
Api Proof

Release 1.0

Galileo Garibaldi, Simon Valiterra

Jan 26, 2022

CONTENTS:

1	Configuration	1
2	Api Student	3
3	Notes	5
4	Controller	7
5	Convert data	9
6	Student	11
7	Indices and tables	13
	Python Module Index	15
	Index	17

CONFIGURATION

Archivo de configuraciones para cada etapa del proyecto

class configuration.**BaseConfig**

Configuración de tipo Base en la cual se ponen los siguientes parametros:

1.- Secret key
2.- Debug
3.- Testing
4.- SQLAlchemy Database URI
5.- SQLAlchemy Track Modifications

class configuration.**DevelopmentConfig**

Configuración de Desarrollo

BaseConfig [class] Configuración base

class configuration.**ProductionConfig**

Configuración de producción

BaseConfig [class] Configuración base

API STUDENT

Archivo encargado de definir las rutas y los parámetros a funcionar dentro de cada endpoint con los protocolos Web (POST, GET, DELETE, PUT)

NOTES

Ruta en la que se pone la API: **/api/student/**

class `api3.student_api.api_student.ControllerStudent`

Clase encargada de definir los protocolos Web Get, Post, Delete, Put.

MethodView [class] Clase heredada de la biblioteca de Flask (flask.views)

delete(*id*)

Método encargado de eliminar registros

id [int] ID del Estudiante a eliminar

JSON JSON con la estructura mencionada en **sendResJson**

get(*id=None*)

Método Get para traer los datos de Estudiantes

id [int, optional] ID del estudiante a encontrar, by default None

JSON JSON con la estructura mencionada en **sendResJson**

post()

Método para publicar un nuevo estudiante

Se describe el Json que se manda para publicar un estudiante:: \$ { "age":2, "name": "Nombre",
"status":1 }

JSON JSON con la estructura mencionada en **sendResJson**

put()

Método encargado de modificar los registros

JSON JSON con la estructura mencionada en **sendResJson**

CONTROLLER

Archivo encargado de hacer las operaciones de DDL en SQL, es decir, agregar, seleccionar y eliminar

`api3.student_api.controller.convertToJson(student: api3.student_api.models.Student.Student)`

Convertir el objeto de tipo **Student**

student [Student] Objeto de tipo **Student**, el cual tiene como parámetros: id, name, age, state

JSON json con el formato de los tipos de dato

`api3.student_api.controller.delete_student(id)`

Eliminar un registro de el estudiante por su ID

id [int] ID del estudiante (PK)

Json json con el formato de los tipos de dato

`api3.student_api.controller.insert_student(json_data)`

Insertar datos al json de Estudiante

json_data [json] Json con el formato: { "age":2, "name": "Nombre", "status":1 }

Json json con el formato de los tipos de dato

`api3.student_api.controller.select_all_student()`

Seleccionar todos los registros de los estudiantes

Json json con el formato de los tipos de dato

`api3.student_api.controller.select_student_by_id(id)`

Seleccionar registros de estudiantes por su ID

id [int] Id del estudiante (PK)

Json json con el formato de los tipos de dato

CONVERT DATA

Archivo encargado de darle un formato establecido de json, el cual se describe a continuación::

```
{  
  'code': Codigo error o Aprobatorio  
  'data': lista de valores con el json de los datos  
  'message': Sucess - Fail  
}
```

`api3.helper.convert_data.sendResJson(data, code)`

Método encargada de Mapear los datos a un json estandarizado

data [list] lista de datos recibidos de una consulta, ésta puede estar vacía

code [int]

200 -> Satisfactorio.

404 -> No Encontrado.

Cualquier otro, dará fail

JSON Json de respuesta

STUDENT

Modelo de la tabla Student, la cual se mapea a la base de datos con la ayuda de la biblioteca SQLAlchemy

class `api3.student_api.models.Student.Student`(*name, age, status*)

Mapeado de Student

db [Class] Clase heredada de SQLAlchemy, **NO MODIFICABLE**. Se describen los parámetros que tiene la tabla Student:

- 1.- ID: Entero (Llave primaria)
- 2.- Name: Cadena
- 3.- Age: Entero
- 4.- Status: Bool

tuple Tupla con el siguiente formato: “Task Id, Name, Duracion”

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

a

`api3.helper.convert_data`, 7
`api3.student_api.api_student`, 1
`api3.student_api.controller`, 5
`api3.student_api.models.Student`, 9

C

`configuration`, ??

INDEX

A

`api3.helper.convert_data`
 module, 7
`api3.student_api.api_student`
 module, 1
`api3.student_api.controller`
 module, 5
`api3.student_api.models.Student`
 module, 9

B

`BaseConfig` (class in configuration), 1

C

`configuration`
 module, 1
`ControllerStudent` (class in
 `api3.student_api.api_student`), 5
`convertToJson()` (in module
 `api3.student_api.controller`), 7

D

`delete()` (`api3.student_api.api_student.ControllerStudent`
 method), 5
`delete_student()` (in module
 `api3.student_api.controller`), 7
`DevelopmentConfig` (class in configuration), 1

G

`get()` (`api3.student_api.api_student.ControllerStudent`
 method), 5

I

`insert_student()` (in module
 `api3.student_api.controller`), 7

M

module
 `api3.helper.convert_data`, 7
 `api3.student_api.api_student`, 1
 `api3.student_api.controller`, 5
 `api3.student_api.models.Student`, 9
 configuration, 1

P

`post()` (`api3.student_api.api_student.ControllerStudent`
 method), 5
`ProductionConfig` (class in configuration), 1
`put()` (`api3.student_api.api_student.ControllerStudent`
 method), 5

S

`select_all_student()` (in module
 `api3.student_api.controller`), 7
`select_student_by_id()` (in module
 `api3.student_api.controller`), 7
`sendResJson()` (in module `api3.helper.convert_data`),
 9
`Student` (class in `api3.student_api.models.Student`), 11