

# More on Connecting to a Database

CS174

Chris Pollett

Dec 4, 2006.

# Outline

- Connecting a Web applications to a Database in Perl
- Connecting a Web Application to a Database in Java.

# Perl

- Perl has a standard object-oriented module for connecting to a database called DBI.
- An application program uses the methods and attributes of DBI to specify database access that will be accessed and how it is accessed.
- Each database requires its own database driver module (DBD) to access the physical database.

# Getting the Software

- First you need to get both DBI and the DBD::mysql packages from CPAN.
- These can be found at:

<http://search.cpan.org/~timb/DBI/DBI.pm>

<http://search.cpan.org/~capttofu/DBD-mysql-3.0002/lib/DBD/mysql.pm>

- There might be newer versions so you can always do a search.
- One downloads these files
- Then un-gzip and un-tar them.
- Install DBI first at the command prompt in the relevant directory for each do:

```
perl Makefile.pl
```

```
make
```

```
make install
```

# Example DBI program

```
use DBI;
```

```
$dbh = DBI->connect("DBI:mysql:test") || die "Error -- $dbh->errstr\n"; #test was the database used
```

```
$stmt = $dbh->prepare("SELECT * FROM bob"); #bob was a table with two columns:
```

```
    #column a var an inst and column b was a varchar
```

```
$stmt->execute() || die "Error execute: $dbh->errstr\n";
```

```
$names = $stmt->{NAME}; # ptr to names of columns
```

```
foreach $field (@$names)
```

```
{
```

```
    print "Field: $field\n"; # print out names of columns
```

```
}
```

```
while(@result_rows = $stmt->fetchrow_array) #get a row
```

```
{
```

```
    while ($#result_rows >= 0) #print that row
```

```
    {
```

```
        $field = shift @result_rows;
```

```
        print $field. " ";
```

```
    }
```

```
    print "\n";
```

```
}
```

```
$stmt -> finish; #done with statement
```

```
$dbh->disconnect(); #close connection
```

# Some Comments

- To do things like insert/delete update we still use prepare:

```
$stmt = $dbh->prepare("INSERT INTO bob values (5, 'ed')");
```

- We then call `$stmt->execute()` like before; we just don't try to do things like `fetchrow_array` afterwards.
- Although the previous program was designed to be run from the command line, we could have used CGI.pm and done the whole thing as a CGI program.

# Connecting with Java to MySQL database

- You need to first get the driver for MySQL from:  
<http://dev.mysql.com/downloads/connector/j/5.0.html>
- You need to unzip the file and copy the Jar file to somewhere useful.
- Then add the jar file to your classpath and you should be ready to go.

# Example

```
import java.sql.*;
public class JDBCTest
{
    public static void main(String[] args)
    {
        try
        {
            Class.forName("org.gjt.mm.mysql.Driver").newInstance();
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/test"); //test is the name of the database
            Statement stmt = conn.createStatement(); //test has one table bob which is the same as in the perl example
            String select="SELECT * FROM bob";
            ResultSet result = stmt.executeQuery(select); // use executeUpdate for inserts/etc
            while(result.next())
            {
                System.out.println("A column:"+result.getInt("a")); // if column was a varchar then use getString
            }
        }
        catch(Exception e) /*Class.forName can give a ClassNotFoundException, JDBC stuff can give
            SQLExceptions */
        {
            e.printStackTrace();
        }
    }
}
```