



# Università degli Studi di Salerno

Corso di Ingegneria del Software  
Classe 1 Resto 0  
Corso di Laurea in Informatica  
A.A. 2022/23

## Quiad Test Execution Report

Versione 2.3  
29/12/2022



**QUIAD**  
Family Tree

**Partecipanti al progetto e scriventi**

Nome	Matricola
Di Pasquale Valerio	0512110638
Troisi Vito	0512109807

---

**Revision History**

Data	Versione	Descrizione	Autore
23/12/2022	1.0	Prima stesura Test Execution Report e Indice	T.V.
27/12/2022	2.0	Testing di unità lato server	D.P.V. T.V.
27/12/2022	2.1	Testing REST API: prima parte	T.V.
28/12/2022	2.2	Testing REST API: seconda parte e inserimento report	D.P.V. T.V.
29/12/2022	2.3	Spostati test di unità in documento apposito	T.V.

**Indice**

1. Overview.....	p.4
2. Riferimenti.....	p.4
3. Report del testing di unità server-side.....	p.5
4. Report del testing della REST API.....	p.7
5. Report del testing di unità client-side.....	p.
6. Report del testing di integrazione.....	p.
7. Report del testing di sistema.....	p.

## Overview

Il presente documento riporta i risultati dell'esecuzione dei test effettuati, ed è comprensivo di ogni attività svolta, dal testing di unità a quello di sistema.

Si procederà innanzitutto analizzando i risultati del testing della REST API il che, con riferimento allo Object Design Document, è effettuabile inviando richieste opportune agli endpoint HTTP. Saranno poi osservati i risultati dell'esecuzione dei test di unità, integrazione e sistema. Si precisa che NON saranno riportati gli output di metodi elementari (i.e. getters/setters).

## Riferimenti

Per procedere al meglio nella lettura del presente documento, si rimanda a documenti ulteriori legati ai requisiti del sistema Quiad, alla sua progettazione, ed al piano di testing redatto a monte di tale attività. Nello specifico, si segnalano:

- Il Requirements Analysis Document (RAD)
- Il System Design Document (SDD)
- Lo Object Design Document (ODD)
- Il Test Plan
- Unit Test Cases Specification (UnitTCS)

Il Test Plan, il quale contiene, per ogni funzionalità da testare, i casi di test esplicitati mediante tecniche quali *Category Partition*, è di particolare rilievo nel contesto del Test Execution Report presente, ed esplicita inoltre le modalità e gli strumenti impiegati nell'esecuzione stessa dei test. Un analogo discorso è da replicare per il documento Unit Test Cases Specification, per quanto concerne i test di unità.

**Report del testing di unità server-side****PASS** dist/account/controllers/account.controller.test.js**Account Controller**

- ✓ Should retrieve an existing username (1 ms)
- ✓ Should not retrieve a non existing username (1 ms)
- ✓ Should create an account (2 ms)

**PASS** dist/account/services/auth.service.test.js**Authorization Service**

- ✓ Should authenticate a valid account (68 ms)
- ✓ Should not authenticate an account with non valid password (66 ms)
- ✓ Should not authenticate an account with non valid username (2 ms)
- ✓ Should not authenticate an account with empty username (1 ms)
- ✓ Should not authenticate an account with empty password (2 ms)
- ✓ Should not authenticate an account with empty password (2 ms)

**PASS** dist/account/services/registration.service.test.js**Registration Service**

- ✓ Should register an account (70 ms)
- ✓ Should not register an account (65 ms)

**PASS** dist/tree/controllers/node.controller.test.js**Node Controller**

- ✓ Should find a list of nodes filtered by owner (4 ms)
- ✓ Should create a node (3 ms)
- ✓ Should update a node (4 ms)
- ✓ Should delete a node (2 ms)
- ✓ Should bind a document to a node (3 ms)
- ✓ Should unbind a document from a node (7 ms)

**PASS** dist/tree/services/tree.service.test.js**Tree Service**

- ✓ Should find a list of nodes filtered by owner (6 ms)
- ✓ Should create a node (2 ms)
- ✓ Should update a node (4 ms)
- ✓ Should handle any unknown error during creation (3 ms)
- ✓ Should delete a node (3 ms)
- ✓ Should bind a document to a node (3 ms)
- ✓ Should unbind a document from a node (4 ms)

**PASS** dist/document/controllers/document.controller.test.js**Document Controller**

- ✓ Should find a list of existing documents (1 ms)

**PASS** dist/document/services/document.service.test.js**Document Service**

- ✓ Should find a list of documents (5 ms)

**Unit Testing (Server-side) : coverage report**

File ▲		Statements ⇅		Branches ⇅		Functions ⇅		Lines ⇅	
account/controllers/src/account/controllers	<div></div>	100%	13/13	100%	2/2	100%	3/3	100%	10/10
account/services/src/account/services	<div></div>	100%	36/36	100%	7/7	100%	4/4	100%	32/32
document/controllers/src/document/controllers	<div></div>	100%	13/13	100%	2/2	100%	3/3	100%	10/10
document/services/src/document/services	<div></div>	100%	16/16	100%	0/0	100%	3/3	100%	13/13
document/utls/src/document/utls	<div></div>	100%	7/7	75%	12/16	100%	2/2	100%	6/6
tree/controllers/src/tree/controllers	<div></div>	100%	29/29	100%	6/6	100%	7/7	100%	22/22
tree/models/src/tree/models	<div></div>	100%	17/17	100%	0/0	100%	3/3	100%	15/15
tree/services/src/tree/services	<div></div>	100%	40/40	100%	0/0	100%	7/7	100%	33/33
utls/src/utls	<div></div>	100%	7/7	100%	0/0	100%	2/2	100%	6/6

## Report del testing della REST API

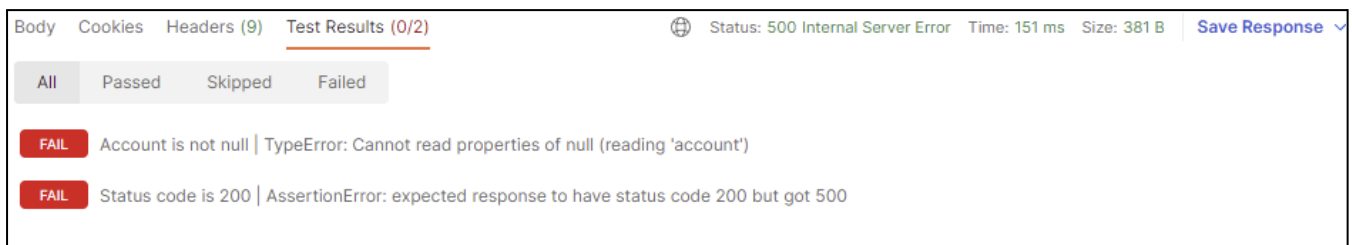
Per consultare i test eseguiti, riferirsi al Test Plan, sezione 9.2, la quale contiene inoltre gli identificativi associati a ciascun caso di test. Ulteriori endpoint non considerati da Category Partition sono esplicitati separatamente.

### Incident Report: "BirthDate" is undefined - Test case: R-R-TC1

La data di nascita all'atto dell'inserimento di un nodo non viene inserita affatto. Dopo un'analisi del codice di AccountController, si riconduce il problema a

- La mancata inizializzazione della data nel metodo `createNode()`;
- Un (banale) errore di spelling tra i campi della richiesta HTTP simulata mediante Postman (i.e. "birtDate" anziché "birthDate").

Si comunica che è stato effettuato regression testing della REST API, in particolare dei test cases di registrazione in seguito all'incident ivi descritto.



In seguito alla correzione ed al regression testing, ogni test si è concluso con un PASS, come mostrato di seguito.

## Quiad - Run results

Run Again

Automate Run ▾

+ New Run

↗ Export Results

👤 Run on Today, 12:13:56 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	2s 586ms	45	25 ms

[All Tests](#)   Passed (45)   Failed (0)   Skipped (0)[View Summary](#)

Iteration 1

GET	GetNodes	{{BASE_URL}}/nodes/1	/ Tree / GetNodes	200 OK	15 ms	2.576 KB
	Pass	Status code is 200				
	Pass	Nodes owner check				
POST	R-I-TC1	{{BASE_URL}}/nodes	/ Tree / R-I-TC1	200 OK	15 ms	612 B
	Pass	Status code is 200				
	Pass	Node is not null				
POST	R-I-TC2	{{BASE_URL}}/nodes	/ Tree / R-I-TC2	200 OK	15 ms	600 B
	Pass	Status code is 200				
	Pass	Node is not null				
POST	R-I-TC3	{{BASE_URL}}/nodes	/ Tree / R-I-TC3	200 OK	60 ms	543 B
	Pass	Status code is 200				
	Pass	Account is not null				
PATCH	R-M-TC1	{{BASE_URL}}/nodes/4	/ Tree / R-M-TC1	200 OK	15 ms	620 B
	Pass	Status code is 200				
	Pass	Node is not null				



PATCH	R-M-TC2	{{(BASE_URL)}}/nodes/4	/ Tree / R-M-TC2	200 OK	16 ms	550 B
	Pass	Status code is 200				
	Pass	Node is not null				
DELETE	R-E-TC1	{{(BASE_URL)}}/nodes/2	/ Tree / R-E-TC1	500 Internal Server Error	3 ms	381 B
	Pass	Status code is 500				
	Pass	Node is null				
DELETE	R-E-TC2	{{(BASE_URL)}}/nodes/14	/ Tree / R-E-TC2	200 OK	13 ms	585 B
	Pass	Status code is 200				
	Pass	Node is not null				
DELETE	R-E-TC3	{{(BASE_URL)}}/nodes/300	/ Tree / R-E-TC3	500 Internal Server Error	8 ms	381 B
	Pass	Status code is 500				
	Pass	Node is null				
PUT	R-AS-TC1	{{(BASE_URL)}}/nodes/4/bind/1	/ Tree / R-AS-TC1	200 OK	16 ms	559 B
	Pass	Status code is 200				
	Pass	Node is not null				
PUT	R-AS-TC2	{{(BASE_URL)}}/nodes/700/bind/1	/ Tree / R-AS-TC2	500 Internal Server Error	9 ms	381 B
	Pass	Status code is 500				
	Pass	Node is null				

PUT	R-AS-TC3	{{BASE_URL}}/nodes/4/bind/3	/ Tree / R-AS-TC3	500 Internal Server Error	9 ms	381 B
	Pass	Status code is 500				
	Pass	Node is null				
PUT	R-DS-TC1	{{BASE_URL}}/nodes/4/unbind/1	/ Tree / R-DS-TC1	200 OK	20 ms	550 B
	Pass	Status code is 200				
	Pass	Node is not null				
PUT	R-DS-TC2	{{BASE_URL}}/nodes/700/unbind/1	/ Tree / R-DS-TC2	500 Internal Server Error	12 ms	381 B
	Pass	Status code is 500				
	Pass	Node is null				
PUT	R-DS-TC3	{{BASE_URL}}/nodes/4/unbind/100	/ Tree / R-DS-TC3	500 Internal Server Error	5 ms	381 B
	Pass	Status code is 500				
	Pass	Node is null				
POST	R-A-TC1	{{BASE_URL}}/login	/ Account / R-A-TC1	200 OK	76 ms	1.64 KB
	Pass	Account is not null				
	Pass	Password is undefined				
	Pass	Status code is 200				

<b>POST</b>	R-A-TC2	{{BASE_URL}}/login	/ Account / R-A-TC2	401 Unauthorized	80 ms	372 B
Pass	Rspnse is null					
Pass	Status code is 401					
<b>POST</b>	R-A-TC3	{{BASE_URL}}/login	/ Account / R-A-TC3	401 Unauthorized	7 ms	372 B
Pass	Rspnse is null					
Pass	Status code is 401					
<b>POST</b>	R-R-TC1	{{BASE_URL}}/register	/ Account / R-R-TC1	200 OK	81 ms	620 B
Pass	Account is not null					
Pass	Status code is 200					
<b>POST</b>	R-R-TC2	{{BASE_URL}}/register	/ Account / R-R-TC2	500 Internal Server Error	77 ms	381 B
Pass	Account is null					
Pass	Status code is 500					
<b>GET</b>	R-RD-TC1	{{BASE_URL}}/documents?retrievalPlace=Roma&originPlace=Roma	/ Document / R-RD-TC1	200 OK	3 ms	630 B
Pass	Status code is 200					
Pass	Your test name					
<b>GET</b>	R-RD-TC2	{{BASE_URL}}/documents?categoryId=2	/ Document / R-RD-TC2	200 OK	3 ms	438 B
Pass	Status code is 200					
Pass	Your test name					

Report completo REST API:

	1
▶ GET GetNodes	2 0
▶ POST R-I-TC1	2 0
▶ POST R-I-TC2	2 0
▶ POST R-I-TC3	2 0
▶ PATCH R-M-TC1	2 0
▶ PATCH R-M-TC2	2 0
▶ DELETE R-E-TC1	2 0
▶ DELETE R-E-TC2	2 0
▶ DELETE R-E-TC3	2 0
▶ PUT R-AS-TC1	2 0
▶ PUT R-AS-TC2	2 0
▶ PUT R-AS-TC3	2 0
▶ PUT R-DS-TC1	2 0
▶ PUT R-DS-TC2	2 0
▶ PUT R-DS-TC3	2 0
▶ POST R-A-TC1	3 0
▶ POST R-A-TC2	2 0
▶ POST R-A-TC3	2 0
▶ POST R-R-TC1	2 0
▶ POST R-R-TC2	2 0
▶ GET R-RD-TC1	2 0
▶ GET R-RD-TC2	2 0

Dove sono stati testati anche gli endpoint ulteriori:

**GetNodes:** GET /nodes/1

L'endpoint a cui rivolgersi per ottenere la lista dei nodi di un albero associato ad un dato utente.

**Oracolo:** ogni nodo nella risposta HTTP ha ownerId = 1.

**GetDocument:** GET /documents/1

L'endpoint a cui rivolgersi per visualizzare un documento.

**Oracolo:** nella risposta HTTP vi è il documento con ID = 1

## **Report del testing di unità client-side**