CKEditor And SimpleUploads Configuration In Spring Web Application

1,What is CKEditor?

CKEditor is a ready-for-use HTML text editor designed to simplify web content creation. It's a WYSIWYG editor that brings common word processor features directly to your web pages. Enhance your website experience with our community maintained editor.

JSP Example

To use CKEditor in a Java web application, we need to add following dependency:

```
<!--HTML editor-->
<dependency>
    <groupId>com.ckeditor</groupId>
    <artifactId>ckeditor-java-core</artifactId>
    <version>3.5.3</version>
</dependency>
```

After adding this dependency to our project, we can use it in a JSP file as follows:

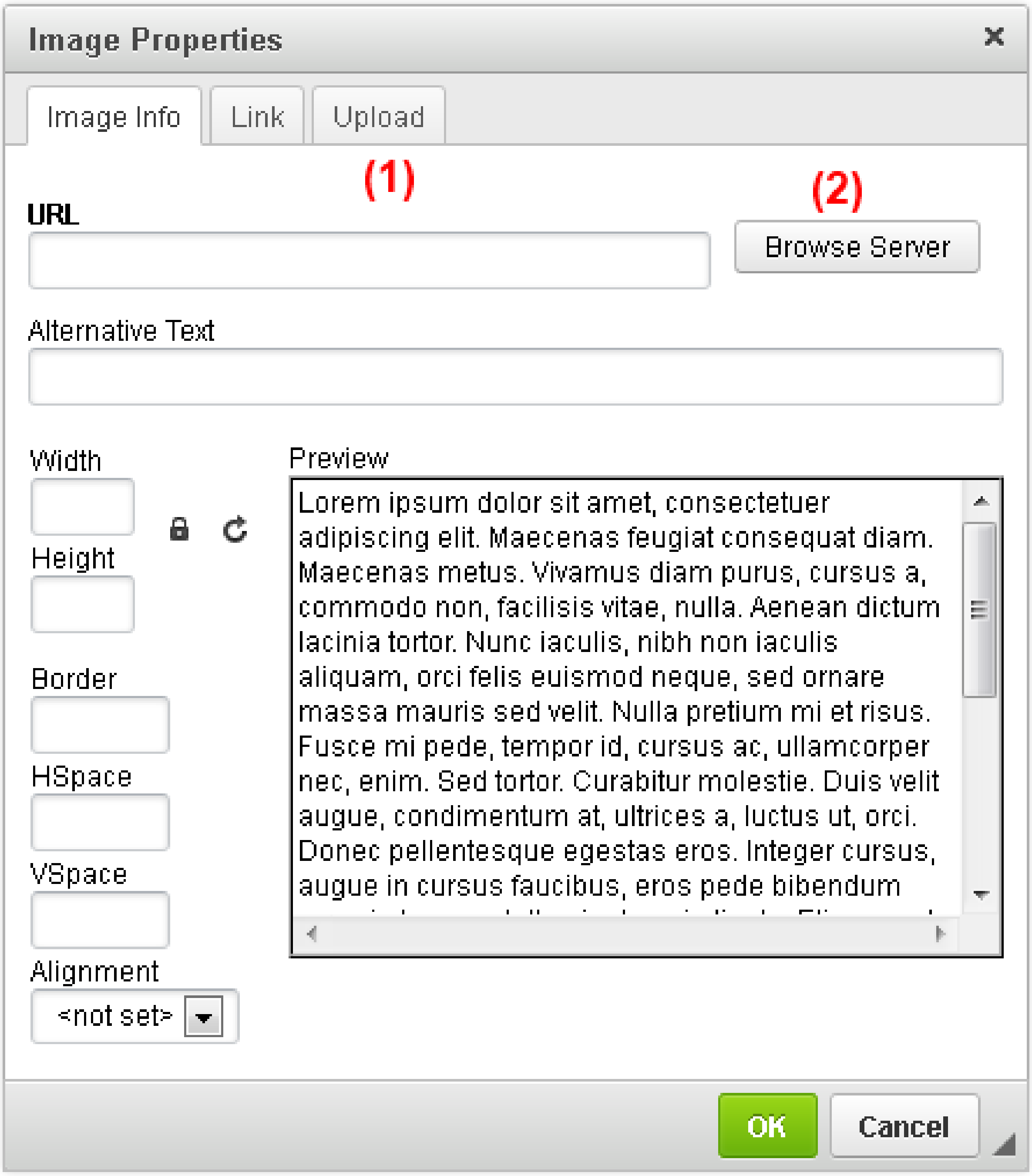
```
1   <div class="admin admin-add-problem">
2       <form:form method="post" commandName="problem">
3           <fieldset>
4               <legend>CKEditor Sample</legend>
5               <label for="editor">Write Content</label>
6               <form:textarea id="editor" path="content"/>
7               <button type="submit" class="buttonSubmit">Save</button>
8           </fieldset>
9       </form:form>
10  </div>
11  <ckeditor:replace replace="editor" basePath="/resources/ckeditor/"/>
```

In line 11, we specified base path of the editor. This means that CKEditor files located in `webapp/resources/ckeditor`.

File Uploads

CKEditor can be easily integrated with an external file browser/uploader.

Once properly set up, all file browser features will automatically become available. This includes the Upload tab (1) in the Link, Image, and Flash Properties dialog windows as well as the Browse Server button



Alignment <not set>

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas feugiat consequat diam. Maecenas metus. Vivamus diam purus, cursus a, commodo non, facilisis vitae, nulla. Aenean dictum lacinia tortor. Nunc iaculis, nibh non iaculis aliquam, orci felis euismod neque, sed ornare massa mauris sed velit. Nulla pretium mi et risus. Fusce mi pede, tempor id, cursus ac, ullamcorper nec, enim. Sed tortor. Curabitur molestie. Duis velit augue, condimentum at, ultrices a, luctus ut, orci. Donec pellentesque egestas eros. Integer cursus, augue in cursus faucibus, eros pede bibendum

Basic Configuration

- The filebrowserBrowseUrl setting contains the location of an external file browser that should be launched when the Browse Server button is pressed.
-
- The filebrowserUploadUrl setting contains the location of a script that handles file uploads. If set, the Upload tab will appear in some dialog windows — the ones where such functionality is available, i.e. Link, Image and Flash Properties.
-

Example 1 — Adding File Browser Scripts

The sample below shows basic configuration code that can be used to insert a CKEditor instance with the file browser configured.

```
CKEDITOR.editorConfig= function (config) {
      // Define changes to default configuration here.
      config.startupFocus= false;
      if(isIE()){
            if (!window.location.origin) {
                  window.location.origin= window.location.protocol + "//" + window.location.hostname
+ (window.location.port ? ':' + window.location.port: '');
            }
            config.filebrowserBrowseUrl= '/admin/home/uploadClassic';
      }else{
            config.filebrowserBrowseUrl= '/admin/home/upload';
      }
      config.filebrowserWindowWidth= '10';
      config.filebrowserWindowHeight= '10';
      config.selectMultiple= true;
};
function isIE() {
      var ua = window.navigator.userAgent;
      var msie = ua.indexOf("MSIE ");
      return !!(msie > 0 || !!navigator.userAgent.match(/Trident.*rv\:11\./));
}
```

Note: If you are using SimpleUploads plugin, your config should be as follows:

```
CKEDITOR.editorConfig= function (config) {
        // Define changes to default configuration here
        config.extraPlugins= 'simpleuploads';
        config.startupFocus= false;
        if(isIE()){
            config.filebrowserBrowseUrl= '/admin/home/uploadClassic';
        }else{
            config.filebrowserBrowseUrl= '/admin/home/upload';
        }
        config.filebrowserImageUploadUrl= '/admin/home/upload-drag-drop';
        config.filebrowserUploadUrl= '/admin/home/upload-drag-drop';
        config.filebrowserWindowWidth= '10';
        config.filebrowserWindowHeight= '10';
        config.selectMultiple= true;

};
function isIE() {
        var ua = window.navigator.userAgent;
        var msie = ua.indexOf("MSIE ");
        return !!(msie > 0 || !!navigator.userAgent.match(/Trident.*rv\:11\./));
}
```

JSP Pages

We will create three JSP pages to allow single, multiple and ajax-based file uploads.

upload.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8" %>
<%@ page session="false" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<!DOCTYPE html>
<html>
<head>
        <link href="<c:url value="/resources/css/main.css" />" rel="stylesheet">
    <title>File Upload</title>
    <script type="text/javascript">
        $(document).ready(function () {
            $('progress').hide();

            var _href = $("#oldUploadLink").attr("href");
            $('#oldUploadLink').attr("href", _href + "?CKEditor=" + getUrlParameter("CKEditor"));
            $('.trigger-file-input').click(function () {
                $('#file').click();
            });
            $("input:file").change(function () {

                $('progress').show();

                var files = document.getElementById('file').files;

                var formData = new FormData();
                formData.append("${_csrf.parameterName}", "${_csrf.token}");
                formData.append("CKEditor", getUrlParameter("CKEditor"));
                formData.append("CKEditorFuncNum", getUrlParameter("CKEditorFuncNum"));
                formData.append("langCode", getUrlParameter("langCode"));

                for (var i = 0; i < files.length; i++) {
                    formData.append("file" + i, files[i]);

                }
                $.ajax({
                    url: "/admin/home/upload-ajax", //Server script to process data
                    type: 'POST',
                    xhr: function () { // Custom XMLHttpRequest
                        var myXhr = $.ajaxSettings.xhr();
                        if (myXhr.upload) {
                            myXhr.upload.addEventListener('progress', progressHandlingFunction,
false);
                        }
                        return myXhr;
                    },
                    //Ajax events
                    beforeSend: beforeSendHandler,

                    error: errorHandler,
                    // Form data
                    data: formData,
                    //Options to tell jQuery not to process data or worry about content-type.
                    cache: false,
                    contentType: false,
                    processData: false,
                    success: function (e) {
                        $("#result").html(e);
                    }

                });
            });

            function beforeSendHandler() {

            }

            function errorHandler(e) {
                alert("Bir hata meydana geldi. Lütfen tekrar deneyiniz");
            }

            function getUrlParameter(sParam) {
                var sPageURL = window.location.search.substring(1);
                var sURLVariables = sPageURL.split('&');
                for (var i = 0; i < sURLVariables.length; i++) {
                    var sParameterName = sURLVariables[i].split('=');
                    if (sParameterName[0] == sParam) {
                        return sParameterName[1];
                    }
                }
            }

            function progressHandlingFunction(e) {
                if (e.lengthComputable) {
                    $('progress').attr({value: e.loaded, max: e.total});
                }
            }
        });
    </script>
    <style type="text/css">
        ul li {
            list-style: none;
        }
    </style>
</head>
<body>

<div class=".b-modern-upload">
    <span>Choose files to be uploaded
        If you encounter problems please <a id="oldUploadLink" href="<c:url
value="/admin/home/uploadClassic"/>">click</a></span>

    <ul>
        <li>
            <div>
                <button type="button" class="buttonSubmit trigger-file-input">DOSYA SEÇ</button>
            </div>
        </li>
        <li>
            <input type="file" id="file" multiple name="file[]">
        </li>
        <li>
            <progress></progress>
        </li>
    </ul>
    <div id="result">
    </div>
</div>
</body>
</html>
```

uploadClassic.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="false" %>
<%@page pageEncoding="UTF-8" contentType="text/html; charset=UTF-8" %>
<html>
<head>
    <title>File Upload</title>
    <link href="<c:url value="/resources/css/main.css" />" rel="stylesheet">
        <script type="text/javascript">
        $(document).ready(function () {
            var _href = $("#formUploadLink").attr("action");
            $('#formUploadLink').attr("action", _href + "?CKEditor=" + getUrlParameter("CKEditor"));
            $('.trigger-file-input').click(function () {
                $('input[type=file]').click();
            });
            $('.save-files').click(function () {
                var inp = document.getElementById('file');
                var fileNames = "Following files are uploading...";
                for (var i = 0; i < inp.files.length; ++i) {
                    fileNames += inp.files.item(i).name + "";
                }
                $('#fileNames').html(fileNames);
            });
            function getUrlParameter(sParam) {
                var sPageURL = window.location.search.substring(1);
                var sURLVariables = sPageURL.split('&);
                for (var i = 0; i < sURLVariables.length; i++) {
                    var sParameterName = sURLVariables[i].split('=');
                    if (sParameterName[0] == sParam) {
                        return sParameterName[1];
                    }
                }
            }
        });
    </script>
    <style type="text/css">
        ul li {
            list-style: none;
        }
    </style>
</head>
<body>

<form method="POST" id="formUploadLink" action="<c:url value="/admin/home/upload-classic" />"
     enctype="multipart/form-data">
    <span>Choose file or files to upload</span>
    <ul>
        <li>
            <div style="float: left;">
                <button type="button" class="buttonSubmit trigger-file-input">Choose File</button>
                <button type="submit" style="margin-left: 20px" class="buttonSubmit save-
files">UPLOAD</button>
            </div>
            <div style="clear: both"></div>
        </li>
        <li>
            <span id="fileNames"></span>
        </li>
        <li>
            <input type="file" id="file" multiple name="file[]">
        </li>

    </ul>
    <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
</form>

</body>
</html>
```

Spring Controller Class

```
/**
  * Handles requests for the application file upload requests
  */
@Controller
@RequestMapping(value = "/admin/home/", method = RequestMethod.GET)
public class FileUploadController {
     private static final Logger logger = LoggerFactory
             .getLogger(FileUploadController.class);

     private static String FILE_SEPARATOR = "/"

     @RequestMapping(value = {"upload", "uploadClassic"})
     public void upload() {

     }
     @RequestMapping(value = "/upload-single", method = RequestMethod.POST,
             produces = "text/html;charset=UTF-8")
     @ResponseBody
     public String uploadFileHandler(
             @RequestParam("file") MultipartFile file, HttpServletRequest request) {
         String url;
```

```
        String storedFolderLocation = createStoredFolder(request);
        String uploadedFileName = file.getOriginalFilename();
        try {
            byte[] bytes = file.getBytes();
            String storedFileLocation = storedFolderLocation + FILE_SEPARATOR + uploadedFileName;
            File serverFile = new File(storedFileLocation);
            BufferedOutputStream stream = new BufferedOutputStream(
                    new FileOutputStream(serverFile));
            stream.write(bytes);
            stream.close();
            url = getDomainName(request)
                    + getRelativePath() + FILE_SEPARATOR + uploadedFileName;
            if (isFileTypeImage(uploadedFileName)) {
                url= "<img src=\"" + url + "\" />";
            } else {
                url= "<a href=\"" + url + "\">" + url + "</a>";
            }
        } catch (Exception e) {
            return e.getMessage();
        }
        return "Loaded File:"+url;
    }
    /**
     * Upload multiple file using Spring Controller
     */
    @RequestMapping(value = "upload-classic", method = RequestMethod.POST,
            produces = "text/html;charset=UTF-8")
    @ResponseBody
    public String uploadMultipleFileHandler(@RequestParam("CKEditor") String ckEditor,
            @RequestParam("file[]") MultipartFile[] files, HttpServletRequest request) {

        String storedFolderLocation = createStoredFolder(request);

        String urls = "";

        for (MultipartFile file : files) {
            String uploadedFileName = file.getOriginalFilename();
            try {
                byte[] bytes = file.getBytes();

                String storedFileLocation = storedFolderLocation + FILE_SEPARATOR +
uploadedFileName;
                // Create the file on server
                File serverFile = new File(storedFileLocation);
                BufferedOutputStream stream = new BufferedOutputStream(
                        new FileOutputStream(serverFile));
                stream.write(bytes);
                stream.close();

                logger.info("Server File Location="
                        + serverFile.getAbsolutePath());
                String url = getDomainName(request)
                        + getRelativePath() + FILE_SEPARATOR + uploadedFileName;
                if (isFileTypeImage(uploadedFileName)) {
                    urls += "<img src=\"" + url + "\" />";
                } else {
                    urls += "<a href=\"" + url + "\">" + url + "</a>";
                }

            } catch (Exception e) {
                return "You failed to upload " + uploadedFileName + " => " + e.getMessage();
            }
        }
        if(!ckEditor.equals("undefined")){
            urls = "<script
type=\"text/javascript\">window.opener.CKEDITOR.instances."+ckEditor+".insertHtml('"
                    + urls + "');window.opener.CKEDITOR.dialog.getCurrent().hide();" +
                    "window.close();</script>";
        }else{
            urls="Uploaded files:"+urls;
        }

        return urls;
    }

    @RequestMapping(value = "upload-ajax", method = RequestMethod.POST)
    @ResponseBody
    public String uploadMultipleFiles(@RequestParam("CKEditor") String ckEditor,
                                      MultipartHttpServletRequest request) {
        String filePaths=uploadedFiles(request);
        if(!ckEditor.equals("undefined")){
            filePaths = "<script
type=\"text/javascript\">window.opener.CKEDITOR.instances."+ckEditor+".insertHtml('"
                    + filePaths + "');window.opener.CKEDITOR.dialog.getCurrent().hide();" +
                    "window.close();</script>";
        }else{
            filePaths="Yüklenen dosyalar:"+filePaths;
        }
        return filePaths;
    }
    private String uploadedFiles(MultipartHttpServletRequest request){
        CommonsMultipartFile multipartFile = null;
        Iterator<String> iterator = request.getFileNames();
        String filePaths = "";
        while (iterator.hasNext()) {
            String key = iterator.next();
            // create multipartFile array if you upload multiple files
            multipartFile = (CommonsMultipartFile) request.getFile(key);
            String uploadedFileName = multipartFile.getOriginalFilename();
            try {
                byte[] bytes = multipartFile.getBytes();

                String storedFileLocation = createStoredFolder(request) + FILE_SEPARATOR +
uploadedFileName;
                // Create the file on server
                File serverFile = new File(storedFileLocation);
                BufferedOutputStream stream = new BufferedOutputStream(
                        new FileOutputStream(serverFile));
                stream.write(bytes);
                stream.close();

                logger.info("Server File Location="
                        + serverFile.getAbsolutePath());
                String url = getDomainName(request)
                        + getRelativePath() + FILE_SEPARATOR + uploadedFileName;
                if (isFileTypeImage(uploadedFileName)) {
                    filePaths += "<img src=\"" + url + "\" />";
                } else {
                    filePaths += "<a href=\"" + url + "\">" + url + "</a>";
                }
            } catch (Exception e) {
                return "You failed to upload " + uploadedFileName + " => " + e.getMessage();
            }
        }
        return filePaths;
    }

    private boolean isFileTypeImage(String fileName) {
        String imagePattern =
                "([^\\s]+(\\.(?i)(jpg|jpeg|png|gif|bmp))$)";
        return Pattern.compile(imagePattern).matcher(fileName).matches();

    }
    private String getDomainName(HttpServletRequest request) {
        return request.getProtocol().toLowerCase().replaceAll("[0-9./]", "") + "://" +
                request.getServerName() + ":" + request.getServerPort();
    }

    private String createStoredFolder(HttpServletRequest request) {
        String realPath = request.getSession().getServletContext().getRealPath("/");
        String relativePath = getRelativePath();
        String storedFolderLocation = realPath + relativePath;
        File dir = new File(storedFolderLocation);
        if (!dir.exists()) {
            dir.mkdirs();
        }
        return storedFolderLocation;
    }

    private String getRelativePath() {
        String fileSeparator = "/";
        dateUtil = new DateUtil();
        int[] yearMonthDay = getDayMonthYear();
        return "/resources/uploads/" + yearMonthDay[0] + fileSeparator
                + yearMonthDay[1] + fileSeparator + yearMonthDay[2];
    }
    public int[] getDayMonthYear() {
        Calendar now = Calendar.getInstance();
        int year = now.get(Calendar.YEAR);
        int month = now.get(Calendar.MONTH); // Note: zero based!
        int day = now.get(Calendar.DAY_OF_MONTH);
        int[] date = new int[3];
        date[0] = year;
        date[1] = month;
        date[2] = day;
        return date;
    }
}
```

Note: If you are using SimpleUploads plugin, you should add following controller methods:

```
    @RequestMapping(value = "upload-drag-drop", method = RequestMethod.POST,produces =
"text/html;charset=UTF-8")
    @ResponseBody
    public String uploadDragDrop(@RequestParam("CKEditorFuncNum") String funcNumber,
                                 MultipartHttpServletRequest request){
        String fileName=request.getFileNames().next();
        CommonsMultipartFile multipartFile = (CommonsMultipartFile) request.getFile(fileName);
        String url;
        try {
            byte[] bytes = multipartFile.getBytes();
            String storedFileLocation = createStoredFolder(request) + FILE_SEPARATOR +
fileName;
            // Create the file on server
            File serverFile = new File(storedFileLocation);
            BufferedOutputStream stream = new BufferedOutputStream(
                    new FileOutputStream(serverFile));
            stream.write(bytes);
            stream.close();
            logger.info("Server File Location=" + serverFile.getAbsolutePath());
            url = getDomainName(request)
                    + getRelativePath() + FILE_SEPARATOR + fileName;
            return "<script
type=\"text/javascript\">window.parent.CKEDITOR.tools.callFunction("+funcNumber+
                    ",\""+url+"\", \"\");</script>";
        } catch (Exception e) {
            return "You failed to upload " + fileName + " => " + e.getMessage();
        }
    }
```

How can we verify CSRF in CKEditor Simpleuploads Plugin?

When we enable CSRF verification in a web application, we can encounter csrf verification error while using simpleuploads plugin. To solve this problem, we should use extra fields property as follows:

```
<script type="text/javascript">
    CKEDITOR.on('instanceReady', function (e) {
        // the real listener simpleuploads plugin
        e.editor.on('simpleuploads.startUpload', function (ev) {

            var csrfValue = $('input[name=_csrf]').val();
            var extraFields = {
                "_csrf": csrfValue

            };
            ev.data.extraFields = extraFields;
        });
    });
</script>
```