

Proyecto de Curso

Tema: Python: Introducción y servicios web

Fecha de entrega máxima: 6 de agosto 2023, 23:59

Este proyecto es autoguiado, es decir el estudiante deberá tomar ventaja de las diapositivas, el internet, la documentación oficial, o cualquier herramienta disponible para conseguir información correspondiente que le permita descubrir soluciones a los problemas presentados. El proyecto está diseñado para ser completado en un tiempo máximo de **20 horas**.

Detalles

En este proyecto expondremos una simple API REST para acceder a ciertas funciones de una escuela. El proyecto es **individual** y se calificará de acuerdo con la rúbrica especificada al final del presente documento.

Es recomendado revisar las diapositivas ya que los recursos provistos y los ejemplos son suficiente para poder empezar.

REPOSITORIO

1. Crearemos un repositorio en [GitHub](#) con el fin de mantener el código del proyecto. Para este propósito dirigirse al servicio y crear una cuenta en caso de no tenerla.
2. Una vez que tengamos una cuenta, crear un repositorio con el nombre **"ista-python-curso-2023"**.
3. El estudiante puede decidir si el repositorio es privado o público, sin embargo, será necesario que el capacitador tenga acceso, por lo que, en caso de crear un repositorio privado, agregar como colaborador a la cuenta **"cursosdoc"**.

ESTRUCTURA DEL REPOSITORIO

4. En el repositorio crearemos un archivo **README.md** que deberá contener el nombre completo del estudiante, así como cualquier detalle relevante sobre el código que deseen compartir.
5. En el repositorio crearemos una carpeta **"escuela_api"** que será donde vivirá todo nuestro código de Python para el servicio.
6. Para simplificar el acceso a información y las dependencias, la base de datos será nada más una serie de archivos que leeremos y cambiaremos con la ayuda Python. Creemos la carpeta **"datos"** para este propósito.
7. El repositorio solo debe contener los artefactos esenciales, recomendado usar un **.gitignore** que excluya archivos innecesarios durante el desarrollo del proyecto <https://www.toptal.com/developers/gitignore/api/python,flask>.

Como podemos ver, nuestro repositorio contendrá todos los requisitos necesarios para ejecutar el servicio.

BASE DE DATOS

Por simpleza, la base de datos es una serie de archivos CSV, un formato de texto plano que representa tablas mediante comas. Las dos tablas, **estudiante**, y **asistencia** están en el repositorio de la clase para que puedan copiarse sin problemas.

estudiante: una tabla que incluye los datos personales de los estudiantes, el identificador del estudiante es su número de cédula.

asistencia: una tabla que incluye los registros de asistencia de los estudiantes, el identificador del estudiante es su número de cédula.

8. Copiar las tablas CSV de la carpeta proyecto en [cursosdoc/ista-07-2023-python \(github.com\)](https://github.com/cursosdoc/ista-07-2023-python) en nuestra carpeta “**datos**”.
9. Agregar una línea a **estudiante.csv** con tus datos (NO ES NECESARIO USAR LA CÉDULA REAL).
10. Agregar 5 líneas a **asistencia.csv** con 5 días diferentes en la materia “**python**” para el estudiante agregado en el punto anterior.

PROYECTO: “escuela_api”

Queremos tener la habilidad de obtener cierta información sobre la asistencia de los estudiantes de nuestra escuela.

11. Agregar **flask**, **pytest** a un archivo **requirements.txt** dentro de este proyecto, e instalarlo mediante **pip**.
12. Creemos un paquete para un proyecto básico de **flask** dentro de esta carpeta (recuerda el `__init__.py`), con el módulo principal llamado **api.py**.
13. Creemos un punto de acceso **GET** en nuestra API “**lista_estudiantes**” que lea la tabla **estudiante** de nuestro repositorio y devuelva una lista de objetos JSON, con la lista de nombres y apellidos ordenados en orden **ascendente**. Tip: Buscar documentación sobre funciones para ordenar objetos en Python.
14. Creemos un punto de acceso **POST** en nuestra API “**registro_asistencia**” que reciba una cédula, nombre de materia, año, mes y hora, y las registre en la tabla **asistencia**.
15. Creemos un punto de acceso **GET** llamado “**total_asistencias**”, que reciba un parámetro de **cedula_estudiante**, y el nombre del curso, y devuelva el **total de asistencias** del estudiante para el curso provisto, en caso de que no existan asistencias para el curso solicitado, devolver **0**.

Importante: Ya que la base de datos es una serie de archivos, no necesitamos instalar nada complejo para leer los mismos.

TESTS

16. Dentro de cualquier proyecto crea una carpeta **tests** (recuerda el `__init__.py`), con un archivo **test_unidad.py**. Este archivo puede ser tan complejo como desees, lo importante es entender como llamarlos correctamente mediante **pytest**.

EXTRAS

Es importante que se implementen estos elementos **SÓLO SI EXISTE TIEMPO SUFICIENTE**.

Indicar en el archivo **README.md** si alguno de estos ítems ha sido implementado por el estudiante.

1. En la función **total_asistencias**, devolvemos un 0 si el estudiante o la clase requerida no existen. Sin embargo, es quizás más útil devolver un error HTTP con un mensaje claro que indique al consumidor el error específico. Si se decide usar un error HTTP específico, justificar por qué se utilizó el código escogido.
2. Crear un punto de acceso en la API que genere una tabla HTML de la lista de alumnos en lugar de devolver un objeto JSON.
3. Cambia la base de datos a base de archivos CSV por un mecanismo más eficiente. Por ejemplo, tinydb, o inclusive algo basado en SQL como sqlite/MySQL.
4. Ejecuta chequeos en tu repositorio, por ejemplo para tests, explora Github Actions o CircleCI.

Calificación

Enviar el nombre del repositorio y sus datos personales a la siguiente ficha:

<https://forms.gle/d1H7QuMXnJQWA1KL8>

El código se calificará en su estado a las 11:59pm del 6 de agosto, es decir que deben asegurarse que todas los cambios se encuentren publicados a esta hora. **Cualquier cambio luego de esta fecha no se considerará aceptable.**

Rubro	Descripción	%
Repositorio	README.md correspondiente y estructura de carpeta de acuerdo con la guía.	5
Base de datos	Cambios a los archivos originales.	5
API escuela, lista_estudiantes	Punto de acceso de acuerdo con los requisitos.	25
API escuela, registro_asistencia	Punto de acceso de acuerdo con los requisitos.	25
API escuela, total_asistencias	Punto de acceso de acuerdo con los requisitos.	35
Tests	Tests en cualquiera de los proyectos.	5
		100
Extras	Cualquier tarea adicional implementada.	10