# Ajax

In order to exemplify the concepts of Ajax we will make some modifications in the project `classspring`.

1. Verify if your table in the database has a field named "`graduated`", whose type should be `boolean`. Note: this new field was added in the database in the Spring class.
2. Modify the list of students (file `list.jsp`), adding a column to the field “`graduated`”:

```
<table border=1>
      <tr>
            <td><b>Id</b></td>
            <td><b>Name</b></td>
            <td><b>E-mail</b></td>
            <td><b>Address</b></td>
            <td><b>Graduated?</b></td>
            <td><b>Change?</b></td>
            <td><b>Remove?</b></td>
      </tr>
      <c:forEach items="${students}" var="student">
            <tr>
                  <td>${student.id}</td>
                  <td>${student.name}</td>
                  <td>${student.email}</td>
                  <td>${student.address}</td>
                  <c:if test="${student.graduated eq false}">
                        <td>No</td>
                  </c:if>
                  <c:if test="${student.graduated eq true}">
                        <td>Yes</td>
                  </c:if>
                  <td><a href="showStudent?id=${student.id}">Change</a>
                  </td>
                  <td><a href="removeStudent?id=${student.id}">Remove</a>
                  </td>
            </tr>
      </c:forEach>
</table>
```

2. Modify the file `show.jsp` adding a field to store the value of the new field:

```
<input type="hidden" id="graduated" name="graduated" value="${student.graduated}"/
><br />
```

**Ajax (*Asynchronous Javascript and XML*)**: technique that allows sending asynchronous requests. In other words, this means that we can keep a page unchanged while we wait for the response to a request.

Let us implement in our project a link for each record for which the student is said to be graduated, without however leaving the current page, i.e., the page `listStudents`.

- Add Ajax in your project: create a folder named "`resources`" inside `WebContent`. There, create a folder named "`js`". Download the libraries from `jquery.com`, and add them in the folder `resources/js`.

- In the file `spring-context.xml`, add a command to map a place where the resources of the application can be found (images, css, etc). This also aims to warn the Spring controller to not answer requests that ask for this type of content:

  ```
  <mvc:resources location="/resources/" mapping="/resources/**"/>
  ```

- In the file `list.jsp`, add another column in the table:

```
<table border=1>
      <tr>
            <td><b>Id</b></td>
            <td><b>Name</b></td>
            <td><b>E-mail</b></td>
            <td><b>Address</b></td>
            <td><b>Graduated?</b></td>
            <td><b>Change status?</b></td>
            <td></td>
            <td></td>
      </tr>
      <c:forEach items="${students}" var="student">
            <tr>
                  <td>${student.id}</td>
                  <td>${student.name}</td>
                  <td>${student.email}</td>
                  <td>${student.address}</td>
                  <c:if test="${student.graduated eq true}">
                        <td>Yes</td>
                  </c:if>
                  <c:if test="${student.graduated eq false}">
                        <td id="td1student_${student.id }">No</td>
                  </c:if>
                  <c:if test="${student.graduated eq true}">
                        <td></td>
                  </c:if>
                  <c:if test="${student.graduated eq false}">
                        <td id="td2student_${student.id }"><a href="#"
onClick="changeStatus(${student.id})">Change</a></td>
                  </c:if>
                  <td><a href="showStudent?id=${student.id}">Change</a></td>
                  <td><a href="remove?id=${student.id}">Remove</a></td>
            </tr>
      </c:forEach>

</table>
```

- Still in the file `list.jsp`, modify the tag head as follows:

```
<head>
<script type="text/javascript" src="resources/js/jquery.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>List of Students</title>
</head>
```

- Still in the file `list.jsp`, add the script:

```
<script type="text/javascript">
function changeStatus(id){
      $.post("changeStatus",{'id' : id},function(){
            $("#td1student_"+id).html("Yes");
            $("#td2student_"+id).html("");
      })
}
</script>
```

- In the class `StudentController.java`, add the action:

```
@RequestMapping("changeStatus")
public void changeStatus(Long id,HttpServletResponse response){
      StudentDAO dao = new StudentDAO();
      dao.ends(id);
      response.setStatus(200);
}
```