

Taller Unidad de Repaso: Todo Listo

Introducción

En este taller se debe desarrollar la aplicación "Todo Listo", una aplicación front-end para mostrar, crear, y editar tareas. Para ello se deben utilizar las tecnologías de front-end vistas en la unidad de repaso: HTML, CSS, Javascript, y jQuery. Además se debe utilizar Bootstrap.

Instrucciones Preliminares

Esta aplicación utilizará un backend desarrollado en Django. Por ahora usted debe solo descargar y ejecutar este backend, y hacer que su aplicación lo utilice como una caja negra. **No es necesario entender ni modificar el backend**

1. Siga las instrucciones sobre instalación de Anaconda, y cree un nuevo ambiente `todolisto-repaso`. En este ambiente instale los siguientes paquetes Python.
2. Clone la aplicación Django para el backend de la aplicación desde el repositorio público:
`http://bitbucket.org/ifigueroap-pucv/todo-listo-backend`
3. Active su ambiente virtual `todolisto-repaso`, y diríjase a la carpeta `todo-listo-backend` recién clonada. Instale las dependencias de Python ejecutando el comando `pip install -r dependencies.txt`. Luego ejecute el comando `python manage.py migrate`.
4. Ejecute el backend con el comando `python manage.py runserver`. El backend estará corriendo, por defecto, en la url `http://localhost:8000/`. Observe que en esta URL se muestra una interfaz para revisar el API que provee este backend.
5. Utilice XAMPP, u otro servidor similar, para ejecutar la aplicación Todo Listo que se va a desarrollar en este taller. Su aplicación se debe ejecutar en el puerto 80.

Instrucciones

Implemente la aplicación Todo Listo según el diseño que se presenta en el archivo gif complementario `todolisto.gif` (ignore el uso del puerto 4200, que corresponde a Angular). Más específicamente debe implementar lo siguiente:

1. (10 pts) Utilizando Bootstrap, su aplicación debe tener el layout que aparece en la imagen: un encabezado, un pie de página, una columna con un formulario, y otra columna con una tabla.
2. (10 pts) Al iniciar su aplicación, se deben cargar las tareas que existan en el backend, utilizando AJAX y la url `http://localhost:8000/tareas`, usando método GET. Las tareas así obtenidas, en JSON, deben ser renderizadas en la tabla, con un enlace para poder seleccionar cada tarea.
3. (20 pts) Al seleccionar una tarea, se debe habilitar una sección abajo donde aparece la descripción de la tarea y un botón para pasar de estado al siguiente estado. Los estados posibles son: `Creada`, `En Proceso`, y `Terminada`. Una tarea terminada no debe mostrar botón para pasar a un estado siguiente. Al presionar el botón, el cambio debe ser persistente en el backend, para ello utilice AJAX para realizar

un request con método PUT a la url <http://localhost:8000/tareas/<ID-TAREA>>, donde [<ID-TAREA>](http://localhost:8000/tareas/<ID-TAREA>) se debe reemplazar con el id de la tarea que se está modificando.

4. (10 ptos) Implemente el formulario para que realice la validación correspondiente, según se muestra en la especificación. Utilice Bootstrap para la presentación del formulario.
5. (15 ptos) Implemente el procesamiento del formulario utilizando AJAX, teniendo como objetivo la URL <http://localhost:8000/tareas/>, y usando el método POST. Así, el backend creará una nueva tarea con los datos proporcionados.
6. (15 ptos) Agregue una barra de búsqueda que vaya filtrando de manera inmediata el listado de las tareas, de acuerdo a la similitud del título con el texto buscado.
7. (20 ptos) Modifique su aplicación para que sea posible modificar el título y la descripción de una tarea seleccionada, y que estos cambios persistan en el backend. Utilice el mismo método y URL que en la actualización de estado.