

GESTIONE CASINÒ | Diario di lavoro - 22.02.2019

Matan Davidi, Thor Düblin, Matteo Forni, Carlo Pezzotti, Mattia Toscanelli

Trevano, 22 Febbraio 2019

Lavori svolti

Durante la giornata di oggi Matan ha terminato lo schema logico usato per la progettazione del database, che trovate di seguito.

database bisogna passare come parametro le seguenti informazioni ([host],[porta],[nomeDatabase],[username],[password]):

```
private $db;

function __construct($host,$port,$dbname,$username,$password)
{
    $this->db = new PDO("mysql:host=$host;port=$port;dbname=$dbname", $username, $password);
}
```

i metodi che ho reputato di essere utili sono i seguenti:

- executeQuery --> metodo che data una query passata come parametro la esegue e ritorna il risultato

```
public function executeQuery($query){
    $result = $this->db->query($query);
    if ($result === FALSE) {
        throw new InvalidArgumentException("Failed to load schema is not exists or you are not permission");
    }
    return $result->fetch();
}
```

- printTableQuery --> metodo che data una query di select stampa una tabella html

```
public function printTableQuery($selectQuery){
    $result = $this->db->query($selectQuery);
    if ($result === FALSE) {
        throw new InvalidArgumentException("Failed to load schema is not exists or you are not permission");
    }
    echo "<table id='tbl'><tr>";
    while($row = $result->fetch())
    {
        echo "<tr>";
        for ($i=0;$i<(sizeof($row)-1);$i++)
        {
            echo "<td>".$row[$i]."</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
}
```

- insertUser --> metodo che dato un utente passato da parametro lo inserisce nel database.

```
public function insertUser($user){
    if(gettype($user) == "object"){
        if(get_class($user) == "User"){
            $query = "Insert into users(id,nome,cognome) values(null, '".$user->getName()."', '".$user->getSurname()."' )";
            $this->executeQuery($query);
        }else{
            throw new InvalidArgumentException(get_class($user)." is not a User class");
        }
    }else{
        throw new InvalidArgumentException(gettype($user)." is not a User class");
    }
}
```

La classe di test non fa altro che testare la classe sia a livello globale che a livello modulare dei singoli metodi

```
public function testDatabaseConnect(): void
{
    $this->assertInstanceOf(Database::class,new Database("127.0.0.1",3306,"provaCasino","root","root"));
}
public function testCannotConnect(): void
{
    $this->expectException(PDOException::class);
    $this->assertInstanceOf(Database::class,new Database("127.0.0.2",3306,"provaCasino","root","root"));
}
public function testInserUser() : void{
    $db = new Database("127.0.0.1",3306,"provaCasino","root","root");
    $lastId = $db->executeQuery("select max(id) from users")[0];
    $db->insertUser(new User(
        "Carlo",
        "Pezzotti",
        "2000-12-01",
        "Capolago",
        "Via laveggio",
        9,
        "0788159957",
    ));
}
```

```

        "carlo.pezzotti@samtrevano.ch",
        "male",
        "Password&1",
        "Password&1"
    ));
    $nowId = $db->executeQuery("select max(id) from users")[0];
    $this->assertTrue(($lastId+1 == $nowId));
}

public function testCannotInsertUser():void{
    $this->expectException(InvalidArgumentException::class);
    $db = new Database("127.0.0.1",3306,"provaCasino","root","root");
    $db->insertUser("prova");
}

```

Matteo oggi ha risolto il problema della scorsa lezione con l'aiuto di Carlo per la quale non si riusciva a collegare Jenkins a Github e quindi ha messo in comunicazione i due sistemi facendo sì che quando si esegue un push il sistema esegue i test di prova creati in precedenza da Carlo. Inoltre ha iniziato a scrivere la documentazione sull'installazione e configurazione di Jenkins portando a termine la prima parte.

Problemi riscontrati e soluzioni adottate

Carlo ha sollevato il discorso dell'indice utilizzato per la tabella 'User' del database: secondo lui, infatti, sarebbe meglio aggiungere una colonna 'id' all'entity set perché questo renderebbe le operazioni in PHP più semplici e veloci, mentre Matan afferma che così facendo si aggiungerebbe una colonna inutile all'entità che appesantirebbe qualsiasi operazione su di essa.

Abbiamo intenzione di discuterne al prossimo stand-up.

Inoltre Matan ha avuto problemi a scannerizzare il foglio sul quale ha disegnato il diagramma E/R e lo schema logico sulla stampante di scuola. Alla fine ha dovuto salvare il risultato della scansione, sotto forma di file PDF, su una chiavetta USB, utilizzando il menù "Pannello USB".

Thor ha risolto il problema degli errori quando digitava "mvn clean install" nella cartella del progetto di Selenium. I problemi erano causati da un'errata configurazione del file pom.xml, che non riconosceva delle versioni di Maven.

Il seguente codice è quello modificato, che permette il corretto funzionamento del "mvn clean install". Inoltre ha installato Eclipse.

Mattia Toscanelli ha avuto difficoltà ad accedere al server ftp in quanto aveva dimenticato i dati d'accesso. Inoltre ha avuto problemi nel mettere responsive l'input select. Dopo qualche modifica in CSS è riuscito.

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>MySel20Proj</groupId>
    <artifactId>MySel20Proj</artifactId>

```

Un altro problema riscontrato oggi è quello della configurazione proxy da parte di Maven, che non permetteva lo scaricamento da internet, per risolvere questo problema, è bastato andare nella cartella conf di Maven ed aggiungere il codice per la configurazione del Proxy.

```

<proxy>
    <id>cpt.local</id>
    <active>true</active>
    <protocol>http</protocol>
    <host>proxy.cpt.local</host>
    <port>8080</port>
    <username>proxyuser</username>
    <password>somepassword</password>
</proxy>

```

Il problema nel collegamento era dovuto alle chiavi SSH e non è stato risolto ma semplicemente aggirato utilizzando al posto delle chiavi un webhook dato che la repository di github è pubblica e non necessita quindi di un'autenticazione. Il webhook creato è il seguente:

Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type

Secret

Which events would you like to trigger this webhook?

- ☒ Just the push event.
- ☐ Send me **everything**.
- ☐ Let me select individual events.

☒ Active

We will deliver event details when this hook is triggered.

In seguito a ciò abbiamo riscontrato un altro problema che è quello per cui non vi è un'impostazione propria di GitHub nei WebHook per eseguire il trigger dopo un commit e quindi Carlo ha iniziato a scrivere uno script che esegua questo compito.

Punto della situazione rispetto alla pianificazione

Siamo in orario con la pianificazione

Programma di massima per la prossima giornata di lavoro

Matan deve preoccuparsi di mostrare lo schema logico al cliente in modo da poter correggere eventuali errori.

Mattia continuerà la pagina di password smarrita e la pagina di verifica della email.

Carlo Pezzotti si occuperà di fare una classe che si occuperà di mettere insieme tutte le classi sviluppate fin ora

Matteo la prossima lezione vuole riuscire a far funzionare correttamente GitHub con Jenkins e terminare la documentazione riguardante l'installazione e configurazione di Jenkins.