



Oggetto: Documentazione di Progetto "Sciurus"

Versione: 1.0

Data: 03/2024

Corso: Programmazione Avanzata

Anno Accademico: 2023-2024

Sviluppatore: Alex Moriconi (615958)

Email: a.moriconi9@studenti.unipi.it

Indice dei Contenuti

Introduzione a “Sciurus”	3
Origine del nome	3
Campo d’impiego	3
Struttura di un articolo	3
Diagramma delle componenti	4
Componenti del Software	4
Interfaccia di Accesso	4
Complessivo degli Articoli	5
Inserimento nuovo Articolo	6
Specifiche di Implementazione	7
Struttura del Database	7
Metodi Nativi Utilizzati:	7
Query Native:	7
API (Application Programming Interface)	8
Java JUnit Test	8
File Tree “sciurus-server” & “sciurus-client”	8

Introduzione a “Sciurus”

Origine del nome

L'applicazione di cui tratta la seguente documentazione è denominata “Sciurus”, termine latino per indicare gli scoiattoli.

L'attribuzione del nome fa riferimento alla capacità di questi animali di adoperarsi all'organizzazione delle provviste con ordine e rapidità.

Questa scelta suggerisce che l'applicazione possieda caratteristiche simili, mirando a offrire funzionalità che permettano agli utenti di organizzare articoli in modo efficiente e rapido, proprio come gli scoiattoli organizzano le loro risorse.

Campo d'impiego

L'applicazione “Sciurus” è stata appositamente progettata per offrire agli utenti singoli la possibilità di gestire articoli testuali in modo semplice e intuitivo su una rete privata sicura. Con “Sciurus”, gli utenti possono memorizzare, creare ed eliminare articoli con facilità, senza la necessità di complessi sistemi di autenticazione e autorizzazione degli utenti.

Sviluppata per operare su reti private protette, “Sciurus” assicura un accesso diretto e senza ostacoli ai dati per ogni istanza del client. Questo significa che gli utenti hanno il pieno controllo su tutti gli aspetti dell'applicazione, senza restrizioni di accesso o gerarchie di autorizzazione interne.

L'approccio senza fronzoli di “Sciurus” si adatta perfettamente agli ambienti in cui la sicurezza della rete è già attentamente gestita esternamente. In contesti come reti aziendali o intranet, l'applicazione si integra perfettamente, offrendo un'esperienza utente fluida senza richiedere ulteriori passaggi di autenticazione.

Struttura di un articolo

Il singolo articolo memorizzato è composto da cinque campi fondamentali di cui solo tre editabili dall'utente.

{ID, Titolo, Argomento, Corpo, Data di Creazione}

ID → Indice incrementale di identificazione univoca dell'articolo (Gestito dal software)

Titolo → Titolo dell'articolo

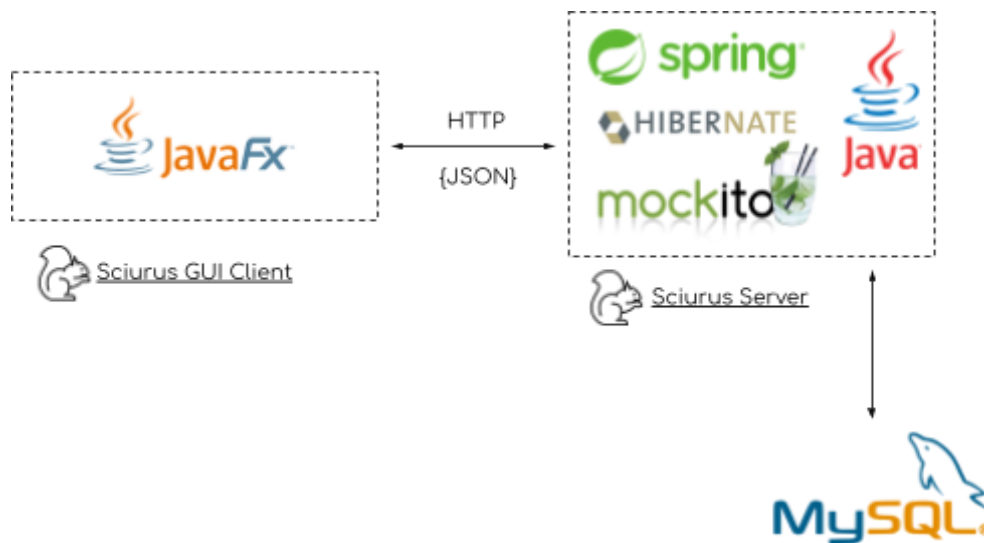
Argomento → Argomento di cui tratta l'articolo

Corpo → Contenuto effettivo dell'articolo

Data di Creazione → Data di inserimento dell'articolo (Gestita dal software)

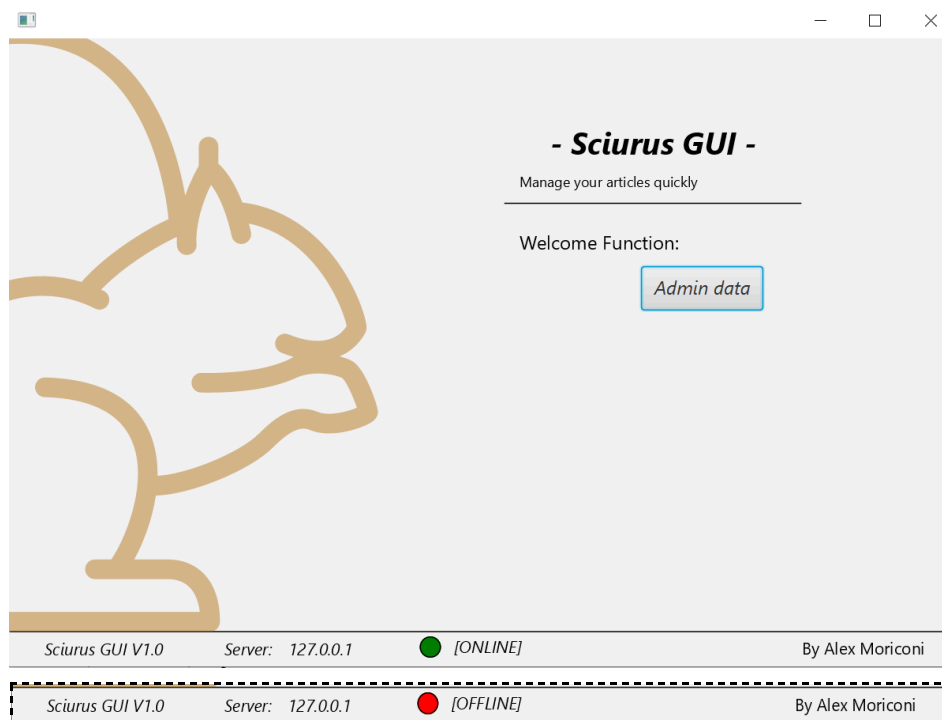
Nota: Come vincolo aggiuntivo è stato inserito che non vi possano essere articoli con titolo strettamente uguale. Questo controllo è effettuato ad ogni nuovo inserimento.

Diagramma delle componenti



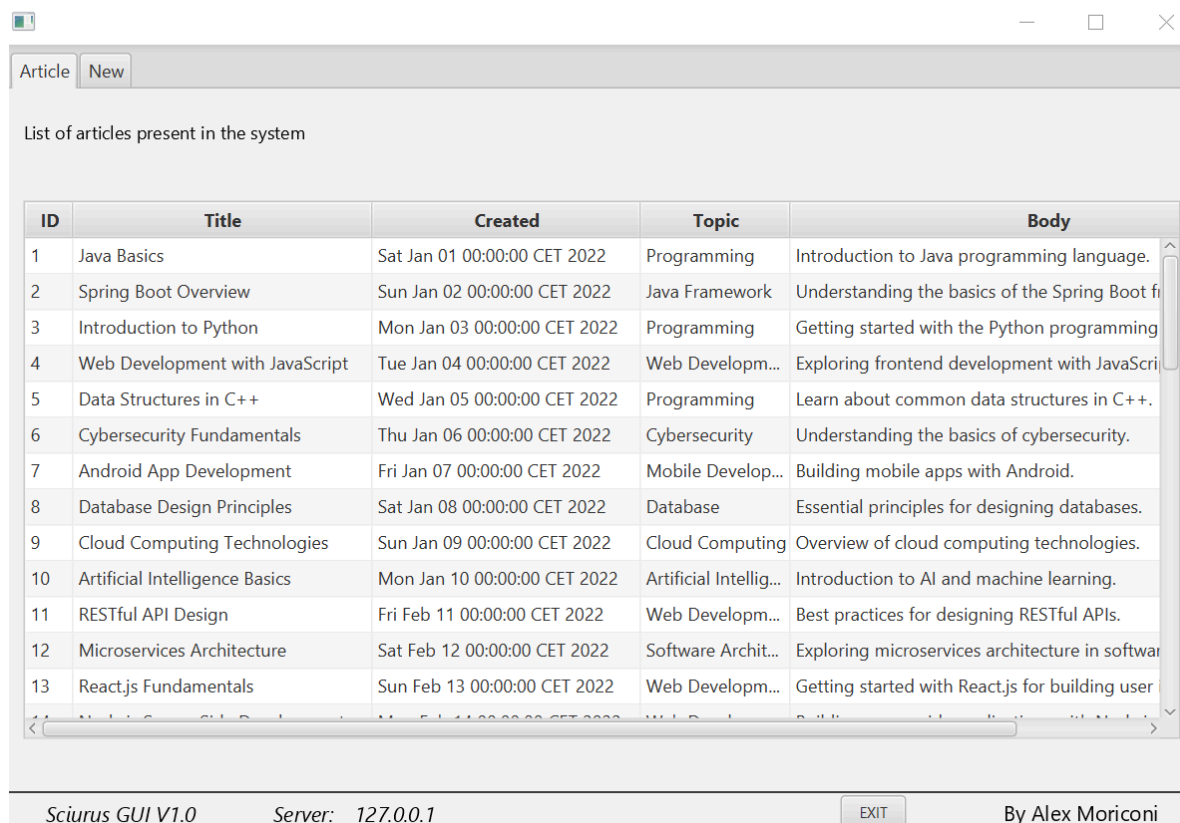
Componenti del Software

Interfaccia di Accesso



Interfaccia di avvio dell'applicazione: riporta la versione del client in esecuzione, l'indirizzo IP del server Sciurus ed infine lo stato all'avvio del server [ONLINE] o [OFFLINE]. Attraverso il bottone "Admin data" l'utente potrà aprire l'interfaccia di amministrazione degli articoli. Alla pressione verrà verificato lo stato del server e se e solo se quest'ultimo risulta online sarà permesso l'accesso all'interfaccia successiva.

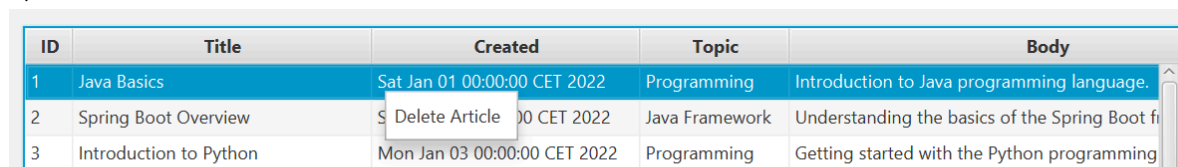
Comlessivo degli Articoli



ID	Title	Created	Topic	Body
1	Java Basics	Sat Jan 01 00:00:00 CET 2022	Programming	Introduction to Java programming language.
2	Spring Boot Overview	Sun Jan 02 00:00:00 CET 2022	Java Framework	Understanding the basics of the Spring Boot framework.
3	Introduction to Python	Mon Jan 03 00:00:00 CET 2022	Programming	Getting started with the Python programming language.
4	Web Development with JavaScript	Tue Jan 04 00:00:00 CET 2022	Web Development	Exploring frontend development with JavaScript.
5	Data Structures in C++	Wed Jan 05 00:00:00 CET 2022	Programming	Learn about common data structures in C++.
6	Cybersecurity Fundamentals	Thu Jan 06 00:00:00 CET 2022	Cybersecurity	Understanding the basics of cybersecurity.
7	Android App Development	Fri Jan 07 00:00:00 CET 2022	Mobile Development	Building mobile apps with Android.
8	Database Design Principles	Sat Jan 08 00:00:00 CET 2022	Database	Essential principles for designing databases.
9	Cloud Computing Technologies	Sun Jan 09 00:00:00 CET 2022	Cloud Computing	Overview of cloud computing technologies.
10	Artificial Intelligence Basics	Mon Jan 10 00:00:00 CET 2022	Artificial Intelligence	Introduction to AI and machine learning.
11	RESTful API Design	Fri Feb 11 00:00:00 CET 2022	Web Development	Best practices for designing RESTful APIs.
12	Microservices Architecture	Sat Feb 12 00:00:00 CET 2022	Software Architecture	Exploring microservices architecture in software development.
13	React.js Fundamentals	Sun Feb 13 00:00:00 CET 2022	Web Development	Getting started with React.js for building user interfaces.

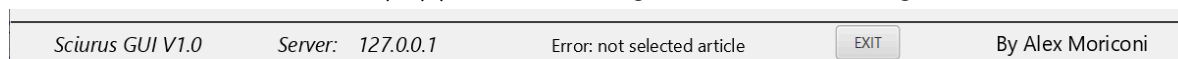
In questa schermata sono visibili tutti gli articoli presenti attualmente sul server. Per favorire la visualizzazione e/o la ricerca cliccando ogni singola colonna è possibile ricorrere ad un ordinamento parziale (crescente o decrescente). Attraverso il bottone "EXIT" sarà possibile tornare in ogni momento alla schermata di avvio.

Cliccando con il tasto destro del mouse sopra un record della tabella Articoli sarà possibile rimuovere un articolo, eliminandolo definitivamente dal server. Questa operazione non è reversibile.



ID	Title	Created	Topic	Body
1	Java Basics	Sat Jan 01 00:00:00 CET 2022	Programming	Introduction to Java programming language.
2	Spring Boot Overview	Sun Jan 02 00:00:00 CET 2022	Java Framework	Understanding the basics of the Spring Boot framework.
3	Introduction to Python	Mon Jan 03 00:00:00 CET 2022	Programming	Getting started with the Python programming language.

Nel caso di errori od anomalie gestite (Esempio: richiesta di eliminazione senza nessun articolo selezionato) apparirà il dettaglio di errore nel seguente box.



Sciurus GUI V1.0	Server: 127.0.0.1	Error: not selected article	EXIT	By Alex Moriconi
------------------	-------------------	-----------------------------	------	------------------

Selezionando la tab "New" in qualsiasi momento sarà possibile accedere all'interfaccia di inserimento di un nuovo articolo.

Inserimento nuovo Articolo

Article New

Interface for inserting a new article

Title

Topic

Body

add

Sciurus GUI V1.0 Server: 127.0.0.1 EXIT By Alex Moriconi

Attraverso questa visualizzazione sarà possibile inserire un nuovo articolo nel server Scirus. Ogni articolo dovrà obbligatoriamente avere contenuto per tutti i suoi campi (Titolo, Argomento, Corpo) e se e solo se il requisito sarà soddisfatto l'articolo potrà essere aggiunto con il bottone "add".

Title Nuvo Articolo

Topic Simulazione di Inserimento

Body

add

Il software controlla in autonomia lo stato dei campi di input alla pressione del bottone "add", nello specifico se un campo risulta vuoto o non valido ne dispone la label rispettiva di colore rosso al fine di segnalare una difformità; di colore verde se il contenuto è corretto.

Title

Topic Simulazione di Inserimento

add

Alla pressione di "add" seguirà la visualizzazione di "OK" se l'inserimento del nuovo articolo ha avuto successo altrimenti "FAIL" in caso di errore (Esempio: titolo di articolo già esistente). La specifica dell'errore potrà apparire nello stesso spazio di descrizione errori visto nell'interfaccia "Complessivo degli Articoli"

Specifiche di Implementazione

Struttura del Database

L'applicazione all'avvio verifica l'esistenza del Data Base 615958, in caso di mancanza procede con la creazione e popolamento con dati iniziali presenti nel file "data.sql". La struttura della DB così realizzata è la seguente:



Schermata ricavata dall'interfaccia di un software ausiliario utilizzato in fase di sviluppo "MySQL WorkBench"

Come da specifiche di progetto le credenziali di accesso all'istanza MySQL sono:

IP: 127.0.0.1 (localhost)

User: root

Password: root

Per la creazione del DB all'avvio in caso di inesistenza è stata utilizzata una funzionalità Spring JPA di interfaccia con MySQL:

```
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:127.0.0.1}:3306/615958?createDatabaseIfNotExist=true
```

Per la verifica specifica della tabella "Article" è stata realizzata una classe "DatabaseInitializer" con un metodo "@Bean" che analizza ciò all'avvio di Spring attraverso una query al information_schema.

```
"SELECT 1 FROM information_schema.tables WHERE table_schema = '615958' AND table_name = 'Article'"
```

L'interazione con il database nel modulo "Sciurus Server" è stata realizzata utilizzando Spring JPA creando un JavaBean modellante i dati della tabella Article. Sono stati utilizzati alcuni metodi nativi di Spring JPA per le query tuttavia per funzionalità più specifiche sono state implementate delle query "native".

Metodi Nativi Utilizzati:

- findByTitle(String Title)
- findById(int Id)
- deleteById(int id)

Query Native:

- @Query(value= "SELECT COALESCE(MAX(id), 0) + 1 FROM Article", nativeQuery = true)

API (Application Programming Interface)

“Sciurus Server” rende disponibili le seguenti API operanti su protocollo HTTP e codifica dei dati scambiati in formato JSON. Specifiche dettagliate sul Javadoc di accompagnamento.

- (GET) /sciurus/status
- (GET) /sciurus/article/all
- (POST) /sciurus/article/new
- (DELETE) /sciurus/article/remove

Java JUnit Test

“Server Sciurus” presenta una Unit Java Test “ServerControllerTest.java” destinata al test della API /sciurus/status. Per l’implementazione dei test è stato utilizzato il framework “Mockito” al fine di emulare gli oggetti reali e specificare il ritorno in modo arbitrario.

File Tree “sciurus-server” & “sciurus-client”

