

Diseño de aplicaciones 2

OBLIGATORIO 1



Facundo Pujol 226033
Serafín Revetria 209143

Facultad ORT Uruguay

GitHub: <https://github.com/ORT-DA2/DA2-Pujol-Revetria.git>

Indice

[Discusión de los criterios seguidos para asegurar que la API cumple con los criterios REST](#)

[Descripción del mecanismo de autenticación de requests.](#)

[Descripción general de códigos de error](#)

[Descripción de los resources de la API](#)

[Administrators](#)

[Get](#)

[Post](#)

[Delete](#)

[Regions](#)

[Get](#)

[Touristic Spots](#)

[Post](#)

[Get](#)

[Accommodations](#)

[Post](#)

[Get](#)

[Put](#)

[Delete](#)

[Bookings](#)

[Post](#)

[Get](#)

[BookingStage](#)

[Post](#)

[Get](#)

Evidencia del diseño y especificación de la API

Discusión de los criterios seguidos para asegurar que la API cumple con los criterios REST

Dentro de las URIs:

Se utilizó sustantivos en minúscula y en plural, como fue recomendado en clase.

Como ejemplo de esto tenemos: administrators, regions o touristicspots.

Consideramos que son intuitivos y simples para que sea más simple de utilizar, ya que solo mediante la URI se logra comprender lo que hace el endpoint.

Se utilizó nombres concretos antes que abstractos como en el caso del endpoint de administrators donde se podía haber puesto persons en vez de administrators.

En cuanto a los niveles solo existe un endpoint que llega a los dos niveles maximos, que es touristicspots que tiene regionId y categories. No tenemos ningún caso que supere los tres niveles. Se utiliza ? en ciertos endpoints, como es el caso de touristicspots?RegionId=3&Categories=1. Y por último no existen endpoints que tengan verbos en la URI

Verbos HTTP:

Como fue enseñado en clase, utilizamos sólo los verbos GET POST PUT DELETE con únicamente su funcionalidad. No se utilizó ningún Verbo HTTP extra a estos como PATCH o TRACE. En el caso de los errores se utilizaron los HTTP status code y también nos ocupamos de que haya una respuesta entendible para la persona que recibe ese código de estado

Respuestas parciales: Se utilizaron DTOs (Data Transfer Object) donde se implemento que debería ser expuesto al usuario, en caso de los DTOs de salida. En el caso de los DTOs de entrada se determinó que información debe aportar el usuario, tomando en cuenta que no tiene toda la información de la entidad de negocio. Y por ultimo la devolucion de un objeto

Descripción del mecanismo de autenticación de requests.

Para la autenticación decidimos utilizar el método de autenticación simple mediante tokens, los cuales son generados por JWT (JSON Web Token), de esta manera podemos asegurarnos que los tokens serán hashado de manera correcta y segura. Para la autenticación utilizamos un IAuthenticationFilter que se encarga de llamar a la clase SessionLogic la cual tiene la responsabilidad tanto de crear el token en el login de administrators como de validarlo al momento de realizar una request que requiera autorización. A la hora de hacer login se crea un token utilizando como clave el email del

administrador y se lo devuelve al usuario, el cual deberá incluirlo en el header de las consultas que requiera autorización.

Descripción general de códigos de error

los códigos que más utilizamos fueron:

200 - OK

Lo encontramos en por ejemplo el login de un administrador con email y contraseña correcta.

400 - Bad Request

Lo encontramos por ejemplo al intentar crear un administrador sin contraseña.

401 - Unauthorized

Lo encontramos por ejemplo al intentar crear un administrador con email y contraseña válidas pero sin token.

403 - Forbidden

Lo encontramos por ejemplo al intentar crear un administrador con email y contraseña válidas pero con un token invalido para esa consulta.

404 - Not Found

Lo encontramos por ejemplo al intentar crear un administrador con email o contraseña no válida.

Descripción de los resources de la API

Administrators

URI: /api/administrators

```
{
  "email": string,
  "password": string
}
```

Get

Chequea si existe un administrador con ese email y esa contraseña.

URI: /login?email=admin&password=123

email	Identificador único de un administrador	Debe existir
password	Se chequea en caso de coincidir el email necesario	Debe ser igual, en caso de existir el email

200 - token (devuelve un token)
404 - Administrator Not Found

Post

Crea un nuevo administrador con ese email y esa contraseña. El usuario debe estar autenticado para realizar esta consulta.

Header: Authorization

Body:

```
{  
  "email": "test@test.com",  
  "password": "contrasena"  
}
```

email	Identificador único de un administrador	No debe existir en la BD
password	Se asocia a el email y se va a solicitar para hacer Login	

200 - Administrator created successfully
401 - No token provided
400 - The Admin Email cannot be null
400 - The Admin Password cannot be null
400 - The Administrator Already Exists

Delete

Borra un administrador del sistema

URI: /2

Header: Authorization

AdministratorId	Debe existir un administrador con esa id	Debe ser un número
-----------------	--	--------------------

200 -
404 - Administrator not found
401 - No token provided
400 - {
 "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
 "title": "One or more validation errors occurred.",
 "status": 400,
 "traceId": "|8eed864a-4902b34a6f04b64f.",
 "errors": {
 "id": [
 "The value 'a' is not valid."]]
 }

Regions

URI: /api/regions

```
{
  "id":int,
  "Name":string
}
```

Get

Trae todas las regiones

URI: -

200 -

```
{
  {
    "id": 1,
    "name": "Region Oeste"
  }
}
```

Touristic Spots

URI: /api/touristicspots

```
{
  "Name":string,
  "RegionId": int,
  "Description": string,
  "Image": string,
  "Categories": [int]
}
```

Post

Crea un nuevo punto turístico

Header: Authorization

Body: {

```
  "Name":"La Paloma",
  "RegionId": 3,
  "Description": "Pueblo de hermosas playas",
  "Image": "FOTO",
  "Categories": [1,2]
}
```

Name	Identifica a un TouristicSpot	Debe ser un string, requerido
RegionId	Identifica a la región asociada al TouristicSpot	Debe ser un int y debe existir un región con ese id,

		requerido
Description	Describe al TouristicSpot	Debe ser un string, required
Image	Imagen del TouristicSpot	Debe ser un string

Response:

```

200 - {
  "id":3,
  "name":"La Paloma",
  "description": "Pueblo de hermosas playas",
  "Image": "FOTO"
}
401 - No token provided
400 - The Spot Already Exists
400 - The Spot Categories cannot be null
400 - The Spot Name cannot be null
400 - The Spot Region cannot be null
400 - The Spot Description cannot be null
400 - {
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "|b5aea160-4b77b2f80b3aa9c0.",
  "errors": {
    "$.Name": [
      "The JSON value could not be converted to System.String. Path: $.Name |
LineNumber: 1 | BytePositionInLine: 12."
    ]
  }
}

```

Get

Retorna puntos turísticos, dependiendo de si cumplen o no con las características solicitadas. De no haber características, devuelve todos.

RegionId	Identifica a qué región deben estar asociadas los puntos turísticos a traer	Debe ser un int, opcional, va en la query
Categories	Identifica a qué categoría deben pertenecer los puntos turísticos a traer	Debe ser un int, opcional, va en el query

200 - [{

```

        "id": 3,
        "name": "La Paloma",
        "description": "Pueblo de hermosas playas",
        "image": "FOTO"
    }}
200 - []
400- {
    "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
    "title": "One or more validation errors occurred.",
    "status": 400,
    "traceId": "|183746ac-4338205a586477e7.",
    "errors": {
        "categories": [
            "The value " is invalid."
        ]
    }
}

```

Accommodations

URI: /api/accommodations

```

{
    "id": int,
    "name": string,
    "address": string,
    "contactNumber": string,
    "information": string,
    "price": int,
    "images": [string]
}

```

Post

Crea una nueva accommodation

Header: Authorization

Body: {

```

    "Name": "Hilton",
    "Address": "Mar Sereno 1045",
    "Price": 48.2,
    "Contact": "+598",
    "Information": "Un Hotel muy lindo",
    "SpotId": 1,
    "Images": ["fotoafuera", "fotohabitacion", "fotojardin"]
}

```

Name	Identificador de una	Debe ser un string,
------	----------------------	---------------------

	accommodation, nombre de la misma	requerida
Address	Dirección de la accommodation	Debe ser un string, requerida
Price	Precio por noche de la accommodation	Double, requerido
Contact	Contacto de la accommodation	Debe ser un string, requerido
Information	Breve informacion de la accommodation	Debe ser un string, requerido
SpotId	Id del Spot al que se encuentra asociada la accommodation	Debe ser un int, requerido
Image	Imagenes del accommodation	Debe ser un string, requerido

Response:

200 - {

```
"id": 4,
"name": "Hilton",
"address": "Mar Sereno 1045",
"contactNumber": "+598",
"information": "Un Hotel muy lindo",
"price": 48.2,
"images": [
  "fotoafuera",
  "fotohabitacion",
  "fotojardin"
]
}
```

401 - No token provided

400 - The Accommodation Already Exists

400 - Spot ID

400 - The Accommodation Name cannot be null

400 - The Accommodation Address cannot be null

400 - The Accommodation Spot cannot be null

400 - The Accommodation Images cannot be null

400 - The Accommodation Contact cannot be null

400 - The Accommodation Info cannot be null

400 - {

```
"type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
"title": "One or more validation errors occurred.",
"status": 400,
"traceId": "|183746be-4338205a586477e7.",
"errors": {
```

```

    "$.Name": [
      "The JSON value could not be converted to System.String. Path: $.Name |
LineNumber: 1 | BytePositionInLine: 12."
    ]
  }
}
400 - {
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "|183746c0-4338205a586477e7.",
  "errors": {
    "$.Price": [
      "The JSON value could not be converted to System.Double. Path: $.Price |
LineNumber: 3 | BytePositionInLine: 16."
    ]
  }
}
400 - {
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "|183746c2-4338205a586477e7.",
  "errors": {
    "$.Price": [
      "The JSON value could not be converted to System.Double. Path: $.Price |
LineNumber: 3 | BytePositionInLine: 15."
    ]
  }
}

```

Get

Retorna accommodations que cumplan con las características solicitadas.

Id	Identifica a una accommodation	Debe ser un int, opcional, va en la query. Si hay Id no hay SpotId URI: /1
SpotId	Identifica a qué TouristicSpot deben pertenecer las accommodations a traer	Debe ser un int, opcional, va en el query. Si hay SpotId no hay Id URI: /spot/1

```

200 - {
  "id": 3,
  "name": "La Paloma",
  "description": "Pueblo de hermosas playas",
  "image": "FOTO"
}
200 - [{
  "id": 3,

```

```

    "name": "La Paloma",
    "description": "Pueblo de hermosas playas",
    "image": "FOTO"
  }
}
200 - {}
400- {
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "|8eed8660-4902b34a6f04b64f.",
  "errors": {
    "spot": [
      "The value 'f' is not valid."
    ]
  }
}

```

Put

Actualiza la capacidad de una accommodation

Header: Authorization

Body: true

Id	Identifica a que accommodation se le quiere cambiar la capacidad	Debe ser un int, requerido, va en query URI: /Id
----	--	---

Response:

200 - Accommodation updated

404 - Accommodation Not Found

```

415 - {
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.13",
  "title": "Unsupported Media Type",
  "status": 415,
  "traceId": "|8eed8668-4902b34a6f04b64f."
}

```

Delete

Elimina un accommodation de la base de datos

Header: Authorization

Id	Identifica que accommodation se quiere borrar	Debe ser un int, requerido, va en la query URI: /Id
----	---	--

Response:

404 - Accommodation Not Found

401 - No token provided

```

200 - {
  "id": 7,
  "spot": {
    "id": 1,
    "name": "La Paloma",
    "region": null,
    "regionId": 3,
    "description": "Pueblo de hermosas playas",
    "categories": null,
    "accommodations": [],
    "image": "FOTO"
  },
  "spotId": 1,
  "name": "Nirvana",
  "address": "Capicua 1045",
  "contactNumber": "+598",
  "information": "Gran Hotel",
  "pricePerNight": 50.3,
  "full": false,
  "bookings": null,
  "images": [
    {
      "id": 13,
      "image": "fotoafuera",
      "accommodationId": 7,
      "accommodation": null
    },
    {
      "id": 14,
      "image": "fotohabitacion",
      "accommodationId": 7,
      "accommodation": null
    },
    {
      "id": 15,
      "image": "fotojardin",
      "accommodationId": 7,
      "accommodation": null
    }
  ]
}
}

```

Bookings

URI: /api/bookings

```

{
  "accommodationId": int,
  "checkIn": string,
  "CheckOut": string,

```

```

"GuestEmail": string,
"GuestName": string,
"GuestLastName": string,
"Guests": [
  {
    "Amount": int,
    "Multiplier": double
  }
]
}
}

```

Post

Crea una nueva booking en el sistema

Body: {

```

"accommodationId": 1,
"checkIn": "2020-12-12",
"CheckOut": "2020-12-24",
"GuestEmail": "test@test.com",
"GuestName": "test",
"GuestLastName": "lastTest",
"Guests": [
  {
    "Amount": 2,
    "Multiplier": 1
  }
]
}
}

```

accommodationId	Identifica a que accommodation está asociada la booking	Debe ser un int, requerido
checkIn	Identifica en qué fecha se ingresa al hospedaje	Debe ser un string, requerido
checkOut	Identifica en qué fecha se retira hospedaje	Debe ser un string, requerido
GuestEmail	Identifica el email del HeadGuest	Debe ser un string, requerido
GuestName	Identifica el Name del HeadGuest	Debe ser un string, requerido
GuestLastName	Identifica el LastName del HeadGuest	Debe ser un string, requerido
Guests	Lista de Guests	Debe ser un <int,double>, requerido

Response:

```
200 - {
  "id": 4,
  "accommodationId": 1,
  "accommodationName": null,
  "accommodationAddress": null,
  "accommodationContact": null,
  "checkIn": "2020-12-12T00:00:00",
  "checkOut": "2020-12-24T00:00:00",
  "price": 6,
  "guestEmail": "test@test.com"
}
400 - The Booking Guests cannot be null
400 - The Tourist Last Name cannot be null
400 - The Tourist Name cannot be null
400 - The Tourist Email cannot be null
400 - The CheckIn entered is invalid
400 - The CheckOut entered is invalid
400 - The Booking Accommodation cannot be null
```

Get

Retorna bookings que cumplan con las características solicitadas. En caso de no haber características trae todas las bookings.

Id	Identifica que booking debe traer	Debe ser un int, es opcional, va en la query URI: /1.
----	-----------------------------------	---

Response:

```
200 - {
  "id": 1,
  "accommodationId": 1,
  "accommodationName": "Hilton",
  "accommodationAddress": "Mar Sereno 1045",
  "accommodationContact": "+598",
  "checkIn": "2020-12-12T00:00:00",
  "checkOut": "2020-12-24T00:00:00",
  "price": 6,
  "guestEmail": "test@test.com"
}
404 - Booking Not Found
200 - [
  {
    "id": 1,
    "accommodationId": 1,
    "accommodationName": "Hilton",
```

```

    "accommodationAddress": "Mar Sereno 1045",
    "accommodationContact": "+598",
    "checkIn": "2020-12-12T00:00:00",
    "checkOut": "2020-12-24T00:00:00",
    "price": 6,
    "guestEmail": "test@test.com"
  },
  {
    "id": 4,
    "accommodationId": 1,
    "accommodationName": "Hilton",
    "accommodationAddress": "Mar Sereno 1045",
    "accommodationContact": "+598",
    "checkIn": "2020-12-12T00:00:00",
    "checkOut": "2020-12-24T00:00:00",
    "price": 6,
    "guestEmail": "test@test.com"
  }
]

```

BookingStage

URI: /api/bookingstages

```

{
  "BookingId": int,
  "Description": string,
  "Status": string,
  "AdminId": int
}

```

Post

Crea un bookingStage asociado a una booking

BookingId	Identifica a qué booking está relacionado	Debe ser un int, debe existir esa booking,required
-----------	---	--

404 - BookingStage Booking Not Found

200 - {

```

  "description": "Se Pago",
  "status": "Accepted"
}

```

404 - Status Not Found

400 - The Booking Stage Description cannot be null

Get

Retorna el bookingStage actual de una booking en específico.

bookingId	Identifica de qué booking debe traer el actual bookingStage	Debe ser un int, debe existir esa booking, required
-----------	---	---

```
200 - {  
  "description": "Se Pago",  
  "status": "Accepted"  
}
```

200 - The booking has not been changed by an Administrator