

JSON Data

Let's take a second to back up and look at the ability to read and write JSON data. Using the example code shown in this chapter, only the constructor and destructor will need to be adjusted when you use other forms of data besides XML. Accessing and using JSON data is even easier than using XML data.

```
...
$json = file_get_contents($this->dog_data_JSON);
$this->dogs_array = json_decode($json,TRUE);
if ($this->dogs_array === null && json_last_error() !== JSON_ERROR_NONE)
{
    throw new Exception("JSON error: " . json_last_error_msg());
}
...
```

In the constructor, after the `if else` structure that retrieves the location of the data from the `dog_application.xml` file, the multiple lines accessing and formatting the XML data can be replaced by the lines shown previously. The `json_decode` method (as shown previously) will attempt to format the data from the text file into the associate array format. If the data is not a valid JSON format, an exception is thrown passing the error message. Since the `Exception` class is used, the `dog_interface` program would log this information in the error log, e-mail the support personnel, and display a general message to the users.

```
$json = json_encode($this->dogs_array);
file_put_contents($this->dog_data_JSON,$json);
```

The complete code for the destructor requires only two lines. The `json_encode` method will convert the associate array data into JSON format. The `file_put_contents` method will then save the information to the proper location of the JSON file (`$this->dog_data_JSON`). No changes are required to any of the other methods in `dog_data`. Note: An example application using JSON data is available on the book's web site under Chapter 6.

MySQL Data

This book is intended as an introduction to the PHP language. Thus, you will not spend much time learning about database usage. However, this is a good time to give a brief example of adjustments you can make to the constructor and destructor methods to access and update database information.

■ **Note** `mysql` has been removed since PHP5.5. It is recommended that you use `mysqli` or `pdo_mysql`.

```
$mysqli =mysqli_connect($server, $db_username, $db_password, $database);

if (mysqli_connect_errno())
{
    throw new Exception("MySQL connection error: " . mysqli_connect_error());
}

$sql="SELECT * FROM Dogs";
$result=mysqli_query($con,$sql);
```