**EXERCISES**

|  |  |
| --- | --- |
|  | Create a servlet that displays the current date and time. |
|  | Create a HTML form with three input fields: first name, last name and e-mail. Use the get method to pass these values to a servlet. Notice how data is attached to the URL. In the servlet, verify all input fields are non-null and display them back to the client. |
|  | Create a Web application for dynamic FAQs.  The application should obtain the information to create the dynamic FAQ Web page from a database that consists of a topics table and an faq table.  The topics table has two fieldsa unique integer ID for each topic (topicID) and a name for each topic (topicName).  The faq table has three fieldsthe topicID (a foreign key), a string representing the question (question) and the answer to the question (answer).  When the servlet is invoked, it should read the data from the database and return a dynamically created Web page containing each question and answer, sorted by topic.  [ **Note:** The examples folder for this chapter contains the SQL script faq.sql with which you can create the faq database for this example. For information on executing the SQL script, please refer to [Chapter 25](http://underpop.online.fr/j/java/help/accessing-databases-with-jdbc.html.gz#ch25).]  In the servlet's init method, create a CachedRowSet and set the database URL for the CachedRowSet. Sun's reference implementation of CachedRowSet is named CachedRowSetImpl and is located in the package com.sun.rowset. |
|  | Modify the Web application in [Exercise 26.5](http://underpop.online.fr/j/java/help/exercises-servlets.html.gz#ch26qa2q3) so that the initial request to the servlet returns a Web page of topics in the FAQ database. Then, the user can link to another servlet that returns only the frequently asked questions for a particular topic. |
|  | Modify the Web application in [Example 26.21](http://underpop.online.fr/j/java/help/multitier-applications-using-jdbc-from-a-servlet-servlets.html.gz#ch26table19) to allow the user to see the survey results without responding to the survey. |
|  | Recall that the Web application in [Example 26.21](http://underpop.online.fr/j/java/help/multitier-applications-using-jdbc-from-a-servlet-servlets.html.gz#ch26table19) implements a Web-based survey.  Write a Web application that can be used generically with any survey of the appropriate form i.e., a question followed by a set of possible answers. Your Web application should have three servlets. The first is called to dynamically generate a list of survey names. When the user selects a survey, the second servlet should dynamically generate a form containing the survey options. When the user chooses an option, the third servlet should update the database and return the survey results. The survey database for this exercise has two tablessurveyCategory and surveyResult. Table surveyCategory has three fieldsa unique integer ID for each survey category (id), a string representing the survey name (name) and a string representing the survey question (question). Table surveyResult has three fieldsan integer ID (a foreign key) that identifies the survey category (id), a string representing the survey option (surveyOption) and an integer representing the total votes that option has received so far (voteCount). [**Note:** The examples folder for this chapter contains the SQL script survey.sql with which you can create the survey database for this example. For information on executing the SQL script, refer to [Chapter 25](http://underpop.online.fr/j/java/help/accessing-databases-with-jdbc.html.gz#ch25). The sample database contains sample data for three surveysAnimals, Fruits and Sports.] |
|  | Write a Web application that consists of a servlet (DirectoryServlet) and several Web documents. Document index.html should be the first document the user sees. In it, you should have a series of links for other Web pages in your site. When clicked, each link should invoke the servlet with a get request that contains a page parameter. The servlet should obtain parameter page and redirect the request to the appropriate document. |