

02/11/11
02:18:37

Olive README

1

Olive makes it easy to edit your videos.

02/11/11
02:18:37

Olive

WebContent/META-INF/context.xml

1

```
<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/olive">
  <Resource name="jdbc/OliveData" auth="Container" type="javax.sql.DataSource"
    username="olive" password="Xsbpdrhr" driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/" maxActive="15" maxIdle="3" />
</Context>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Application Server 9.0
Servlet 2.5//EN" "http://www.sun.com/software/appserver/dtds/sun-web-app_2_5-0.dtd"
">
<sun-web-app error-url="">
  <context-root>/olive</context-root>
  <class-loader delegate="true" />
  <jsp-config>
    <property name="keepgenerated" value="true">
      <description>Keep a copy of the generated servlet class' java code.</de
scription>
    </property>
  </jsp-config>
</sun-web-app>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns
  /javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/n
  s/javaee/web-app_2_5.xsd"
  version="2.5">
  <servlet>
    <servlet-name>OliveServlet</servlet-name>
    <servlet-class>com.readytalk.olive.servlet.OliveServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>OliveServlet</servlet-name>
    <url-pattern>/OliveServlet</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <resource-ref>
    <description>Database</description>
    <res-ref-name>jdbc/OliveData</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>
</web-app>
```

```
/* -----  
   Edit account page  
   -----  
*/  
#edit-account-container {  
    padding: 5%;  
}  
  
#edit-account-container h2 {  
    margin: .75em 0;  
}
```

```
/*
-----
Editor page
-----
*/
#videos-container {
    position: absolute;
    top: 0px;
    right: 70%;
    bottom: 160px;
    left: 0px;
    text-align: center;
    /* 100% - (video width) - (2 x (sum of padding, border, margin)) */
}

#videos-title {
    position: absolute;
    top: 0%;
    right: 0%;
    bottom: 10%;
    left: 0%;
    margin: 5px;
}

#videos-controls {
    position: absolute;
    top: 10%;
    right: 0%;
    bottom: 20%;
    left: 0%;
    border-top: 1px solid gray;
    border-right: 1px solid gray;
    border-bottom: 0px solid gray; /* different */
    border-left: 1px solid gray;
}

#videos {
    position: absolute;
    top: 20%;
    right: 0%;
    bottom: 0%;
    left: 0%;
    text-align: left;
    border: 1px solid gray;
    overflow: auto;
}

.video-container {
    display: block;
    float: left;
    width: 100px;
    margin: 10px;
    padding: 10px;
    border: 1px solid green;
    background-color: #FEFEFE;
    text-align: center;
}

.video-icon {
    display: block;
    border: 1px solid gray;
    width: 50px;
    height: 50px;
    margin-left: auto;
    margin-right: auto;
    padding: 5px;
}
```

```
#player-div {
    position: absolute;
    top: 0px;
    right: 0px;
    bottom: 160px;
    left: 30%;
    /* A bit smaller than #videos-container's right edge (compensate for border, et
c. */
    border: 1px solid gray;
}

#player-container {
    position: absolute;
    top: 0%;
    right: 0%;
    bottom: 10%;
    left: 0%;
}

#player-controls {
    position: absolute;
    bottom: 0%;
    left: 0%;
    right: 0%;
    height: 10%;
}

video {
    float: right;
    width: 100%;
    height: 100%;
}

#timeline {
    position: absolute;
    right: 200px;
    bottom: 0px;
    left: 0px;
    height: 160px;
    border: 1px solid gray;
    overflow-x: auto;
}

#tick-anchor {
    position: absolute;
    top: 50%;
    height: 0px;
    width: 100%;
    border-top: 1px solid green;
}

#export {
    position: absolute;
    right: 0px;
    bottom: 0px;
}
```

```
/*
-----
Login page
-----
*/
#splash-container {
    position: absolute;
    top: 0%;
    right: 50%;
    bottom: 0%;
    left: 0%;
    padding: 10% 10% 10% 10%;
}

#splash-image {
    float: right;
    width: 379px;
    height: 145px;
}

#login-form-container {
    position: absolute;
    top: 0%;
    right: 0%;
    bottom: 0%;
    left: 50%;
    padding: 10% 10% 10% 10%;
}

#login-form {
}

/*
-----
jQuery modal dialog
-----
*/
label,input {
    display: block;
}

input.text {
    margin-bottom: 12px;
    width: 95%;
    padding: .4em;
}

fieldset {
    padding: 0;
    border: 0;
    margin-top: 25px;
}

div#users-contain {
    width: 350px;
    margin: 20px 0;
}

div#users-contain table {
    margin: 1em 0;
    border-collapse: collapse;
    width: 100%;
}

div#users-contain table td,div#users-contain table th {
    border: 1px solid #eee;
```

```
padding: .6em 10px;
text-align: left;
}

.ui-dialog .ui-state-error {
    padding: .3em;
}

.validateTips {
    border: 1px solid transparent;
    padding: 0.3em;
}
```

```
@charset "UTF-8";

/* For HTML4 defaults, see: http://www.w3.org/TR/CSS2/sample.html */
/*
-----
HTML elements
-----
*/
body {
/* The default font-size is 16px, but this makes the math easier. For
* example, to set the size of the 'p' element to 16px, set it to 160%.
* To set the size of the 'h1' element to 32px, set it to 320%. Why not just
* set the 'p' element to 16px and the 'h1' element to 32px? Because users
* who want to resize the text on the page won't be able to because the size
* is fixed. It's better to express font-size in terms of percentages, and
* this is a neat way to do that.
*/
font-size: 100%;
line-height: 20px;
font-family: Helvetica, Arial, sans-serif;
text-align: left;
/*background-color: #edf4e6;*/
/* A lightened version of the Olive color */
}

h1,h2,h3,h4,h5,h6 {
color: #92bd6a; /* The Olive color */
font-family: Cambria, Helvetica, Arial, sans-serif;
}

h1 {
font-size: 320%; /* 32px */ /*margin: .67em 0;*/
padding: 20px;
}

h2 {
font-size: 240%; /* 24px */ /*margin: .75em 0*/
}

h3 {
font-size: 187.2%; /* 18.72px */ /*margin: .83em 0;*/
}

h4 {
font-size: 160%; /* 16px */ /*margin: 1.12em 0;*/
}

h5 {
font-size: 132.8%; /* 13.28px */ /*margin: 1.5em 0;*/
}

h6 {
font-size: 120%; /* 12px */ /*margin: 1.67em 0;*/
}

p {
font-size: 120%; /* 12px */
margin: 1.12em 0;
line-height: 1.2em;
}

table {
border-spacing: 40px 10px;
margin-left: 100px;
padding: 10px;
}

tr,td {
padding: 0px 10px 0px 10px;
}

button { /*background-color: #edf4e6;*/
/* A lightened version of the Olive color */
}

ul {
list-style-type: none;
}

strong {
font-weight: bolder;
}

small {
font-size: .83em;
}

/*
-----
General styling
-----
*/
.clear {
clear: both;
}

.center-text {
text-align: center;
}

.center-block {
margin-left: auto;
margin-right: auto;
}

.warning {
color: red;
text-decoration: underline;
cursor: pointer;
}

/*
-----
Common elements to every page
-----
*/
#header {
position: absolute;
top: 0px;
right: 0px;
left: 0px;
height: 50px;
border-bottom: 2px solid #888888;
}

#header-left {
float: left;
text-align: left;
margin-left: 1%;
}

#header-right {
float: right;
text-align: right;
}
```



```
        margin-right: 1%;
    }

    #main {
        position: absolute;
        top: 50px;
        right: 0px;
        bottom: 50px;
        left: 0px;
        min-width: 1000px;
        min-height: 450px;
    }

    #footer {
        position: absolute;
        right: 0px;
        bottom: 00px;
        left: 0px;
        height: 50px;
        border-top: 2px solid #888888;
    }

    #footer-left {
        float: left;
        text-align: left;
        margin-left: 1%;
    }

    #footer-right {
        float: right;
        text-align: right;
        margin-right: 1%;
    }
}
```

```
/*
-----
Projects page
-----
*/
#projects-container {
    position: absolute;
    top: 0%;
    right: 25%;
    bottom: 0%;
    left: 25%;
}

#projects-title {
    position: absolute;
    top: 0%;
    right: 0%;
    bottom: 15%;
    left: 0%;
}

#projects-controls {
    position: absolute;
    top: 15%;
    right: 0%;
    bottom: 20%;
    left: 0%;
}

#project-clips {
    position: absolute;
    top: 20%;
    right: 0%;
    bottom: 0%;
    left: 0%;
    overflow: auto;
}

.project-icon-container {
    float: left;
    width: 100px;
    padding-bottom: 40px;
    padding-right: 40px;
}

#project-clips div img {
    display: block;
    padding: 5px;
    border: 1px solid #000000;
    width: 100px;
    height: 100px;
    border: 1px solid #000000;
}
```

```
@charset "UTF-8";

/*
  This file resets the default browser behavior so the web page is consistent
  across all browsers.

  Modified from: http://html5boilerplate.com/
  To revert, see: http://www.w3.org/TR/CSS2/sample.html
*/ /*
HTML5 â\234° Boilerplate

style.css contains a reset, font normalization and some base styles.

credit is left where credit is due.
much inspiration was taken from these projects:
yui.yahooapis.com/2.8.1/build/base/base.css
camendesign.com/design/
praegnanz.de/weblog/htmlcssjs-kickstart
*/ /*
html5doctor.com Reset Stylesheet (Eric Meyer's Reset Reloaded + HTML5 baseline)
v1.4 9-07-27 | Authors: Eric Meyer & Richard Clark
html5doctor.com/html-5-reset-stylesheet/
*/
html,body,div,span,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,abbr,address,ci
te,code,del,dfn,em,img,ins,kbd,q,samp,small,strong,sub,sup,var,b,i,dl,dt,dd,ol,ul,l
i,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,article,aside
,canvas,details,figcaption,figure,footer,header,hgroup,menu,nav,section,summary,tim
e,mark,audio,video
{
  margin: 0;
  padding: 0;
  border: 0;
  outline: 0;
  /*font-size: ; */
  vertical-align: baseline;
  background: transparent;
}

article,aside,details,figcaption,figure,footer,header,hgroup,menu,nav,section
{
  display: block;
}

nav ul {
  list-style: none;
}

blockquote,q {
  quotes: none;
}

blockquote:before,blockquote:after,q:before,q:after {
  content: '';
  content: none;
}

a {
  margin: 0;
  padding: 0;
  font-size: ;
  vertical-align: baseline;
  background: transparent;
}

ins {
  background-color: #ff9;
  color: # text-decoration : none;
```

```
}

mark {
  background-color: #ff9;
  color: # font-style : italic;
  font-weight: bold;
}

del {
  text-decoration: line-through;
}

abbr[title],dfn[title] {
  border-bottom: 1px dotted;
  cursor: help;
}

/* tables still need cellspacing="0" in the markup */
table {
  border-collapse: collapse;
  border-spacing: 0;
}

hr {
  display: block;
  height: 1px;
  border: 0;
  border-top: 1px solid #ccc;
  margin: 1em 0;
  padding: 0;
}

input,select {
  vertical-align: middle;
```

Favicon Notes
Updated 26 October 2010
By Terry Smith

How to create Olive favicon:

1. Obtain Olive image:
<http://www.sallybernstein.com/food/single-articles/SPANISH%20OLIVES.jpg>
2. Convert 300x290 JPG to 16x16 PNG:
FastStone Photo Resizer
3. Convert PNG to ICO:
<http://www.convertico.com/>
4. Add favicon to project.
5. Add link in <head>:
<link rel="shortcut icon" href="/olive/images/olive.ico">

This is generated by:

```
getServletContext().getRealPath("/temp/");
```

in OliveServlet.java

This is the directory where our temp files are staged to be pushed to the S3 bucket from the server.

```
C:\Users\Alberto\Documents\workspace-sts-2.3.2.RELEASE\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\olive
```

Updated February 6, 2011 by Terry

These are possible POST requests (a list to keep track of them before they're implemented). They are written like Java methods, but will be implemented in JSON. Note that these exclude the usual POST requests submitted via forms. These are simply the POST requests that jQuery will call from within the JavaScript, out of the context of any form.

```
// Ignore createUser(...) because a form already does this for us.  
getUsername()
```

```
getProjects(String userName)  
createProject(String projectName, File icon)  
deleteProjects(String[] projectNames)  
renameProject(String oldProjectName, String newProjectName)
```

```
getVideos(String projectName)  
createVideo(String videoName, File icon, File videoFile)  
deleteVideos(String[] videoNames)  
renameVideo(String oldVideoName, String newVideoName)  
addToTimeline(String[] videoNames)  
removeFromTimeline(String[] videoNames)  
addToSelected(String[] videoNames)  
removeFromSelected(String[] videoNames)  
splitVideo(String[] videoName, float splitPointInSeconds)  
combineVideos(String[] videoNames) // In order
```

Example JSON format of the splitVideo(...) command:

```
{  
  "command" : "splitVideo",  
  "arguments" : {  
    "videos" : [  
      "video1",  
      "video2",  
      "video3"  
    ],  
    "splitPointInSeconds" : 5.3  
  }  
}
```

See: <http://www.jsonlint.com/>

Use:

```
response.setContentType("text/html");  
response.setContentType("application/json");
```

See: <http://stackoverflow.com/questions/2010990/how-do-you-return-a-json-object-from-a-java-servlet>

See: <http://stackoverflow.com/questions/477816/the-right-json-content-type>

02/11/11
02:18:37

Olive

1

WebContent/temp/maintainer.txt

This file exists purely to keep its parent directly under source control.

02/11/11
02:18:37

Olive

WebContent/scripts/account.js

1

```
/*
 * This is Olive's JavaScript file for account.js only.
 */

// Called once the DOM is ready but before the images, etc. load.
// Failsafe jQuery code modified from: http://api.jquery.com/jquery/#jQuery3
jQuery(function($) {

});
```



```

/*
 * This is Olive's JavaScript file for editor.jsp only.
 */

var deleteVideoDialogContext; // TODO Remove this global variable.
var video; // Global

// Called once the DOM is ready but before the images, etc. load.
// Failsafe jQuery code modified from: http://api.jquery.com/jquery/#jQuery3
jQuery(function($) {
    attachDeleteVideoHandlers();

    video = document.getElementById('player-video');

    $('#videos-playpause').click(function () {
        if (video.paused) {
            video.play();
        } else {
            video.pause();
        }
    });

    $('#videos-volume-up').click(function () {
        if (video.volume < 0.85) { // Account for rounding errors
            video.volume += 0.1;
        } else {
            video.volume = 1.0; // Don't allow rounding errors
            $('#videos-volume-up').attr('disabled', 'disabled'); // Disable
        }
        $('#videos-volume-down').removeAttr('disabled'); // Enable
    });

    $('#videos-volume-down').click(function () {
        if (video.volume > 0.15) { // Account for rounding errors
            video.volume -= 0.1;
        } else {
            video.volume = 0.0; // Don't allow rounding errors
            $('#videos-volume-down').attr('disabled', 'disabled'); // Disable
        }
        $('#videos-volume-up').removeAttr('disabled'); // Enable
    });

    // Modified from: http://jqueryui.com/demos/draggable/
    $('#video-container').draggable({
        appendTo : 'body',
        scroll : false,
        connectToSortable : '#timeline',
        helper : 'clone',
        revert : 'invalid',
        snap : '#timeline'
    });

    $('#timeline').sortable({
        revert : true,
        sort: function() {
            if($('#timeline').sortable('items').length > 0){
                $('#export-button').removeAttr('disabled');
            } else {
                $('#export-button').attr('disabled', 'disabled');
            }
        }
    });

    $('#video-container').contextMenu('videoMenu', {
        menuStyle : {
            border : '1px solid #000'
        },

```

```

        itemStyle : {
            fontFamily : 'verdana',
            backgroundColor : '#fff',
            color : 'black',
            border : 'none',
            padding : '1px'
        },
        itemHoverStyle : {
            color : '#fff',
            backgroundColor : '#00c',
            border : 'none'
        },
        bindings : {
            'split' : function (t) {
                alert('Split Video');
                console.log('Split');
            }
        }
    });
});

function attachDeleteVideoHandlers() {
    $('#delete-video').click(function () {
        $('#confirm-delete-video-dialog').dialog('open');
        deleteVideoDialogContext = this; // This is a global variable.
    });

    $('#confirm-delete-video-dialog').dialog({
        autoOpen: false,
        resizable: false,
        height: 275,
        modal: true,
        buttons: {
            'Delete': function () {
                deleteVideo.call(deleteVideoDialogContext); // We don't want the co
                ntext to be the dialog element, but rather the element that triggered it.
                $(this).dialog('close');
            },
            Cancel: function () {
                $(this).dialog('close');
            }
        }
    });
}

// Perform a deleteVideo request (to test POST commands)
function deleteVideo() {
    // Domain: http://stackoverflow.com/questions/2300771/jquery-domain-get-url
    var postUrl = location.protocol + '//' + location.host + '/olive/OliveServlet';
    var postData = '{
        +   "command" : "deleteVideo",
        +   "arguments" :
        +   [
        +       {
        +           "video" : "" + $(this).attr('id') + ""
        +       }
        +   ]
    }';

    // Encoding: http://stackoverflow.com/questions/26620/how-to-set-encoding-in-ge
    tjson-jquery
    $.ajax({
        type: 'POST',
        url: postUrl,
        contentType: 'application/json; charset=utf-8',
        data: postData,
        success: function (data) {
            location.reload(); // Reload the page.
        }
    });
}

```

```
        console.log(data);
    },
    error: function (XMLHttpRequest, textStatus, errorThrown) {
        // Don't reload on error so the message can be seen unaltered.
        console.log(XMLHttpRequest.responseText);
        alert('Video deletion not yet implemented.');
```

```
    });
}

function openNewVideoForm() {
    window.open("new-video-form.jsp", "videoUploadForm",
        "menubar=no,width=320,height=200,toolbar=no");
}
```

```

6, 64);
        bValid = bValid
            && checkLength(password,
                "register-password", 5, 128);
        bValid = bValid
            && checkRegexp(name,
                /^[a-z]([0-9a-z_])+$/i,
                "Username may consist of a-z, 0-9,
underscores; and must begin with a letter.");
        // From jquery.validate.js (by joern),
        // contributed by Scott Gonzalez:
        // http://projects.scottsplayground.com/email_address_validation/
        bValid = bValid
            && checkRegexp(
                email,
                /^((([a-z]|\d|!#$%&'*\+\/=\?^_\{|\}~)|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF])+(.[a-z]|\d|!#$%&'*\+\/=\?^_\{|\}~)|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF]))*)|((\x22)((\x20|\x09)*(\x0d\x0a)?(\x20|\x09)+)?((\[x01-[x08\x0b\x0c\x0e-\x1f\x7f]|\\x21|[x23-x5b]|\\x5d-\\x7e)|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF])|(\\((\[x01-[x09\x0b\x0c\x0d-\\x7f]|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF]))))*(\\(\x20|\x09)*(\x0d\x0a)?(\x20|\x09)+)?(\x22)))@((([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF])|((([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF])([a-z]|\d|-|\.|_|~)![\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF])*([a-z]|\d|[\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF])))).)+( ([a-z] | [\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF]) | (( [a-z] | [\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF]) * ([a-z] | [\u00A0-\uD7FF\uF900-\uFDCF\uFDF0-\uFFFF]) ) ) \. ? $ / i ,
                "Email should look like: example@example.com");
        bValid = bValid
            && checkRegexp(password,
                /^( [0-9a-zA-Z ] + ) $ / ,
                "Password field only allow : a-z 0-9");
        bValid = bValid
            && checkPasswordsEqual(password,
                cPassword,
                "Passwords are not equal")
        if (bValid) {
            $("#register-form").submit();
            $(this).dialog("close");
        }
    },
    Cancel : function() {
        window.location.reload(true);
        $(this).dialog("close");
    },
    close : function() {
        allFields.val("").removeClass("ui-state-error");
    }
});

$("#create-user").button().click(function() {
    $("#dialog-form").dialog("open");
});
});

```

```
/*
 * This is Olive's master JavaScript file for every JSP page.
 */

// Called once the DOM is ready but before the images, etc. load.
// Failsafe jQuery code modified from: http://api.jquery.com/jquery/#jQuery3
jQuery(function($) {
    includeHeader(); // TODO Fix for index.jsp (which is different)
    includeFooter();
});

function includeHeader() {
    $.get('/olive/header.jsp', function(data) {
        $('#header').append($(data));
        attachDialogToLink('help');
    });
}

function includeFooter() {
    $.get('/olive/footer.jsp', function(data) {
        $('#footer').append($(data));
        attachDialogToLink('about');
    });
}

function attachDialogToLink(linkName) {
    $('#'+ linkName + '-dialog').dialog( {
        autoOpen : false
    });

    $('#'+ linkName + '-dialog-opener').click(function() {
        $('#'+ linkName + '-dialog').dialog('open');
        return false;
    });
}

function openHelpWindow() {
    window.open("help.jsp", "HelpWindow",
        "menubar=no,width=500,height=500,toolbar=no");
}
```

```
/*
 * This is Olive's JavaScript file for projects.jsp only.
 */

var deleteProjectDialogContext; // TODO Remove this global variable.

// Called once the DOM is ready but before the images, etc. load.
// Failsafe jQuery code modified from: http://api.jquery.com/jquery/#jQuery3
jQuery(function($) {
    attachDeleteProjectHandlers();
});

function attachDeleteProjectHandlers() {
    $('#delete-project').click(function () {
        $('#confirm-delete-project-dialog').dialog('open');
        deleteProjectDialogContext = this; // This is a global variable.
    });

    $('#confirm-delete-project-dialog').dialog({
        autoOpen: false,
        resizable: false,
        height: 275,
        modal: true,
        buttons: {
            'Delete': function () {
                deleteProject.call(deleteProjectDialogContext); // We don't want th
                $(this).dialog('close');
            },
            Cancel: function () {
                $(this).dialog('close');
            }
        }
    });
}

// Perform a deleteProject request (to test POST commands)
function deleteProject() {
    // Domain: http://stackoverflow.com/questions/2300771/jquery-domain-get-url
    var postUrl = location.protocol + '//' + location.host + '/olive/OliveServlet';
    var postData = '{
        "command" : "deleteProject",
        "arguments" :
        [
            {
                "project" : "' + $(this).attr('id') + '"
            }
        ]
    }';

    // Encoding: http://stackoverflow.com/questions/26620/how-to-set-encoding-in-ge
    tjson-jquery
    $.ajax({
        type: 'POST',
        url: postUrl,
        contentType: 'application/json; charset=utf-8',
        data: postData,
        success: function (data) {
            console.log(data);
            console.log('Project deleted successfully.');
```

```
            location.reload();
        },
        error: function (XMLHttpRequest, textStatus, errorThrown) {
            // console.log(XMLHttpRequest.responseText);
            console.log(XMLHttpRequest.responseText);
            console.log('Could not delete project.');
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Olive | The Online Video Editor</title>

<link rel="shortcut icon" href="/olive/images/olive.ico" />

<link rel="stylesheet" type="text/css" href="/olive/css/reset.css" />
<link rel="stylesheet" type="text/css"
    href="/olive/scripts/jquery-ui-1.8.6.custom/css/ui-lightness/jquery-ui-1.8.6.cu
    stom.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/master.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/account.css" />

<script src="/olive/scripts/jquery-1.4.4.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.core
    .js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.widg
    et.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.mous
    e.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.butt
    on.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.posi
    tion.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.dial
    og.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drag
    gable.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drop
    pable.js"></script>
<script src="/olive/scripts/master.js"></script>
<script src="/olive/scripts/account.js"></script>
<script src="/olive/scripts/google-analytics.js"></script>
</head>
<body>
<%
    Boolean isAuthorized = (Boolean) session
        .getAttribute(Attribute.IS_AUTHORIZED.toString()); // Nasty cast
    if (isAuthorized == null) {
        response.sendRedirect("index.jsp");
    } else if (!isAuthorized) {
        response.sendRedirect("index.jsp");
    }

    Boolean editSuccessfully = (Boolean) session
        .getAttribute(Attribute.EDIT_SUCCESSFULLY.toString());
    Boolean passwordsMatch = (Boolean) session
        .getAttribute(Attribute.PASSWORDS_MATCH.toString());
    String editConfirmation;
    if (editSuccessfully == null) {
        editConfirmation = "";
    } else if (editSuccessfully) {
        editConfirmation = "Your information has been changed successfully.";
    } else {
        if (passwordsMatch == null) { // Not safe
```

```
        editConfirmation = "Invalid characters or length on one or more fields.
    ";
    } else { // Safe, but differing passwords
        editConfirmation = "Passwords do not match.";
    }
}
session.removeAttribute(Attribute.EDIT_SUCCESSFULLY.toString());
session.removeAttribute(Attribute.PASSWORDS_MATCH.toString());

String username = (String) session.getAttribute(Attribute.USERNAME
    .toString());
String name = (String) session.getAttribute(Attribute.NAME
    .toString());
String email = (String) session.getAttribute(Attribute.EMAIL
    .toString());
String password = (String) session.getAttribute(Attribute.PASSWORD
    .toString());
String securityQuestion = (String) session
    .getAttribute(Attribute.SECURITY_QUESTION.toString());
String securityAnswer = (String) session
    .getAttribute(Attribute.SECURITY_ANSWER.toString());
%>
<div id="header">
<div id="header-left">
<h1>Olive</h1>
</div>
<!-- end #header-left -->
<div id="header-right">
<div>Welcome, <%=username%>!&nbsp;<a href="logout.jsp">Logout</a></div>
<div><strong><a href="projects.jsp">My Projects</a></strong>&nbsp;<span
    id="help-dialog-opener"><a href="#">Help</a></span></div>
<div id="help-dialog" title="How to use Olive">
<ul>
    <li>1. Create a new account.</li>
    <li>2. Create a new project.</li>
    <li>3. Upload your videos.</li>
    <li>4. Edit your videos.</li>
    <li>5. Export to your computer.</li>
</ul>
</div>
</div>
<!-- end #header-right -->
<!-- end #header -->

<div class="clear"></div>

<div id="main">
<div id="edit-account-container">
<h2>Edit account information</h2>
<form id="edit-account-form" action="OliveServlet" name="process"
    method="post">
<p><label for="new-name">Name</label><br />
<input type="text" name="new-name" id="new-name" value="<%=name%>"
    size="32" maxlength="32" /></p>
<p><label for="new-email">Email</label><br />
<input type="text" name="new-email" id="new-email" value="<%=email%>"
    size="32" maxlength="64" /></p>
<p><label for="new-password">Password</label><br />
<input type="password" name="new-password" id="new-password"
    value="<%=password%>" size="32" maxlength="128" /></p>
<p><label for="confirm-new-password">Confirm password</label><br />
<input type="password" name="confirm-new-password"
    id="confirm-new-password" value="<%=password%>" size="32"
    maxlength="128" /></p>
<p><label for="security_question">Security Question</label><br />
<input type="text" name="security_question" id="security_question"
```

```
        value="<%=securityQuestion%>" size="32" maxlength="128" /></p>
<p><label for="security_answer">Security Answer</label><br />
<input type="text" name="security_answer" id="security_question_answer"
        value="<%=securityAnswer%>" size="32" maxlength="128" /></p>
<input type="hidden" name="FormName" value="EditUser"></input><br />
<input type="submit" value="Update information" /><span><%=editConfirmation%></span>
></form>
<form id="delete-account" action="OliveServlet" name="process"
        method="post">
    <input type="hidden" name="FormName" value="DeleteAccount"></input><br />
    <input type="submit" value="Delete Account" />
    <!-- TODO: Add a "Are you sure you want to delete your account window?" here -
->
    </form>
</div>
<!-- end #login-form-container --></div>
<!-- end #main -->

<div id="footer"></div>
</body>
</html>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.logic.OliveDatabaseApi"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Edit Project | Olive</title>

<link rel="shortcut icon" href="/olive/images/olive.ico">

<link rel="stylesheet" type="text/css" href="/olive/css/reset.css" />
<link rel="stylesheet" type="text/css"
    href="/olive/scripts/jquery-ui-1.8.6.custom/css/ui-lightness/jquery-ui-1.8.6.cu
stom.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/master.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/editor.css" />

<script src="/olive/scripts/jquery-1.4.4.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.core
.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.widg
et.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.mous
e.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.butt
on.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.posi
tion.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.dial
og.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drag
gable.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drop
pable.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.sort
able.js"></script>
<script src="/olive/scripts/master.js"></script>
<script src="/olive/scripts/editor.js"></script>
<script src="/olive/scripts/google-analytics.js"></script>
<script src="/olive/scripts/contextMenu.js"></script>
</head>
<body>
<%
    String username = "";
    Boolean isAuthorized = (Boolean) session
        .getAttribute(Attribute.IS_AUTHORIZED.toString()); // Nasty cast
    String projectName = "";
    String videosHtml = "";
    if (isAuthorized == null) {
        response.sendRedirect("index.jsp");
    } else if (!isAuthorized) {
        response.sendRedirect("index.jsp");
    } else {
        username = (String) session.getAttribute(Attribute.USERNAME
            .toString());
        projectName = (String) session
            .getAttribute(Attribute.PROJECT_NAME.toString());
```

```
    if (projectName == null) {
        response.sendRedirect("projects.jsp");
    }

    int accountId = OliveDatabaseApi.getAccountId(username);
    int projectId = OliveDatabaseApi.getProjectId(projectName, accountId);
    videosHtml = OliveDatabaseApi.populateVideos(projectId);
}
%>
<div id="header">
<div id="header-left">
<h1>Olive</h1>
</div>
<!-- end #header-left -->
<div id="header-right">
<div>Welcome, <a href="account.jsp"><%=username%>!</a>&nbsp;<a
    href="logout.jsp">Logout</a></div>
<div><strong><a href="projects.jsp">My Projects</a></strong>&nbsp;<span
    id="help-dialog-opener"><a href="">Help</a></span></div>
<div id="help-dialog" title="How to use Olive">
<ul>
    <li>1. Create a new account.</li>
    <li>2. Create a new project.</li>
    <li>3. Upload your videos.</li>
    <li>4. Edit your videos.</li>
    <li>5. Export to your computer.</li>
</ul>
</div>
<div id="confirm-delete-video-dialog" title="Warning!">
    <p>Delete video?</p>
</div>
</div>
<!-- end #header-right --></div>
<!-- end #header -->

<div class="clear"></div>

<div id="main">

<div id="videos-container">
<div id="videos-title">
<h3><%=projectName%></h3>
</div>
<!-- end #videos-title -->
<div id="videos-controls"><!-- http://stackoverflow.com/questions/1106720/how-to-di
splay-html-form-as-inline-element/1106747#1106747 -->

<!-- This should be refactored to be multiple forms, but then the CSS should be
    changed so the buttons all stay on the same line. -->
<form id="split-form" action="OliveServlet" name="process" method="post">
<input type="button" value="Upload New" onclick="openNewVideoForm();" />
<input type="submit" value="Split" onclick="alert('Split');" /> <input
    type="hidden" name="FormName" value="SplitVideo"></input> <input
    type="button" value="Delete" onclick="alert('Delete');" /> <input
    type="button" value="Select All" onclick="alert('Select All');" /></form>
</div>
<!-- end #videos-controls -->
<div id="videos"><%=videosHtml%></div>
<!-- end #videos --></div>
<!-- end #videos-container -->

<div class="contextMenu" id="videoMenu">
<ul>
    <li id="split">Split Video</li>
</ul>
</div>
<!-- end #contextMenu -->
```



```
<div id="player-div">
<div id="player-container"><video id="player-video"
  poster="/olive/images/bbb480.jpg" preload="preload"
  src="/olive/videos/bbb_trailer_iphone.m4v"></video></div>
<div id="player-controls" class="center-text">
<button id="videos-playpause">Play/pause</button>
<button id="videos-volume-down">Volume down</button>
<button id="videos-volume-up" disabled="disabled">Volume up</button>
</div>
</div>
<!-- end #player -->

<div class="clear"></div>

<div id="timeline">
<div id="ticks-container">
<div id="tick-anchor"></div>
</div>
</div>

<div class="clear"></div>

<div id="export">
<button id="export-button" type="button" disabled="disabled" onclick="alert('Export
');">Export to Computer</button>

</div>
<!-- end #export --></div>
<!-- end #main -->

<div class="clear"></div>

<div id="footer"></div>
</body>
</html>
```

```
<div id="footer-left"><span id="about-dialog-opener"><a
  href="">About Us</a></span></div>
<div id="about-dialog" title="About Olive">
<p>&copy; 2010 ReadyTalk</p>
</div>
<div id="footer-right">&copy; 2010 ReadyTalk</div>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Forgot Password? | The Online Video Editor</title>

<link rel="shortcut icon" href="/olive/images/olive.ico" />

<link rel="stylesheet" type="text/css" href="/olive/css/reset.css" />
<link rel="stylesheet" type="text/css"
    href="/olive/scripts/jquery-ui-1.8.6.custom/css/ui-lightness/jquery-ui-1.8.6.cu
    stom.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/master.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/account.css" />

<script src="/olive/scripts/jquery-1.4.4.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.core
    .js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.widg
    et.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.mous
    e.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.butt
    on.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.posi
    tion.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.dial
    og.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drag
    gable.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drop
    pable.js"></script>
<script src="/olive/scripts/master.js"></script>
<script src="/olive/scripts/account.js"></script>
<script src="/olive/scripts/google-analytics.js"></script>
</head>
<body>
<%
    Boolean isSafe = (Boolean) session.getAttribute(Attribute.IS_SAFE.toString());
    Boolean isCorrect = (Boolean) session.getAttribute(Attribute.IS_CORRECT.toStrin
    g());
    String editConfirmation = "";
    if(isCorrect == null){
        editConfirmation = "";
    }
    else if(isCorrect == false){
        if(isSafe == null){
            editConfirmation = "";
        }
        else if(isSafe){
            editConfirmation = "Question or answer do not match database records";
        }
    }
    else{
        editConfirmation = "Unsafe input";
    }
%>
```

```
    else if(isCorrect){
        editConfirmation = "";
        response.sendRedirect("new-password-form.jsp");
    }
    session.removeAttribute(Attribute.IS_SAFE.toString());
    session.removeAttribute(Attribute.IS_CORRECT.toString());
%>
<div id="header">
<div id="header-left">
<h1>Olive</h1>
</div>
<!-- end #header-left -->
<div id="header-right"><span
    id="help-dialog-opener"><a href="">Help</a></span></div>
<div id="help-dialog" title="How to use Olive">
<ul>
<li>1. Create a new account.</li>
<li>2. Create a new project.</li>
<li>3. Upload your videos.</li>
<li>4. Edit your videos.</li>
<li>5. Export to your computer.</li>
</ul>
</div>
<!-- end #header-right -->
<!-- end #header -->

<div class="clear"></div>

<div id="main">
<div id="edit-account-container">
<h2>Forgot Password?</h2>
<p> Please enter your username and your security question and answer. Thank you</p>
<!-- end #about-title -->

<form id="security-question-form" action="OliveServlet" name="process"
    method="post">
    <p><label for="username">Username</label><br />
    <input type="text" name="username" id="username" value=""
        size="32" maxlength="32" /></p>
    <p><label for="security_question">Security Question</label><br />
    <input type="text" name="security_question"
        id="security_question" value="" size="32"
        maxlength="128" /></p>
    <p><label for="security_answer">Security Answer</label><br />
    <input type="text" name="security_answer"
        id="security_question_answer" value="" size="32"
        maxlength="128" /></p>
    <input type="hidden" name="FormName" value="security-question-form"></input><br />
    <input type="submit" value="Recover Password" /><span><%=editConfirmation%></span>
</form>
</div>
<!-- end #main --></div>

<div id="footer"></div>
</body>
</html>
```

02/11/11
02:18:37

Olive

WebContent/header.jsp

1

```
<!-- This can't be used yet because of JSP complications (how to pass around the Java variables). -->
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Olive | The Online Video Editor</title>

<link rel="shortcut icon" href="/olive/images/olive.ico" />

<link rel="stylesheet" type="text/css" href="/olive/css/reset.css" />
<link rel="stylesheet" type="text/css"
    href="/olive/scripts/jquery-ui-1.8.6.custom/css/ui-lightness/jquery-ui-1.8.6.cu
stom.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/master.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/index.css" />

<script src="/olive/scripts/jquery-1.4.4.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.core
.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.widg
et.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.mous
e.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.butt
on.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.posi
tion.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.dial
og.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drag
gable.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drop
pable.js"></script>
<script src="/olive/scripts/master.js"></script>
<script src="/olive/scripts/index.js"></script>
<script src="/olive/scripts/google-analytics.js"></script>
</head>
<body>
<%
    Boolean isAuthorized = (Boolean) session
        .getAttribute(Attribute.IS_AUTHORIZED.toString()); // Nasty cast
    String loginMessage;
    if (isAuthorized == null) {
        loginMessage = "Please log in.";
    } else if (isAuthorized) {
        loginMessage = "You have been successfully logged in.";
    } else {
        loginMessage = "Incorrect username and/or password.";
    }
%>
<div id="header">
<div id="header-left">
<h1>Olive</h1>
</div>
<!-- end #header-left -->
<div id="header-right"><span id="help-dialog-opener"><a
    href="#">Help</a></span></div>
<div id="help-dialog" title="How to use Olive">
```

```
<ul>
    <li>1. Create a new account.</li>
    <li>2. Create a new project.</li>
    <li>3. Upload your videos.</li>
    <li>4. Edit your videos.</li>
    <li>5. Export to your computer.</li>
</ul>
</div>
<!-- end #header-right --></div>
<!-- end #header -->

<div class="clear"></div>

<div id="main">
<div id="splash-container"></div>
<!-- end #splash-container -->

<div id="login-form-container">

<form id="login-form" action="OliveServlet" name="process" method="post">
<p><label for="username">Username</label> <input type="text"
    name="username" id="login-username" size="32" maxlength="16" /></p>
<p><label for="password">Password</label> <input type="password"
    name="password" id="login-password" size="32" maxlength="128" /></p>
<input type="hidden" name="FormName" value="LoginUser"></input> <input
    type="submit" value="Login" /><br />
<span><%=loginMessage%> <a href="forgot.jsp">Forgot password?</a></span></form>
<p>Don't have an account? <a id="create-user" href="javascript:;"
    title="">Sign up!</a></p>
</div>
<!-- end #login-form-container -->

<div id="dialog-form" title="Create new user">
<p class="validateTips">All form fields are required.</p>
<form id="register-form" action="OliveServlet" name="process"
    method="post">
<fieldset><label for="name">Username</label> <input
    type="text" name="name" id="register-name"
    class="text ui-widget-content ui-corner-all" maxlength="16" /> <label
    for="email">Email</label> <input type="text" name="email"
    id="register-email" value=""
    class="text ui-widget-content ui-corner-all" maxlength="64" /> <label
    for="password">Password</label> <input type="password" name="password"
    id="register-password" value=""
    class="text ui-widget-content ui-corner-all" maxlength="128" /> <label
    for="confirm_password">Confirm Password</label> <input type="password" name="co
nfirm_password"
    id="confirm-register-password" value=""
    class="text ui-widget-content ui-corner-all" maxlength="128" /></fieldset>
<input type="hidden" name="FormName" value="AddUser"></input></form>
</div>
<!-- end #dialog-form --></div>
<!-- end #main -->

<div id="footer"></div>
</body>
</html>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Logout | Olive</title>
<script src="/olive/scripts/google-analytics.js"></script>
</head>
<body>
<%
    // Modified from: http://stackoverflow.com/questions/1104975/for-loop-to-iterate-over-enum-in-java
    for (Attribute attribute : Attribute.values()) {
        session.removeAttribute(attribute.toString());
    }
    response.sendRedirect("index.jsp");
%>
</body>
</html>
```

WebContent/new-password-form.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Forgot Password? | The Online Video Editor</title>

<link rel="shortcut icon" href="/olive/images/olive.ico" />

<link rel="stylesheet" type="text/css" href="/olive/css/reset.css" />
<link rel="stylesheet" type="text/css"
    href="/olive/scripts/jquery-ui-1.8.6.custom/css/ui-lightness/jquery-ui-1.8.6.cu
stom.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/master.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/account.css" />

<script src="/olive/scripts/jquery-1.4.4.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.core
.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.widg
et.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.mous
e.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.butt
on.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.posi
tion.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.dial
og.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drag
gable.js"></script>
<script
    src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drop
pable.js"></script>
<script src="/olive/scripts/master.js"></script>
<script src="/olive/scripts/account.js"></script>
<script src="/olive/scripts/google-analytics.js"></script>
</head>
<body>
<%
    Boolean isSafe = (Boolean) session.getAttribute(Attribute.IS_SAFE.toString());
    Boolean passwordsMatch = (Boolean) session.getAttribute(Attribute.PASSWORDS_MAT
CH.toString());
    Boolean editSuccess = (Boolean) session.getAttribute(Attribute.EDIT_SUCCESSFUL
Y.toString());
    String confirmation = "";
    String index = "'index.jsp'";
    if(editSuccess == null){
        confirmation = "";
    }
    else if(editSuccess == false){
        if(isSafe){
            if(passwordsMatch){
                confirmation = "We're sorry, there is an error in inputing to the d
atabase. Please try again";
            }
            else{
                confirmation = "Passwords do not match";
            }
        }
    }
    else{
        confirmation = "Unsafe input";
    }
    }
    else if(editSuccess){
        confirmation = "Congratulations. Your new password has been set. You"
            + "may now sign in to <a href = '"+index+"'>Olive</a>";
    }
    }

    session.removeAttribute(Attribute.IS_SAFE.toString());
    session.removeAttribute(Attribute.PASSWORDS_MATCH.toString());
    session.removeAttribute(Attribute.EDIT_SUCCESSFULLY.toString());
%>
<div id="header">
<div id="header-left">
<h1>Olive</h1>
</div>
<!-- end #header-left -->
<div id="header-right"><span id="help-dialog-opener"><a
    href="#">Help</a></span></div>
<div id="help-dialog" title="How to use Olive">
<ul>
    <li>1. Create a new account.</li>
    <li>2. Create a new project.</li>
    <li>3. Upload your videos.</li>
    <li>4. Edit your videos.</li>
    <li>5. Export to your computer.</li>
</ul>
</div>
<!-- end #header-right --></div>
<!-- end #header -->

<div class="clear"></div>

<div id="main">
<div id="edit-account-container">
<h2>New Password</h2>
<p>Please enter a new password for your account</p>
<!-- end #about-title -->

<form id="password-form" action="OliveServlet" name="process"
    method="post">
<p><label for="password">Password</label><br />
<input type="password" name="password" id="password" value="" size="32"
    maxlength="32" /></p>
<p><label for="password">Confirm Password</label><br />
<input type="password" name="confirm_password" id="confirm_password"
    value="" size="32" maxlength="128" /></p>
<input type="hidden" name="FormName" value="new_password"></input><br />
<input type="submit" value="Submit New Password" /><span><%=confirmation%></span></
form>
</div>
<!-- end #main --></div>

<div id="footer"></div>
</body>
</html>
```

```
    }
    else{
        confirmation = "Unsafe input";
    }
    }
    else if(editSuccess){
        confirmation = "Congratulations. Your new password has been set. You"
            + "may now sign in to <a href = '"+index+"'>Olive</a>";
    }
    }

    session.removeAttribute(Attribute.IS_SAFE.toString());
    session.removeAttribute(Attribute.PASSWORDS_MATCH.toString());
    session.removeAttribute(Attribute.EDIT_SUCCESSFULLY.toString());
%>
<div id="header">
<div id="header-left">
<h1>Olive</h1>
</div>
<!-- end #header-left -->
<div id="header-right"><span id="help-dialog-opener"><a
    href="#">Help</a></span></div>
<div id="help-dialog" title="How to use Olive">
<ul>
    <li>1. Create a new account.</li>
    <li>2. Create a new project.</li>
    <li>3. Upload your videos.</li>
    <li>4. Edit your videos.</li>
    <li>5. Export to your computer.</li>
</ul>
</div>
<!-- end #header-right --></div>
<!-- end #header -->

<div class="clear"></div>

<div id="main">
<div id="edit-account-container">
<h2>New Password</h2>
<p>Please enter a new password for your account</p>
<!-- end #about-title -->

<form id="password-form" action="OliveServlet" name="process"
    method="post">
<p><label for="password">Password</label><br />
<input type="password" name="password" id="password" value="" size="32"
    maxlength="32" /></p>
<p><label for="password">Confirm Password</label><br />
<input type="password" name="confirm_password" id="confirm_password"
    value="" size="32" maxlength="128" /></p>
<input type="hidden" name="FormName" value="new_password"></input><br />
<input type="submit" value="Submit New Password" /><span><%=confirmation%></span></
form>
</div>
<!-- end #main --></div>

<div id="footer"></div>
</body>
</html>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Create new project | Olive</title>
<script src="/olive/scripts/google-analytics.js"></script>
</head>
<body>
<%
    Boolean isAuthorized = (Boolean) session
        .getAttribute(Attribute.IS_AUTHORIZED.toString()); // Nasty cast
    if (isAuthorized == null) {
        response.sendRedirect("index.jsp");
    } else if (!isAuthorized) {
        response.sendRedirect("index.jsp");
    }

    Boolean isSafe = (Boolean) session.getAttribute(Attribute.IS_SAFE
        .toString()); // Nasty cast
    Boolean addedSuccessfully = (Boolean) session
        .getAttribute(Attribute.ADD_SUCCESSFULLY.toString());
    String safeMessage;
    if (isSafe == null) {
        safeMessage = "";
    } else if (isSafe) {
        if(addedSuccessfully){
            // Syntax: http://www.infimum.dk/HTML/JSwindows.html
            safeMessage = "<script> window.opener.location.reload(); window.close()
; </script>";
        }
        else{
            safeMessage = "Project already exists";
        }
    } else {
        safeMessage = "Project name must be between 1 and 32 characters; must consi
st of a-z, 0-9, underscores; and must begin with a letter.";
    }
    session.removeAttribute(Attribute.ADD_SUCCESSFULLY.toString());
    session.removeAttribute(Attribute.IS_SAFE.toString());
%>
<form id="add-project-form" action="OliveServlet" name="process"
    method="post">
<p><label for="ProjectName">Project Name</label> <input type="text"
    name="ProjectName" id="ProjectName" value="" size="32" maxlength="128" />
</p>
<p><%=safeMessage%></p>
<input type="hidden" name="FormName" value="AddProject"></input> <input
    type="submit" value="Create Project" /></form>
</body>
</html>
```



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Upload new video | Olive</title>
<script src="/olive/scripts/google-analytics.js"></script>
</head>
<body>
<%
    Boolean isAuthorized = (Boolean) session
        .getAttribute(Attribute.IS_AUTHORIZED.toString()); // Nasty cast
    if (isAuthorized == null) {
        response.sendRedirect("index.jsp");
    } else if (!isAuthorized) {
        response.sendRedirect("index.jsp");
    }
%>
<form id="UploadForm" action="OliveServlet" name="process"
    enctype="multipart/form-data" method="post"><input type="hidden"
    name="FormName" value="UploadVideo"></input> <input type="file"
    name="file" /> <input type="submit" /></form>
</body>
</html>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="com.readytalk.olive.logic.OliveDatabaseApi"%>
<%@ page import="com.readytalk.olive.util.Attribute"%>
<!doctype html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>My Projects | Olive</title>

<link rel="shortcut icon" href="/olive/images/olive.ico">

<link rel="stylesheet" type="text/css" href="/olive/css/reset.css" />
<link rel="stylesheet" type="text/css"
      href="/olive/scripts/jquery-ui-1.8.6.custom/css/ui-lightness/jquery-ui-1.8.6.cu
stom.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/master.css" />
<link rel="stylesheet" type="text/css" href="/olive/css/projects.css" />

<script src="/olive/scripts/jquery-1.4.4.js"></script>
<script
      src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.core
.js"></script>
<script
      src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.widg
et.js"></script>
<script
      src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.mous
e.js"></script>
<script
      src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.butt
on.js"></script>
<script
      src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.posi
tion.js"></script>
<script
      src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.dial
og.js"></script>
<script
      src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drag
gable.js"></script>
<script
      src="/olive/scripts/jquery-ui-1.8.6.custom/development-bundle/ui/jquery.ui.drop
pable.js"></script>
<script src="/olive/scripts/master.js"></script>
<script src="/olive/scripts/projects.js"></script>
<script src="/olive/scripts/google-analytics.js"></script>
</head>
<body>
<%
    Boolean isAuthorized = (Boolean) session
        .getAttribute(Attribute.IS_AUTHORIZED.toString()); // Nasty cast
    String username = "";
    String projectsHtml = "";
    if (isAuthorized == null) {
        response.sendRedirect("index.jsp");
    } else if (!isAuthorized) {
        response.sendRedirect("index.jsp");
    } else {
        username = (String) session.getAttribute(Attribute.USERNAME
            .toString());
        int accountId = OliveDatabaseApi.getAccountId(username);
        projectsHtml = OliveDatabaseApi.populateProjects(accountId);
    }
%>
<div id="header">
<div id="header-left">
<h1>Olive</h1>
```

```
</div>
<div id="header-right">
<div>Welcome, <a href="account.jsp"><%=username%>!</a>&nbsp;<a
      href="logout.jsp">Logout</a></div>
<div><strong><a href="projects.jsp">My Projects</a></strong>&nbsp;<span
      id="help-dialog-opener"><a href="">Help</a></span></div>
<div id="help-dialog" title="How to use Olive">
<ul>
    <li>1. Create a new account.</li>
    <li>2. Create a new project.</li>
    <li>3. Upload your videos.</li>
    <li>4. Edit your videos.</li>
    <li>5. Export to your computer.</li>
</ul>
</div>
<div id="confirm-delete-project-dialog" title="Warning!">
    <p>Delete project? This will also delete the project's videos.</p>
</div>
</div>
</div>

<div class="clear"></div>

<div id="main">
<div id="projects-container">

<div id="projects-title">
<h1>My Projects</h1>
</div>
<!-- end #projects-title -->

<div id="projects-controls">
<button type="button" onclick="openNewProjectForm()">Create New
Project</button>
</div>
<!-- end #controls -->

<div class="clear"></div>
<div id="project-clips"><%=projectsHtml%></div>
<!-- end #project-clips --></div>
<!-- end #projects-container --></div>
<!-- end #main -->

<div class="clear"></div>

<div id="footer"></div>
</body>
</html>
```

```
package com.readytalk.olive.json;

public class DeleteProjectRequest {

    // No-args constructor
    public DeleteProjectRequest() {

    }

    public String command;
    public Arguments[] arguments;

    // Must be static for Gson to work.
    // See: http://sites.google.com/site/gson/gson-user-guide#TOC-Nested-Classes-including-Inner-Class
    public static class Arguments {

        // No-args constructor
        public Arguments() {

        }

        public String project;
    }
}
```

```
package com.readytalk.olive.logic;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;

// Modified from: http://www.exampledepot.com/egs/java.net/post.html
public class HttpSenderReceiver {
    public static String sendReceive(String data, URL url) {
        String response = "";
        try {
            // Send data
            URLConnection conn = url.openConnection();
            conn.setDoOutput(true);
            OutputStreamWriter outputStreamWriter = new OutputStreamWriter(
                conn.getOutputStream());
            outputStreamWriter.write(data);
            outputStreamWriter.flush();

            // Get the response
            BufferedReader bufferedReader = new BufferedReader(
                new InputStreamReader(conn.getInputStream()));
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                response += line + System.getProperty("line.separator");
            }
            outputStreamWriter.close();
            bufferedReader.close();
        } catch (IOException e) {
            // TODO WARNING: Readers and writers may not have been closed!
            System.err.println("Error communicating with URL: "
                + url.toString());
            e.printStackTrace();
        }
        return response;
    }

    public static void split() {
        try {
            String response = sendReceive(getJson(), new URL(getZencoderUrl()));
            System.out.println(response); // Do this for now.
        } catch (MalformedURLException e) {
            System.err.println(getZencoderUrl());
            e.printStackTrace();
        }
    }

    private static String getJson() {
        // Hard-coded for now.
        String data = "{\"api_key\": \""
            + OliveDatabaseApi.getZencoderApiKey()
            + "\", \"input\": \"https://s3.amazonaws.com/test-bucket-Olive/Wildl
ife.wmv\", \"
            + \"output\": [{\"base_url\": \"https://s3.amazonaws.com/test-buck
et-Olive/\", \"
            + \"filename\": \"Wildlife0-3.wmv\", \"start_clip\": 6, \"
            + \"clip_length\": 2}]}";
        System.out.println(data); // Do this for now.
        return data;
    }

    private static String getZencoderUrl() {
        return "https://app.zencoder.com/api/jobs";
    }
}
```

```
}
}
```

```
package com.readytalk.olive.logic;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Logger;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;

import org.jets3t.service.security.AWSCredentials;

import com.readytalk.olive.model.Project;
import com.readytalk.olive.model.User;

public class OliveDatabaseApi {

    private static Logger log = Logger.getLogger(OliveDatabaseApi.class
        .getName());

    // CAUTION: closeConnection() must be called sometime after this method.
    public static Connection getDBConnection() {
        try {
            Context initCtx = new InitialContext();
            DataSource ds = (DataSource) initCtx
                .lookup("java:comp/env/jdbc/OliveData");
            return ds.getConnection();
        } catch (Exception e) {
            e.printStackTrace();
        }
        // TODO Close the connection here on error.
        return null;
    }

    public static void closeConnection(Connection c) {
        try {
            c.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static Boolean isAuthorized(String username, String password) {
        Connection conn = getDBConnection();
        try {
            Statement st = conn.createStatement();
            String s = "USE OliveData;";
            st.executeUpdate(s);
            s = "SELECT AccountId FROM Accounts WHERE Username = '" + username
                + "' AND Password = Password('" + password + "')";
            ResultSet r = st.executeQuery(s);
            if (r.first()) {
                return true;
            }
            return false;
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            closeConnection(conn);
        }
        return false; // Error!
    }

    public static boolean AddAccount(User user) {
        Connection conn = getDBConnection();
```

```
        try {
            Statement st = conn.createStatement();
            String s = "USE OliveData;";
            st.executeUpdate(s);
            s = "INSERT INTO Accounts (Username, Password, Name, Email) "
                + "VALUES ('" + user.getUsername() + "', Password('"
                + user.getPassword() + "') " + ", '" + user.getName()
                + "', '" + user.getEmail() + "')";
            st.executeUpdate(s);
            return true;
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            closeConnection(conn);
        }
        return false;
    }

    // Used for the general query: "SELECT W FROM X WHERE Y = Z;"
    public static String getUnknownValueFromTable(String unknownLabel,
        String table, String knownLabel, String knownValue) {
        Connection conn = getDBConnection();
        try {
            Statement st = conn.createStatement();
            String s = "USE OliveData;";
            st.executeUpdate(s);

            s = "SELECT " + unknownLabel + " FROM " + table + " WHERE "
                + knownLabel + " = '" + knownValue + "'";
            ResultSet r = st.executeQuery(s);
            String unknownValue = "";
            if (r.first()) {
                unknownValue = r.getString(unknownLabel);
            }
            return unknownValue;
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            closeConnection(conn);
        }
        return null;
    }

    public static int getAccountId(String username) {
        return Integer.parseInt(getUnknownValueFromTable("AccountId",
            "Accounts", "Username", username));
    }

    public static String getUsername(int accountId) {
        return getUnknownValueFromTable("Username", "Accounts", "AccountId",
            Integer.toString(accountId));
    }

    public static String getPassword(int accountId) {
        return getUnknownValueFromTable("Password", "Accounts", "AccountId",
            Integer.toString(accountId));
    }

    public static String getAccountName(int accountId) {
        return getUnknownValueFromTable("Name", "Accounts", "AccountId",
            Integer.toString(accountId));
    }

    public static String getEmail(int accountId) {
        return getUnknownValueFromTable("Email", "Accounts", "AccountId",
            Integer.toString(accountId));
    }
}
```

```

public static Boolean editAccount(User user) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);
        s = "UPDATE Accounts SET Name = '" + user.getName()
            + "' WHERE Username = '" + user.getUsername() + "';";
        st.executeUpdate(s);

        s = "UPDATE Accounts SET Password = Password('"
            + user.getPassword() + "') WHERE Username = '"
            + user.getUsername() + "';";
        st.executeUpdate(s);

        s = "UPDATE Accounts SET Email = '" + user.getEmail()
            + "' WHERE Username = '" + user.getUsername() + "';";
        st.executeUpdate(s);

        s = "UPDATE Accounts SET SecurityQuestion = '" + user.getSecurityQuesti
on()
            + "' WHERE Username = '" + user.getUsername() + "';";
        st.executeUpdate(s);

        s = "UPDATE Accounts SET SecurityAnswer = '" + user.getSecurityAnswer()
            + "' WHERE Username = '" + user.getUsername() + "';";
        st.executeUpdate(s);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
    return false;
}

public static void deleteAccount(int accountId) {
    // TODO implement

    // int projectId = getProjectId(name, accountId);
    // int videoId = getVideoId(name, projectId, accountId);

    //int projectId = -1;
    // int videoId = -1;

    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);
        ResultSet r;
        s = "SELECT ProjectId FROM Projects WHERE AccountId = " + accountId + "';";
        r = st.executeQuery(s);
        if(r.first()){
            do{
                deleteProject(r.getInt("ProjectId"));
            }while(r.next());
        }
        // TODO Broken
        // Delete all videos associated with all projects associated with the a
ccount.
        //s = "DELETE FROM Videos WHERE ProjectID = '" + projectId + "';"; // T
ODO Add error checking
        //st.executeUpdate(s);

        // Delete all projects associated with the account.
        //s = "DELETE FROM Projects WHERE AccountID = '" + accountId + "';"; //
TODO Add error checking
        //st.executeUpdate(s);

        // Delete the account itself.
        s = "DELETE FROM Accounts WHERE AccountId = '" + accountId + "';"; // T
ODO Add error checking
        st.executeUpdate(s);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
}

public static int getProjectId(String projectName, int accountId) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);
        s = "SELECT ProjectID FROM Projects WHERE Name = '" + projectName
            + "' AND AccountID = '" + accountId + "';";
        ResultSet r = st.executeQuery(s);
        int projectId = -1;
        if (r.first()) {
            projectId = Integer.parseInt(r.getString("ProjectID"));
        }
        return projectId;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
    return -1;
}

public static String populateProjects(int accountId) {
    String projects = "";
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);
        s = "SELECT * FROM Projects WHERE AccountID = '" + accountId + "';";
        ResultSet r = st.executeQuery(s);
        if (r.first()) {
            int projectNum = 0;
            do {
                projectNum += 1;
                String projectName = r.getString("Name");
                String projectIcon = "/olive/images/SPANISH OLIVES.jpg";
                projects += "<div id=\"project-"
                    + projectNum
                    + "\" class=\"project-icon-container\">"
                    + "\n"
                    + "<a href=\"OliveServlet?projectName="
                    + projectName
                    + "\"><img src=\""
                    + projectIcon
                    + "\" class=\"project-icon\" alt=\""
                    + projectName
                    + "\" /></a>"
                    + "\n"
                    + "<p><a href=\"OliveServlet?projectName="
                    + projectName
                    + "\">"

```

```

        + projectName
        + "</a></p>"
        + "\n"
        + "<p><small><a id=\"\" // TODO Assign the videoName els
ewhere for the JavaScript to access.
        + projectName
        + "\" class=\"warning delete-project\">Delete</a></smal
l></p>"
        + "\n" + "</div>" + "\n";
    } while (r.next());
    }
    return projects;
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    closeConnection(conn);
}
return projects;
// TODO change db to have unique usernames for accounts and names for
// both projects and videos in one project
}

public static boolean projectExists(String projectName, int accountId) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);

        s = "SELECT Name FROM Projects WHERE Name = '" + projectName
            + "' AND AccountID = '" + accountId + "';";
        ResultSet r = st.executeQuery(s);
        if (r.first()) {
            return true;
        }
        return false;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
    return false;
}

public static boolean AddProject(Project project) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);
        Boolean exists = projectExists(project.getName(), project.getAccountId());

        if(exists){
            return false;
        }
        else{
            s = "INSERT INTO Projects (Name, AccountID, Icon) " + "VALUES ('"
                + project.getName() + "', '" + project.getAccountId()
                + "', '" + project.getIcon() + "');";
            st.executeUpdate(s);
            return true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
}

    return false;
}

public static void deleteProject(int projectId) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);

        // Delete all videos associated with the project.
        s = "DELETE FROM Videos WHERE ProjectID = '" + projectId + "'; // TOD
O Add error checking
        st.executeUpdate(s);

        // Delete the project itself.
        s = "DELETE FROM Projects WHERE ProjectID = '" + projectId + "'; // T
ODO Add error checking
        st.executeUpdate(s);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
}

public static int getVideoId(String videoName, int projectId, int accountId) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);
        s = "SELECT VideoID FROM Videos WHERE Name = '" + videoName
            + "' AND ProjectID = '" + projectId + "' AND AccountID = '"
            + accountId + "';";
        ResultSet r = st.executeQuery(s);
        int videoId = -1;
        if (r.first()) {
            videoId = Integer.parseInt(r.getString("VideoID"));
        }
        return videoId;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
    return -1;
}

public static String populateVideos(int projectId) {
    String videos = "";
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);
        s = "SELECT * FROM Videos WHERE ProjectID = '" + projectId + "';";
        ResultSet r = st.executeQuery(s);
        if (r.first()) {
            int videoNum = 0;
            do {
                videoNum += 1;
                String videoName = r.getString("Name");
                String videoIcon = "/olive/images/olive.png";

                videos += "<span id=\"video-"
                    + videoNum

```

src/com/readytalk/olive/logic/OliveDatabaseApi.java

```

        + "\" class=\"video-container\"><img id=\"olive\"
        + videoNum
        + "\" class=\"video-icon\"\"
        + \"\n\"
        + \"src=\"\"
        + videoIcon
        + "\" alt=\"olive\"
        + videoNum
        + \"\" /><br />\"
        + \"\n\"
        + videoName
        + \"<br />\"
        + \"\n\"
        + \"<small><a id=\"\" // TODO Assign the videoName elsewh
ere for the JavaScript to access.
        + videoName
        + \"\" class=\"warning delete-video\">Delete</a></small>
</span>\"
        + \"\n\";
    } while (r.next());
    }
    return videos;
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    closeConnection(conn);
}
return videos;
// TODO change db to have unique usernames for accounts and names for
// both projects and videos in one project
}

public static void AddVideo(String name, String url, int projectId,
    String icon) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = \"USE OliveData;\";
        st.executeUpdate(s);
        s = \"INSERT INTO Videos (Name, URL, ProjectID, Icon) VALUES ('\"
            + name + \"', '\" + url + \"', '\" + projectId + \"', '\" + icon
            + \"')\";
        st.executeUpdate(s);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
}

public static void deleteVideo(int videoId) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = \"USE OliveData;\";
        st.executeUpdate(s);
        s = \"DELETE FROM Videos WHERE VideoID = '\" + videoId + \"'\"; // TODO Ad
d error checking
        st.executeUpdate(s);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
}

public static AWSCredentials getAwsCredentials() {

```

```

        String awsAccessKeyPropertyName = \"\";
        String awsSecretKeyPropertyName = \"\";
        Connection conn = getDBConnection();
        try {
            Statement st = conn.createStatement();
            String s = \"USE OliveData;\";
            st.executeUpdate(s);
            s = \"SELECT * FROM S3Credentials;\";
            ResultSet r = st.executeQuery(s);
            if (r.first()) {
                awsAccessKeyPropertyName = r.getString(\"AWSAccessKey\");
                awsSecretKeyPropertyName = r.getString(\"AWSSecretKey\");
            } else {
                log.severe(\"Cannot locate AWS credentials\");
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            closeConnection(conn);
        }
    }

    return new AWSCredentials(awsAccessKeyPropertyName,
        awsSecretKeyPropertyName);
}

public static String getZencoderApiKey() {
    String zencoderApiKey = \"\";
    Connection conn = OliveDatabaseApi.getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = \"USE OliveData;\";
        st.executeUpdate(s);
        s = \"SELECT * FROM ZencoderCredentials;\";
        ResultSet r = st.executeQuery(s);
        if (r.first()) {
            zencoderApiKey = r.getString(\"ZencoderAPIKey\");
        } else {
            log.severe(\"Cannot locate Zencoder credentials\");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
    return zencoderApiKey;
}

public static Boolean isCorrectSecurityInfo(String username,
    String securityQuestion, String securityAnswer) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = \"USE OliveData;\";
        st.executeUpdate(s);
        s = \"SELECT AccountId FROM Accounts WHERE Username = '\" + username
            + \"' AND SecurityQuestion = '\" + securityQuestion
            + \"' AND SecurityAnswer = '\" + securityAnswer+\"'\";
        ResultSet r = st.executeQuery(s);
        if (r.first()) {
            return true;
        }
        return false;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {

```



```
        closeConnection(conn);
    }
    return false; // Error!
}

public static Boolean editPassword(String username, String newPassword) {
    Connection conn = getDBConnection();
    try {
        Statement st = conn.createStatement();
        String s = "USE OliveData;";
        st.executeUpdate(s);
        s = "UPDATE Accounts SET Password = Password('"
            + newPassword + "') WHERE Username = '"
            + username + "'";
        st.executeUpdate(s);
        return true;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        closeConnection(conn);
    }
    return false;
}
```

src/com/readytalk/olive/logic/S3Uploader.java

```
package com.readytalk.olive.logic;

import java.io.File;
import java.io.IOException;
import java.security.NoSuchAlgorithmException;
import java.util.logging.Logger;

import org.jets3t.service.S3ServiceException;
import org.jets3t.service.impl.rest.httpclient.RestS3Service;
import org.jets3t.service.model.S3Object;
import org.jets3t.service.security.AWSCredentials;

import com.readytalk.olive.util.InvalidFileSizeException;

// Java code samples: https://bitbucket.org/jmurty/jets3t/src/Release-0_8_0/src/org
// jets3t/samples/CodeSamples.java
// HTML code samples: http://jets3t.s3.amazonaws.com/toolkit/code-samples.html#down
// loading
// JavaDocs: http://jets3t.s3.amazonaws.com/api/org/jets3t/service/model/StorageObj
// ect.html
public class S3Uploader {
    private static final String BUCKET_NAME = "test-bucket-Olive";
    private static final long MAX_SIZE_IN_BYTES = 31457280L; // 30 MB
    private static final long MIN_SIZE_IN_BYTES = 1L; // ~0 MB

    private static Logger log = Logger.getLogger(S3Uploader.class.getName());

    public static boolean uploadFile(File file) throws InvalidFileSizeException {
        if (file.length() > MAX_SIZE_IN_BYTES) {
            throw new InvalidFileSizeException("File larger than "
                + MAX_SIZE_IN_BYTES + " bytes");
        }
        if (file.length() < MIN_SIZE_IN_BYTES) {
            throw new InvalidFileSizeException("File smaller than "
                + MIN_SIZE_IN_BYTES + " bytes");
        }
    }

    try {
        AWSCredentials awsCredentials = OliveDatabaseApi.getAwsCredentials();

        // RestS3Service is similar to HttpClient
        RestS3Service s3Service = new RestS3Service(awsCredentials);

        // Set Content-Length automatically based on the file's extension.
        S3Object fileAsS3Object = new S3Object(file);

        // Upload the data objects.
        s3Service.putObject(BUCKET_NAME, fileAsS3Object);

        return true;
    } catch (IOException e) {
        log.severe("Error connecting with S3");
        e.printStackTrace();
    } catch (S3ServiceException e) {
        log.severe("Error creating RestS3Service object");
        e.printStackTrace();
    } catch (NoSuchAlgorithmException e) {
        log.severe("Error creating S3Object object");
        e.printStackTrace();
    }
    return false; // Error
}

public static File downloadFile(String fileName,
    RestS3Service s3Service) {
    S3Object objectComplete = null;
    try {
```

```
        objectComplete = s3Service.getObject(BUCKET_NAME, fileName); // TODO
        Does this work?
        return objectComplete.getDataInputFile();
    } catch (S3ServiceException e) {
        log.severe("Error accessing object \"" + fileName
            + "\" in S3 bucket \"" + BUCKET_NAME + "\"");
        e.printStackTrace();
    }

    log.severe("Returning null!");
    return null;
}
}
```



```
package com.readytalk.olive.model;

public class Project {

    private String name;
    private int accountId;
    private String icon;

    public Project(String name, int accountId, String icon) {
        this.name = name;
        this.accountId = accountId;
        this.icon = icon;
    }

    @Override
    public String toString() {
        return name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAccountId() {
        return accountId;
    }

    public void setAccountId(int accountId) {
        this.accountId = accountId;
    }

    public String getIcon() {
        return icon;
    }

    public void setIcon(String icon) {
        this.icon = icon;
    }
}
```

```
package com.readytalk.olive.model;

public class User {

    private String username;
    private String password;
    private String name;
    private String email;
    private String securityQuestion;
    private String securityAnswer;

    public User(String username, String password, String name, String email) {
        this.username = username;
        this.password = password;
        this.name = name;
        this.email = email;
    }

    public User(String username, String password, String name,
        String email, String securityQuestion, String securityAnswer) {
        this.username = username;
        this.password = password;
        this.name = name;
        this.email = email;
        this.setSecurityQuestion(securityQuestion);
        this.setSecurityAnswer(securityAnswer);
    }

    @Override
    public String toString() {
        return username;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getEmail() {
        return email;
    }

    public void setSecurityQuestion(String securityQuestion) {
```

```
        this.securityQuestion = securityQuestion;
    }

    public String getSecurityQuestion() {
        return securityQuestion;
    }

    public void setSecurityAnswer(String securityAnswer) {
        this.securityAnswer = securityAnswer;
    }

    public String getSecurityAnswer() {
        return securityAnswer;
    }
}
```

```
package com.readytalk.olive.model;

public class Video {

    private String name;
    private String url;
    private String icon;
    private int projectId;
    private int startTimeStoryboard;

    public Video(String name, String url, String icon, int projectId,
        int startTimeStoryboard) {
        this.name = name;
        this.url = url;
        this.icon = icon;
        this.projectId = projectId;
        this.setStartTimeStoryboard(startTimeStoryboard);
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getUrl() {
        return url;
    }

    public void setUrl(String url) {
        this.url = url;
    }

    public String getIcon() {
        return icon;
    }

    public void setIcon(String icon) {
        this.icon = icon;
    }

    public int getProjectId() {
        return projectId;
    }

    public void setProjectId(int project) {
        this.projectId = project;
    }

    public void setStartTimeStoryboard(int startTimeStoryboard) {
        this.startTimeStoryboard = startTimeStoryboard;
    }

    public int getStartTimeStoryboard() {
        return startTimeStoryboard;
    }

}
```

src/com/readytalk/olive/servlet/OliveServlet.java

```

package com.readytalk.olive.servlet;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.util.Iterator;
import java.util.List;
import java.util.logging.Logger;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileUploadException;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

import com.google.gson.Gson;
import com.readytalk.olive.json.DeleteProjectRequest;
import com.readytalk.olive.logic.HttpSenderReceiver;
import com.readytalk.olive.logic.OliveDatabaseApi;
import com.readytalk.olive.logic.S3Uploader;
import com.readytalk.olive.logic.Security;
import com.readytalk.olive.model.Project;
import com.readytalk.olive.model.User;
import com.readytalk.olive.util.Attribute;
import com.readytalk.olive.util.InvalidFileSizeException;

public class OliveServlet extends HttpServlet {
    // Don't store anything as a member variable in the Servlet.
    // private Object dontDoThis;

    // Generated using Eclipse's "Add generated serial version ID" refactoring.
    private static final long serialVersionUID = -6820792513104430238L;
    // Static variables are okay, though, because they don't change across instance
    s.
    private static Logger log = Logger.getLogger(OliveServlet.class.getName());
    private static final String TMP_DIR_PATH = "/temp/";
    private static File tmpDir;
    private static final String DESTINATION_DIR_PATH = "/temp/";
    private static File destinationDir;

    // Modified from: http://www.jsptube.com/servlet-tutorials/servlet-file-upload-example.html
    // Also see: http://stackoverflow.com/questions/4101960/storing-image-using-htm
    -input-type-file
    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        String realPathTmp = getServletContext().getRealPath(TMP_DIR_PATH);
        tmpDir = new File(realPathTmp);
        // tmpDir = new File(TMP_DIR_PATH);
        log.info(realPathTmp);
        if (!tmpDir.isDirectory()) {
            throw new ServletException(TMP_DIR_PATH + " is not a directory");
        }
        String realPathDest = getServletContext().getRealPath(
            DESTINATION_DIR_PATH);
        destinationDir = new File(realPathDest);
        // destinationDir = new File(DESTINATION_DIR_PATH);
        log.info(realPathDest);
        if (!destinationDir.isDirectory()) {

            throw new ServletException(DESTINATION_DIR_PATH
                + " is not a directory");
        }
    }

    @Override
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (request.getContentType().contains(
            "application/x-www-form-urlencoded")) { // Full value: "application
            /x-www-form-urlencoded"
            // This is a regular text form.
            String id = request.getParameter("FormName");
            log.info("The servlet is responding to an "
                + "HTTP POST request from form: " + id);
            if (id.equals("LoginUser")) {
                handleLogin(request, response, session);
            } else if (id.equals("EditUser")) {
                handleEditUser(request, response, session);
            } else if (id.equals("AddUser")) {
                handleAddUser(request, response, session);
            } else if (id.equals("AddProject")) {
                handleAddProject(request, response, session);
            } else if (id.equals("SplitVideo")) {
                handleSplitVideo(request, response, session);
            } else if (id.equals("security-question-form")) {
                handleSecurityQuestion(request, response, session);
            } else if (id.equals("new_password")) {
                handleNewPassword(request, response, session);
            } else if (id.equals("DeleteAccount")) {
                handleDeleteAccount(request, response, session);
            } else {
                log.severe("HTTP POST request coming from unknown form: " + id);
            }
        } else if (request.getContentType().contains("multipart/form-data")) { // Full
            value: "multipart/form-data; boundary=----WebKitFormBoundaryjAGjLWGWei3ltfBe"
            // This is a file upload form.
            log.info("The servlet is responding to an "
                + "HTTP POST request from a file upload form");
            handleUploadVideo(request, response, session);
        } else if (request.getContentType().contains("application/json")) {
            // This is not a form, but a custom POST request with JSON in it.
            log.info("responding to a custom POST");
            log.info("The servlet is responding to an "
                + "HTTP POST request in JSON format");
            handleJsonPostRequest(request, response, session);
        } else {
            log.severe("Unknown content type");
        }
    }

    private void handleDeleteAccount(HttpServletRequest request,
        HttpServletResponse response, HttpSession session)
        throws UnsupportedEncodingException, IOException {
        String username = (String)session.getAttribute(Attribute.USERNAME.toString());

        int accountId = OliveDatabaseApi.getAccountId(username);
        OliveDatabaseApi.deleteAccount(accountId);
        response.sendRedirect("logout.jsp");
    }

    private void handleNewPassword(HttpServletRequest request,
        HttpServletResponse response, HttpSession session)
        throws UnsupportedEncodingException, IOException {
        // TODO Auto-generated method stub
        String newPassword = request.getParameter("password");

```

src/com/readytalk/olive/servlet/OliveServlet.java

```

String confirmNewPassword = request.getParameter("confirm_password");
Boolean newPasswordSet;
if (Security.isSafePassword(newPassword)
    && Security.isSafePassword(confirmNewPassword)) {
    session.setAttribute(Attribute.IS_SAFE.toString(), true);
    if (newPassword.equals(confirmNewPassword)) {
        session.setAttribute(Attribute.PASSWORDS_MATCH.toString(), true);
        String username = (String) session.getAttribute(Attribute.USERNAME.to
oString());
        newPasswordSet = OliveDatabaseApi.editPassword(username, newPassword
);
        session.setAttribute(Attribute.EDIT_SUCCESSFULLY.toString(), newPas
swordSet);
    } else {
        session.setAttribute(Attribute.PASSWORDS_MATCH.toString(), false);
        session.setAttribute(Attribute.EDIT_SUCCESSFULLY.toString(), false)
;
    }
} else {
    session.setAttribute(Attribute.IS_SAFE.toString(), false);
    session.setAttribute(Attribute.EDIT_SUCCESSFULLY.toString(), false);
}
response.sendRedirect("new-password-form.jsp");
session.removeAttribute(Attribute.USERNAME.toString());
}

private void handleSecurityQuestion(HttpServletRequest request,
    HttpServletResponse response, HttpSession session)
    throws UnsupportedEncodingException, IOException {
    // TODO Auto-generated method stub
    String username = request.getParameter("username");
    String securityQuestion = request.getParameter("security_question");
    String securityAnswer = request.getParameter("security_answer");
    Boolean isCorrect;
    if (Security.isSafeUsername(username)
        && Security.isSafeSecurityQuestion(securityQuestion)
        && Security.isSafeSecurityAnswer(securityAnswer)) {
        session.setAttribute(Attribute.IS_SAFE.toString(), true);
        isCorrect = OliveDatabaseApi.isCorrectSecurityInfo(username, securityQue
stion, securityAnswer);
        session.setAttribute(Attribute.IS_CORRECT.toString(), isCorrect);
        if (isCorrect) {
            session.setAttribute(Attribute.USERNAME.toString(), username);
            response.sendRedirect("new-password-form.jsp");
            session.removeAttribute(Attribute.IS_SAFE.toString()); // Cleared s
o as to not interfere with any other form.
        } else {
            response.sendRedirect("forgot.jsp");
        }
    } else {
        session.setAttribute(Attribute.IS_SAFE.toString(), false);
        session.setAttribute(Attribute.IS_CORRECT.toString(), false);
        response.sendRedirect("forgot.jsp");
    }
}

// http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-Form-Dat
a.html
@Override
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    log.info("The servlet is responding to an HTTP GET request");
    response.setContentType("text/html");
    HttpSession session = request.getSession();

    String projectName = request.getParameter("projectName");
    int accountId = OliveDatabaseApi.getAccountId((String) session
        .getAttribute(Attribute.USERNAME.toString()));
    if (projectName != null && Security.isSafeProjectName(projectName)
        && OliveDatabaseApi.projectExists(projectName, accountId)) { // Sho
rt-circuiting
        session.setAttribute(Attribute.PROJECT_NAME.toString(), projectName);
        response.sendRedirect("editor.jsp");
    } else {
        response.sendRedirect("projects.jsp");
    }
    PrintWriter out = response.getWriter();
    out.println("File uploaded. Please close this window and refresh the editor
page.");
    out.close();
}

private void handleLogin(HttpServletRequest request,
    HttpServletResponse response, HttpSession session)
    throws UnsupportedEncodingException, IOException {
    Boolean isAuthorized;
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    if (Security.isSafeUsername(username)
        && Security.isSafePassword(password)) {
        session.setAttribute(Attribute.IS_SAFE.toString(), true);
        isAuthorized = OliveDatabaseApi.isAuthorized(username, password);
        session.setAttribute(Attribute.IS_AUTHORIZED.toString(),
            isAuthorized);
        if (isAuthorized) { // Take the user to the projects page.
            int accountId = OliveDatabaseApi.getAccountId(username);
            session.setAttribute(Attribute.USERNAME.toString(),
                OliveDatabaseApi.getUsername(accountId));
            session.setAttribute(Attribute.PASSWORD.toString(), password);
            session.setAttribute(Attribute.EMAIL.toString(),
                OliveDatabaseApi.getEmail(accountId));
            session.setAttribute(Attribute.NAME.toString(),
                OliveDatabaseApi.getAccountName(accountId));
            session.removeAttribute(Attribute.IS_SAFE.toString()); // Cleared s
o as to not interfere with any other form.
            response.sendRedirect("projects.jsp");
        } else {
            response.sendRedirect("index.jsp"); // Keep the user on the same pa
ge.
        }
    } else {
        session.setAttribute(Attribute.IS_SAFE.toString(), false);
        session.setAttribute(Attribute.IS_AUTHORIZED.toString(), false);
        response.sendRedirect("index.jsp");
    }
}

private void handleEditUser(HttpServletRequest request,
    HttpServletResponse response, HttpSession session)
    throws UnsupportedEncodingException, IOException {
    String username = (String) session.getAttribute(Attribute.USERNAME
.toString());
    String newName = request.getParameter("new-name");
    String newEmail = request.getParameter("new-email");
    String newPassword = request.getParameter("new-password");
    String confirmNewPassword = request
        .getParameter("confirm-new-password");
    String securityQuestion = request.getParameter("security_question");
    String securityAnswer = request.getParameter("security_answer");
    log.info("Security question: "+securityQuestion+". Security Answer: "+secur
ityAnswer);
    if (Security.isSafeName(newName) && Security.isSafeEmail(newEmail)

```



```

        && Security.isSafePassword(newPassword)
        && Security.isSafePassword(confirmNewPassword)
        && Security.isSafeSecurityQuestion(securityQuestion)
        && Security.isSafeSecurityAnswer(securityAnswer)) {
    if (newPassword.equals(confirmNewPassword)) {
        User updateUser = new User(username, newPassword, newName,
            newEmail, securityQuestion, securityAnswer);
        Boolean editSuccessfully = OliveDatabaseApi
            .editAccount(updateUser);
        session.setAttribute(Attribute.EDIT_SUCCESSFULLY.toString(),
            editSuccessfully);
        session.setAttribute(Attribute.PASSWORDS_MATCH.toString(), true);
        session.setAttribute(Attribute.PASSWORD.toString(), newPassword);
        session.setAttribute(Attribute.EMAIL.toString(), newEmail);
        session.setAttribute(Attribute.NAME.toString(), newName);
        session.setAttribute(Attribute.SECURITY_QUESTION.toString(), securityQuestion);
        session.setAttribute(Attribute.SECURITY_ANSWER.toString(), securityAnswer);
    } else {
        session.setAttribute(Attribute.EDIT_SUCCESSFULLY.toString(),
            false);
        session.setAttribute(Attribute.PASSWORDS_MATCH.toString(),
            false);
    }
} else {
    session.setAttribute(Attribute.EDIT_SUCCESSFULLY.toString(), false);
}
response.sendRedirect("account.jsp");
}

private void handleAddUser(HttpServletRequest request,
    HttpServletResponse response, HttpSession session)
    throws IOException {
    // The jQuery regex should catch malicious input, but sanitize just to
    // be safe.
    String username = Security.stripOutIllegalCharacters(request
        .getParameter("name"));
    String password = Security.stripOutIllegalCharacters(request
        .getParameter("password"));
    String email = Security.stripOutIllegalCharacters(request
        .getParameter("email"));
    User newUser = new User(username, password, "", email);
    Boolean addSuccessfully = OliveDatabaseApi.AddAccount(newUser);
    if (addSuccessfully) {
        session.setAttribute(Attribute.IS_AUTHORIZED.toString(), true);
        session.setAttribute(Attribute.USERNAME.toString(), username);
        session.setAttribute(Attribute.PASSWORD.toString(), password);
        session.setAttribute(Attribute.EMAIL.toString(), email);
        response.sendRedirect("projects.jsp");
    } else {
        response.sendRedirect("index.jsp");
        // TODO Add error message here
    }
}

private void handleAddProject(HttpServletRequest request,
    HttpServletResponse response, HttpSession session)
    throws UnsupportedEncodingException, IOException {
    String projectName = request.getParameter("ProjectName");
    if (Security.isSafeProjectName(projectName)) {
        session.setAttribute(Attribute.IS_SAFE.toString(), true);

        String sessionUsername = (String) session
            .getAttribute(Attribute.USERNAME.toString());
        int accountId = OliveDatabaseApi.getAccountId(sessionUsername);
        String icon = ""; // TODO Get this from user input.

```

```

        Project project = new Project(projectName, accountId, icon);
        Boolean added = OliveDatabaseApi.AddProject(project);
        if (!added) {
            session.setAttribute(Attribute.ADD_SUCCESSFULLY.toString(), false);
        } else {
            session.setAttribute(Attribute.ADD_SUCCESSFULLY.toString(), true);
        }
    } else {
        session.setAttribute(Attribute.IS_SAFE.toString(), false);
    }
    response.sendRedirect("new-project-form.jsp");
}

private static void handleUploadVideo(HttpServletRequest request,
    HttpServletResponse response, HttpSession session)
    throws IOException {
    PrintWriter out = response.getWriter();
    out.println("Uploading file...");

    response.setContentType("text/plain");

    DiskFileItemFactory fileItemFactory = new DiskFileItemFactory();
    // Set the size threshold, above which content will be stored on disk.
    fileItemFactory.setSizeThreshold(1 * 1024 * 1024); // 1 MB

    // Set the temporary directory to store the uploaded files of size above threshold.
    fileItemFactory.setRepository(tmpDir);

    ServletFileUpload uploadHandler = new ServletFileUpload(fileItemFactory);
    try {
        /*
         * Parse the request
         */
        List items = uploadHandler.parseRequest(request);
        Iterator itr = items.iterator();
        while (itr.hasNext()) {
            FileItem item = (FileItem) itr.next();
            /*
             * Handle Form Fields.
             */
            if (item.isFormField()) {
                log.info("Form Name = \"" + item.getFieldName()
                    + "\", Value = \"" + item.getString() + "\"");
            } else {
                // Handle Uploaded files.
                log.info("Field Name = \"" + item.getFieldName()
                    + "\", File Name = \"" + item.getName()
                    + "\", Content type = \"" + item.getContentType()
                    + "\", File Size (bytes) = \"" + item.getSize()
                    + "\"");
                /*
                 * Write file to the ultimate location.
                 */
                File file = new File(destinationDir, item.getName()); // Allocate the space
                item.write(file); // Save the file to the allocated space

                String sessionUsername = (String) session
                    .getAttribute(Attribute.USERNAME.toString());
                int accountId = OliveDatabaseApi
                    .getAccountId(sessionUsername);
                String projectName = (String) session
                    .getAttribute(Attribute.PROJECT_NAME.toString());
                int projectId = OliveDatabaseApi.getProjectId(projectName,
                    accountId);

```

src/com/readytalk/olive/servlet/OliveServlet.java

```

String videoName = file.getName().split("[.]"[0]; // Strip ext
ensions
    if (Security.isSafeVideoName(videoName)
        && S3Uploader.uploadFile(file)) { // Short-circuiting f
or efficiency
        String icon = ""; // TODO Obtain this from S3.
        OliveDatabaseApi.AddVideo(videoName, "http", projectId,
            icon);
        }
        file.delete();
    }
}
out.println("File uploaded. Please close this window and refresh the ed
itor page.");
out.println();
} catch (FileUploadException e) {
    log.severe("Error encountered while parsing the request in the upload h
andler");
    out.println("Upload failed.");
    e.printStackTrace();
} catch (InvalidFileSizeException e) {
    log.severe("Invalid file size");
    out.println("Upload failed (invalid file size)");
    e.printStackTrace();
} catch (Exception e) {
    log.severe("Unknown error encountered while uploading file");
    out.println("Upload failed (unknown reason).");
    e.printStackTrace();
} finally {
    out.close();
}
}

private void handleSplitVideo(HttpServletRequest request,
    HttpServletResponse response, HttpSession session)
    throws IOException {
    HttpSenderReceiver.split();
    response.sendRedirect("editor.jsp");
}

// Gson help: http://code.google.com/p/google-gson/
// http://stackoverflow.com/questions/338586/a-better-java-json-library
// http://stackoverflow.com/questions/1688099/convertng-json-to-java/1688182#1
688182
private void handleJsonPostRequest(HttpServletRequest request,
    HttpServletResponse response, HttpSession session)
    throws IOException {

    // TODO Make this more general (not just a deleteProjectRequest).
    DeleteProjectRequest deleteProjectRequest = new Gson().fromJson(
        request.getReader(), DeleteProjectRequest.class);

    if (!deleteProjectRequest.command.equals("deleteProject")) {
        response.sendError(HttpServletResponse.SC_BAD_REQUEST);
        return;
    }

    response.setContentType("text/plain");
    // response.setStatus(HttpServletResponse.SC_OK); // Unnecessary

    PrintWriter out = response.getWriter();

    out.println("{\"command\":\"" + deleteProjectRequest.command
        + "\",\"arguments\":{\"project\":\""
        + deleteProjectRequest.arguments[0].project + "\"}}");

```

```

String sessionUsername = (String) session
    .getAttribute(Attribute.USERNAME.toString());
int accountId = OliveDatabaseApi.getAccountId(sessionUsername);
String firstProjectToDelete = deleteProjectRequest.arguments[0].project;
int projectId = OliveDatabaseApi.getProjectId(firstProjectToDelete,
    accountId);
OliveDatabaseApi.deleteProject(projectId);

out.println(deleteProjectRequest.arguments[0].project
    + " deleted successfully.");
out.close();
}
}

```

02/11/11
02:18:37

Olive

src/com/readytalk/olive/util/Attribute.java

1

```
package com.readytalk.olive.util;

public enum Attribute {
    IS_AUTHORIZED, USERNAME, PASSWORD, EDIT_SUCCESSFULLY, ADD_SUCCESSFULLY, PROJECT
_NAME, PASSWORDS_MATCH, IS_SAFE, EMAIL, NAME, SECURITY_QUESTION, SECURITY_ANSWER, I
S_CORRECT
}
```

```
package com.readytalk.olive.util;

public class InvalidFileSizeException extends Exception {

    // Created using Eclipse's "Add generated serial version ID" refactoring.
    private static final long serialVersionUID = 1699131254533294330L;

    public InvalidFileSizeException() {
        // TODO Auto-generated constructor stub
    }

    public InvalidFileSizeException(String message) {
        super(message);
        // TODO Auto-generated constructor stub
    }

    public InvalidFileSizeException(Throwable cause) {
        super(cause);
        // TODO Auto-generated constructor stub
    }

    public InvalidFileSizeException(String message, Throwable cause) {
        super(message, cause);
        // TODO Auto-generated constructor stub
    }
}
```