati degli unit test HardShip (1 versione)

2.2.1. GestioneFeedbackControllerTest

✓	✓	GestioneFeedbackControllerTests	473 ms
	✓	getAllFeedback_success()	323 ms
	✓	createNews_success()	113 ms
	✓	getFeedbackWithClientID()	37 ms

2.2.2. GestioneBranoControllerTest

✓	GestioneBranoControllerTests	16 ms
	getBraniWithAlbumId()	16 ms

2.2.3. GestioneNewsControllerTest

✓	GestioneNewsControllerTests	76 ms
	createNews_success()	33 ms
	getAllNews_success()	14 ms
	getNewsById_success()	29 ms



2.2.4. GestioneAlbumControllerTest

✓ GestioneAlbumControllerTests	159 ms
✓ uploadAlbum_success()	25 ms
✓ getAlbums_success()	25 ms
✓ getDetailAlbum_success()	39 ms
✓ getAllAlbumsOrderByEtichetta()	21 ms
✓ createRequestAlbum_success()	16 ms
✓ getAllAlbumsOrderByArtista()	33 ms

2.2.5. GestioneAlbumServiceTest

✓ GestioneAlbumServiceTest	127 ms
✓ givenIdThenShouldReturnAlbumOfThatId()	124 ms
✓ getAllAlbumsTest()	3 ms

2.2.6. GestioneNewsServiceTest

✓ GestioneNewsServiceTest	199 ms
✓ givenGetAllNewsShouldReturnListOfAllNews()	98 ms
✓ givenIdThenShouldReturnNewsOfThatId()	101 ms
✓ givenNewsToAddShouldReturnAddedNews()	

2.2.7. GestioneFeedbackServiceTest

✓ GestioneFeedbackServiceTest	23 ms
✓ givenFeedbackToAddShouldReturnAddedFeedback()	23 ms
✓ givenGetAllNewsShouldReturnListOfAllNews()	

2.3. Coverage dei test eseguiti

71% classes, 51% lines covered in package 'io.hardship.hardshipapi'			
Element	Class, %	Method, %	Line, %
controller	100% (6/6)	93% (14/15)	75% (43/57)
dao	100% (0/0)	100% (0/0)	100% (0/0)
entity	58% (14/24)	47% (91/193)	42% (96/228)
service	83% (5/6)	53% (8/15)	60% (25/41)
utils	100% (2/2)	100% (5/5)	100% (9/9)
HardshipApiApplication	100% (1/1)	0% (0/1)	50% (1/2)



2.4. Dettagli dei risultati degli unit test HardShip (2 versione)

2.4.1. GestioneFeedbackControllerTest

✓	✓	GestioneFeedbackControllerTests	473 ms
	✓	getAlIFeedback_success()	323 ms
	✓	createNews_success()	113 ms
	✓	getFeedbackWithClientID()	37 ms

2.4.2. GestioneBranoControllerTest

✓	GestioneBranoControllerTests	16 ms
	getBraniWithAlbumId()	16 ms

2.4.3. GestioneNewsControllerTest

✓	GestioneNewsControllerTests	76 ms
	createNews_success()	33 ms
	getAlINews_success()	14 ms
	getNewsById_success()	29 ms

2.4.4. GestioneAlbumControllerTest

✓	GestioneAlbumControllerTests	159 ms
	uploadAlbum_success()	25 ms
	getAlbums_success()	25 ms
	getDetailAlbum_success()	39 ms
	getAlIAlbumsOrderByEtichetta()	21 ms
	createRequestAlbum_success()	16 ms
	getAlIAlbumsOrderByArtista()	33 ms

2.4.5. GestioneAlbumServiceTest

✓	GestioneAlbumServiceTest	127 ms
	givenIdThenShouldReturnAlbumOfThatId()	124 ms
	getAlIAlbumsTest()	3 ms



2.4.6. GestioneNewsServiceTest

✓ GestioneNewsServiceTest	199 ms
✓ givenGetAllNewsShouldReturnListOfAllNews()	98 ms
✓ givenIdThenShouldReturnNewsOfThatId()	101 ms
✓ givenNewsToAddShouldReturnAddedNews()	

2.4.7. GestioneFeedbackServiceTest

✓ GestioneFeedbackServiceTest	23 ms
✓ givenFeedbackToAddShouldReturnAddedFeedback()	23 ms
✓ givenGetAllNewsShouldReturnListOfAllNews()	

2.4.8. GestioneUtenteServiceTest

✓ GestioneUtenteServiceTest	24 ms
✓ registerClientTest()	23 ms
✓ loginCliente()	1 ms

2.4.9. GestioneUtenteControllerTest

✓ GestioneUtenteControllerTests	32 ms
✓ registraCliente_success()	32 ms

2.5. Coverage dei test eseguiti

73% classes, 48% lines covered in package 'io.hardship.hardshipapi'			
Element	Class, %	Method, %	Line, %
controller	100% (7/7)	82% (14/17)	61% (44/71)
dao	100% (0/0)	100% (0/0)	100% (0/0)
entity	58% (14/24)	47% (92/195)	41% (97/231)
service	85% (6/7)	50% (9/18)	51% (29/56)
utils	100% (2/2)	100% (5/5)	100% (9/9)
HardshipApiApplication	100% (1/1)	0% (0/1)	50% (1/2)