A dark blue vertical bar on the left side of the page. A magenta arrow points to the right from this bar, containing the date.

22-2-2024

How To Service RestFul Vuelos

Manual de Servicio

Several thin, curved lines in dark blue and light grey that sweep upwards from the bottom left corner of the page.

Sheila Lara García
I.E.S RIBERA DEL TAJO

INDICE

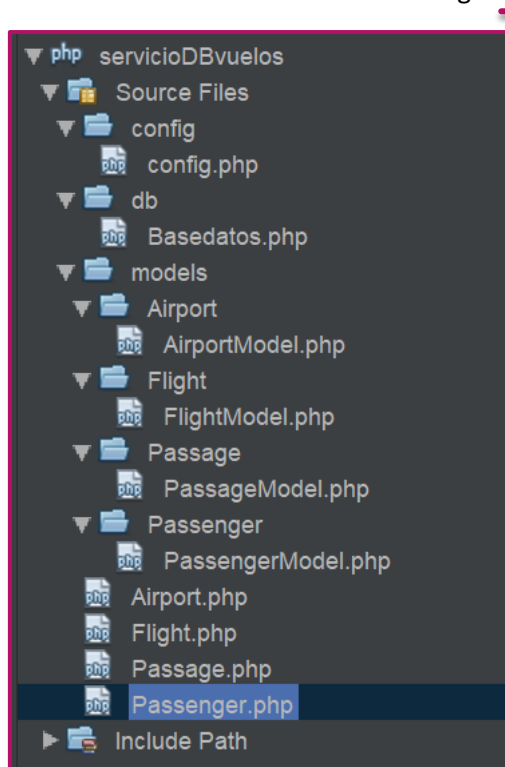
INDICE.....	1
1. DESCRIPCION DEL SERVICIO	2
1.1. Estructuras del servicio	2
2. METODOS.....	2
2.1. Aeropuertos	2
2.1.1. Todos los aeropuertos (Extra)	2
2.1.2. Aeropuerto por código (Extra)	3
2.2. Vuelos.....	4
2.2.1. Todos los vuelos (Obligatorio).....	4
2.2.2. Vuelo por identificador (Obligatorio).....	4
2.3. Pasajes.....	5
2.3.1. Todos los pasajes (Extra)	5
2.3.2. Pasaje por código (Extra).....	6
2.3.3. Pasaje por identificador de vuelo (Opcional).....	6
2.3.4. Insertar un pasaje (Obligatorio)	7
2.3.5. Errores de inserción (Obligatorios)	8
2.3.6. Modificar un pasaje (Obligatorio)	9
2.3.7. Error al actualizar (Obligatorio).....	10
2.3.8. Eliminar un pasaje (Obligatorio).....	11
2.3.9. Error al eliminar un pasaje (Obligatorio).....	11
2.4. Pasajeros	12
2.4.1. Todos los pasajeros (Extra).....	12
2.4.2. Pasajero por código (Extra)	13

1. DESCRIPCION DEL SERVICIO

El servicio estará alojado en la carpeta **_servWeb** ubicada en la carpeta **htdocs** de xampp. El servicio se llama **servicioDBvuelos**. Las peticiones realizadas al servicio se realizan mediante **URI's** y este devolverá un **JSON** con los datos pedidos o con un mensaje.

1.1. Estructuras del servicio

El servicio estará estructurado de la siguiente manera:



Config: Carpeta que contiene la configuración para el acceso a la base de datos.

Db: Carpeta que contiene el archivo para la conexión a la base de datos utilizando el archivo guardado en config.

Models: Carpeta que contiene los distintos modelos que contienen las funciones que se realizan sobre la base de datos.

Los archivos **Airport.php**, **Flight.php**, **Passage.php** y **Passenger.php** son los archivos de ejecución que llama a sus respectivos modelos.

Ilustración 1 Estructura servicio

2. METODOS

2.1. Aeropuertos

2.1.1. Todos los aeropuertos (Extra)

GET → Consulta todos los aeropuertos.



Ilustración 2 URI de todos los aeropuertos

```

1 [
2   {
3     "codaeropuerto": "BRU EBBR",
4     "nombre": "Bruselas-Zaventem",
5     "ciudad": "Bruselas",
6     "pais": "B?lgica",
7     "tasa": 20
8   },
9   {
10    "codaeropuerto": "CPH EKCH",
11    "nombre": "Copenhagen-Kastrup",
12    "ciudad": "Copenhague",
13    "pais": "Dinamarca",
14    "tasa": 150
15  },
16  {
17    "codaeropuerto": "GWY EICM",
18    "nombre": "aeropuerto de Galway",
19    "ciudad": "Galway",
20    "pais": "Irlanda",
21    "tasa": 30
22  }
23 ]

```

Ilustración 3 Resultado

2.1.2. Aeropuerto por código (Extra)

GET → Devuelve los datos de un aeropuerto dependiendo de su código de aeropuerto.

URI

http://localhost/_servWeb/servicioDBvuelos/Airport.php?codaeropuerto=BRU EBBR

Añadimos el código del aeropuerto en la **URI** para obtener los datos de ese aeropuerto.

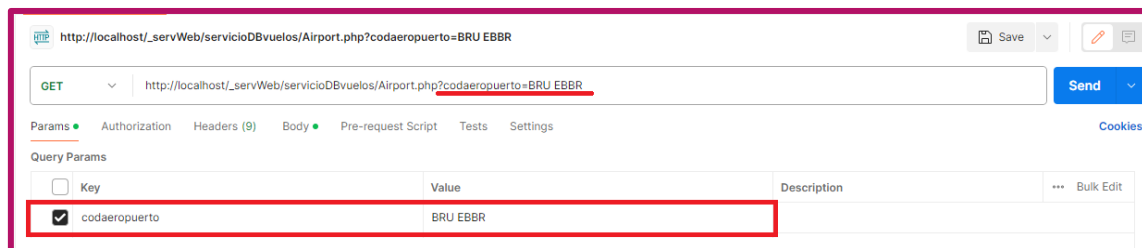


Ilustración 4 URI de un aeropuerto por código

```

1 {
2   "codaeropuerto": "BRU EBBR",
3   "nombre": "Bruselas-Zaventem",
4   "ciudad": "Bruselas",
5   "pais": "B?lgica",
6   "tasa": 20
7 }

```

Ilustración 5 Resultado

2.2. Vuelos

2.2.1. Todos los vuelos (Obligatorio)

GET → Devuelve todos los vuelos.

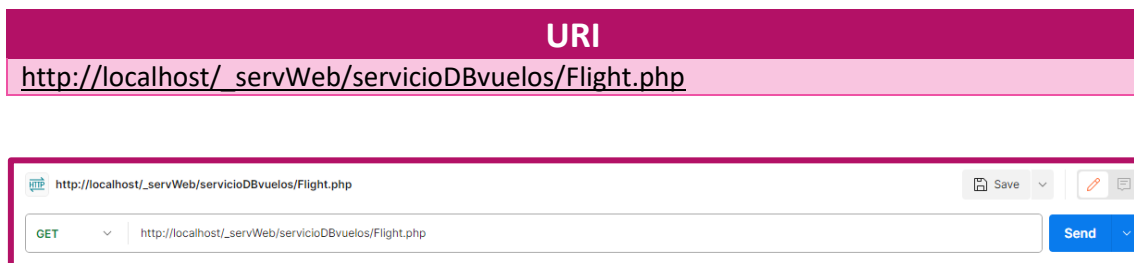


Ilustración 6 URI todos los vuelos

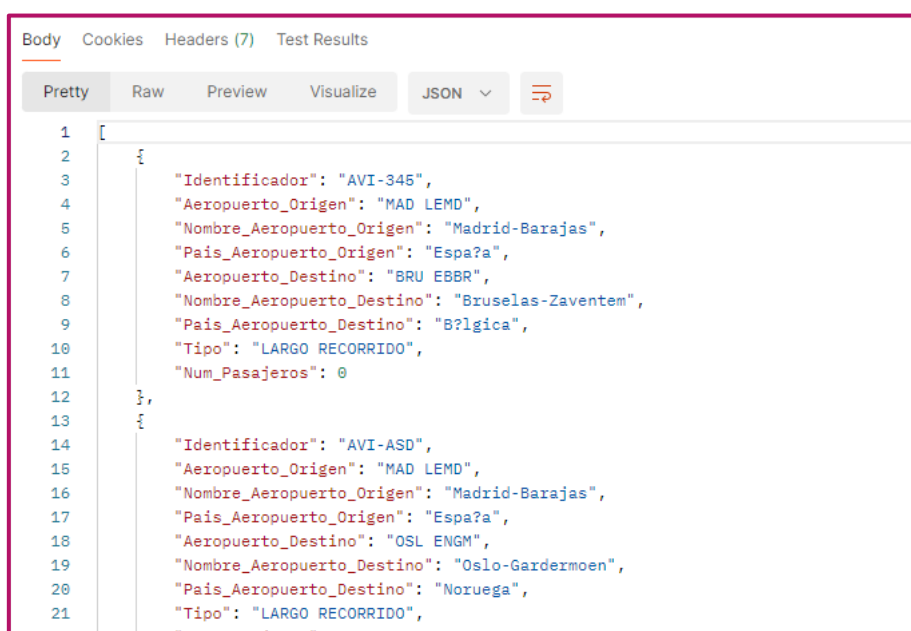


Ilustración 7 Resultado

2.2.2. Vuelo por identificador (Obligatorio)

GET → Devuelve los datos de un vuelo dependiendo del identificador.



Ilustración 8 URI de un vuelo por identificador



```

1 {
2   "Identificador": "AVI-ASD",
3   "Aeropuerto_Origen": "MAD LEMD",
4   "Nombre_Aeropuerto_Origen": "Madrid-Barajas",
5   "Pais_Aeropuerto_Origen": "Españ",
6   "Aeropuerto_Destino": "OSL ENGM",
7   "Nombre_Aeropuerto_Destino": "Oslo-Gardermoen",
8   "Pais_Aeropuerto_Destino": "Noruega",
9   "Tipo": "LARGO RECORRIDO",
10  "Num_Pasajeros": 12
11 }

```

Ilustración 9 Resultado

2.3. Pasajes

2.3.1. Todos los pasajes (Extra)

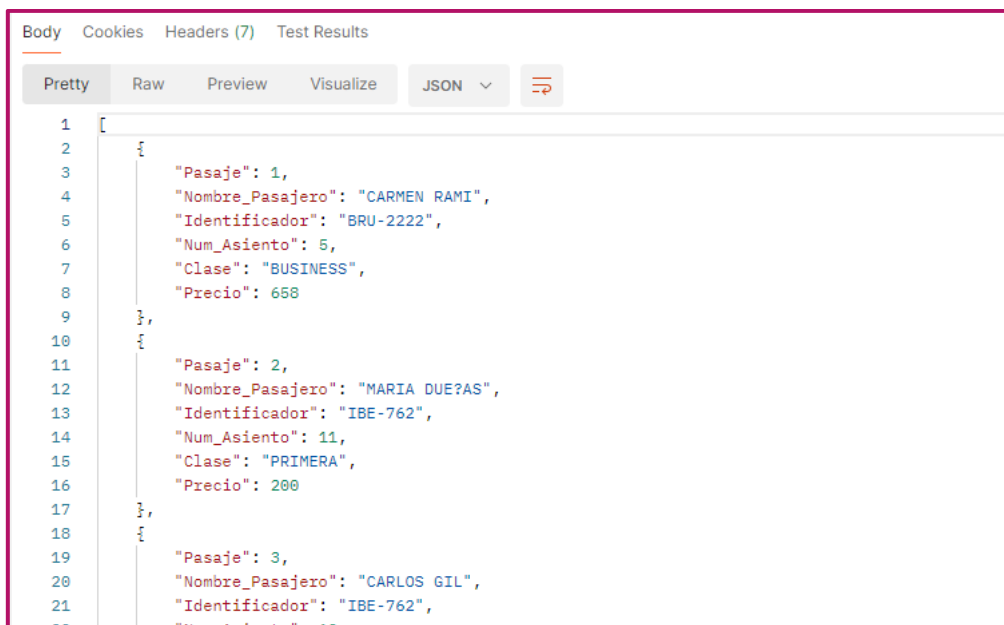
GET → Devuelve todos los pasajes.

URI

http://localhost/_servWeb/servicioDBvuelos/Passage.php



Ilustración 10 URI todos los pasajes



```

1 [
2   {
3     "Pasaje": 1,
4     "Nombre_Pasajero": "CARMEN RAMI",
5     "Identificador": "BRU-2222",
6     "Num_Asiento": 5,
7     "Clase": "BUSINESS",
8     "Precio": 658
9   },
10  {
11    "Pasaje": 2,
12    "Nombre_Pasajero": "MARIA DUEÑAS",
13    "Identificador": "IBE-762",
14    "Num_Asiento": 11,
15    "Clase": "PRIMERA",
16    "Precio": 200
17  },
18  {
19    "Pasaje": 3,
20    "Nombre_Pasajero": "CARLOS GIL",
21    "Identificador": "IBE-762",
22    "Num_Asiento": 11,
23    "Clase": "PRIMERA",
24    "Precio": 200
25  }
26 ]

```

Ilustración 11 Resultado

2.3.2. Pasaje por código (Extra)

GET → Devuelve los datos de un pasaje dependiendo del código del pasaje.

URI
http://localhost/servWeb/servicioDBvuelos/Passage.php?idpasaje=1

Añadimos el código del pasaje para obtener los datos del pasaje.



Ilustración 12 URI de un pasaje por código

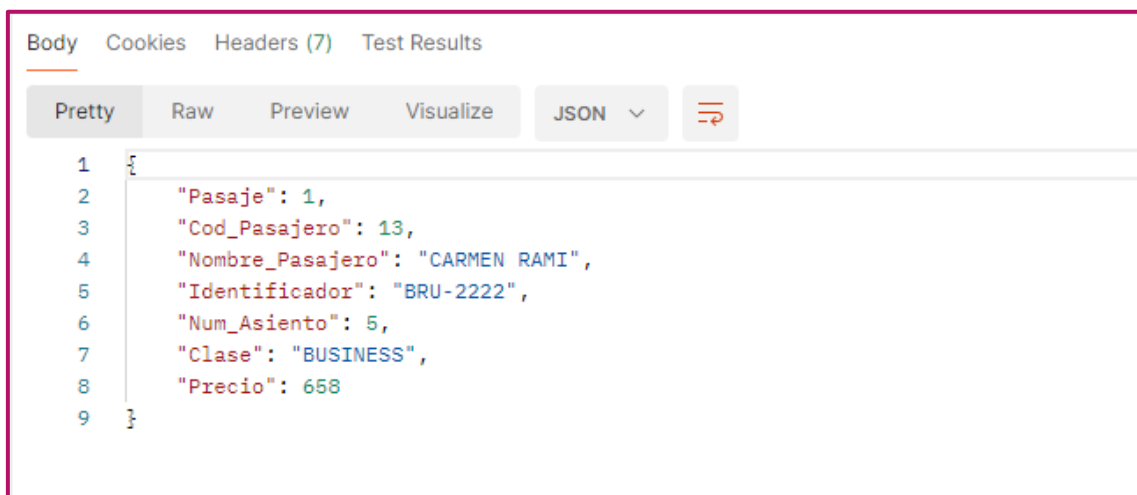


Ilustración 13 Resultado

2.3.3. Pasaje por identificador de vuelo (Opcional)

GET → Devuelve los datos de los pasajes relacionados con un identificador de un vuelo.

URI
http://localhost/servWeb/servicioDBvuelos/Passage.php?identificador=AVI-ASD

Añadimos el identificador del vuelo para obtener los datos de los pasajes relacionados con un vuelo.



Ilustración 14 URI de los pasajes de un vuelo

```

1  [
2    {
3      "Pasaje": 33,
4      "Codigo_Pasajero": 15,
5      "Nombre_Pasajero": "JUANJO P?REZ",
6      "Pais_Pasajero": "PORTUGAL",
7      "Num_Asiento_Pasajero": 30,
8      "Clase": "TURISTA",
9      "Precio": 170
10   },
11   {
12     "Pasaje": 34,
13     "Codigo_Pasajero": 16,
14     "Nombre_Pasajero": "PEDRO AL?A",
15     "Pais_Pasajero": "PORTUGAL",
16     "Num_Asiento_Pasajero": 33,
17     "Clase": "PRIMERA",
18     "Precio": 180
19   },
20   {
21     "Pasaje": 35,
22     "Codigo_Pasajero": 17,
23     "Nombre_Pasajero": "SERGIO GRAC?A",
24     "Pais_Pasajero": "PORTUGAL",
25     "Num_Asiento_Pasajero": 34,
26     "Clase": "PRIMERA",
27     "Precio": 220
28   }
29 ]

```

Ilustración 15 Resultado

2.3.4. Insertar un pasaje (Obligatorio)

POST → Inserta un nuevo pasaje a la tabla pasajes de la base de datos.

URI
http://localhost/_servWeb/servicioDBvuelos/Passage.php

Para poder insertar se hará mediante el método **POST**. En el **body** se pondrán los datos a insertar en código **JSON**.

```

1  {
2    "pasajeecod": 15,
3    "Identificador": "BRU-2222",
4    "numAsiento": 5,
5    "clase": "BUSINESS",
6    "pvp": 200
7  }

```

Ilustración 16 URI y body de la inserción de un nuevo pasaje

```

1  {
2    "message": "Nuevo pasaje insertado."
3  }

```

Ilustración 17 Resultado

<input type="checkbox"/>	Editar	Copiar	Borrar	60	22	QAT-900	25	BUSINESS	170
<input type="checkbox"/>	Editar	Copiar	Borrar	61	23	QAT-900	40	TURISTA	100
<input type="checkbox"/>	Editar	Copiar	Borrar	62	24	QAT-900	30	BUSINESS	220
<input type="checkbox"/>	Editar	Copiar	Borrar	63	25	QAT-900	35	BUSINESS	230
<input type="checkbox"/>	Editar	Copiar	Borrar	64	26	QAT-900	41	PRIMERA	120
<input type="checkbox"/>	Editar	Copiar	Borrar	66	15	BRU-2222	5	BUSINESS	250

Ilustración 18 Verificación en la base de datos

2.3.5. Errores de inserción (Obligatorios)

2.3.5.1. Insertar un pasajero que ya esté en un vuelo

Este error aparecerá cuando al insertar un nuevo pasaje el pasajero ya este relacionado con un vuelo.

Por ejemplo: Intentaremos volver a insertar el pasaje creado en el punto anterior (Insertar un pasaje (Obligatorio)).

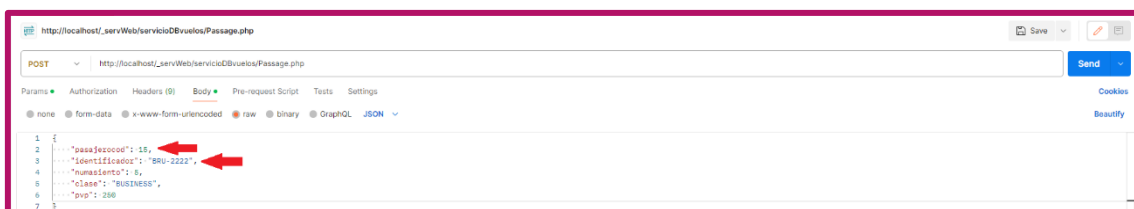


Ilustración 19 Insertar un pasajero que ya esté en un vuelo

Como podemos ver los datos son los mismo que hemos utilizado anteriormente, por lo tanto, el resultado es un mensaje de error.

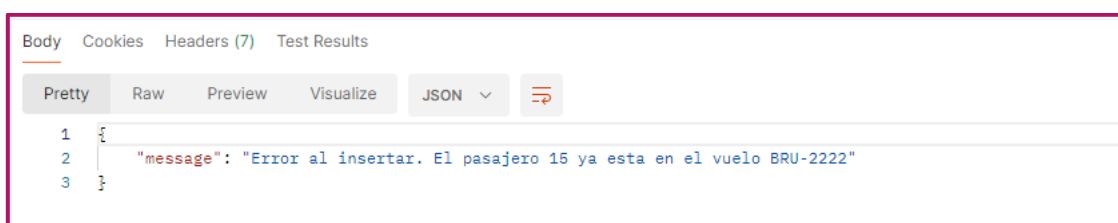


Ilustración 20 Mensaje de error

2.3.5.2. Insertar un pasaje que tenga ocupado el asiento en un vuelo

Este error aparecerá cuando al insertar un pasaje el vuelo tenga ocupado el asiento.

Por ejemplo: Ahora se insertarán otros datos distintos a los anteriores.

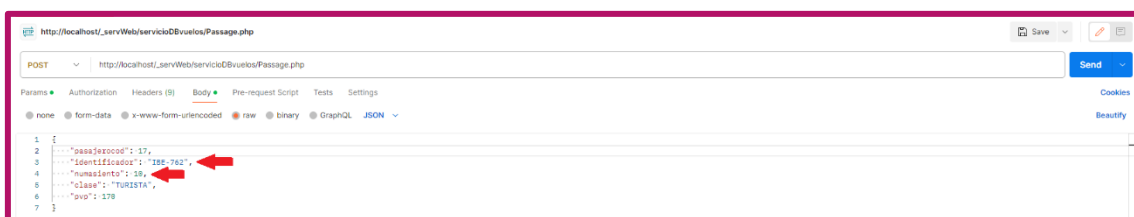


Ilustración 21 Insertar un asiento ya ocupado en el vuelo

Si el asiento ya esta ocupado en ese vuelo, el resultado de la inserción es un mensaje de error.

```

1 {
2   "message": "Error al insertar. El asiento 10 ya esta reservado en el vuelo IBE-762"
3 }

```

Ilustración 22 Mensaje de error

Si vamos a la base de datos podemos comprobar que el error es correcto ya que si buscamos ese vuelo y ese asiento vemos que ya está ocupado.

		idpasaje	pasajero	cod	identificador	numasiento	clase	pvp
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	1	IBE-762	10 PRIMERA	200
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	2	IBE-762	11 PRIMERA	200
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	3	IBE-762	13 PRIMERA	220
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	4	IBE-762	14 TURISTA	230
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5	5	IBE-762	16 TURISTA	250

Ilustración 23 Verificación en la base de datos

2.3.6. Modificar un pasaje (Obligatorio)

PUT → Actualizar un pasaje de la base de datos.

URI
<http://localhost/servWeb/servicioDBvuelos/Passage.php?idpasaje=66>

Para poder insertar se hará mediante el método **PUT**. En la **URI** pasaremos el **id del pasaje** que vamos a actualizar y en el **body** se pondrán los datos a insertar en código **JSON**.

```

1 {
2   "pasajero": 17,
3   "identificador": "BRU-2222",
4   "numasiento": 10,
5   "clase": "PRIMERA",
6   "pvp": 200
7 }

```

Ilustración 24 Actualización del pasaje

```

1 {
2   "message": "Pasaje 66 modificado."
3 }

```

Ilustración 25 Resultado

Si vamos a la base de datos comprobaremos que los datos que hemos introducido se han modificado correctamente.

<input type="checkbox"/>				60	22	QAT-900	25	BUSINESS	170
<input type="checkbox"/>				61	23	QAT-900	40	TURISTA	100
<input type="checkbox"/>				62	24	QAT-900	30	BUSINESS	220
<input type="checkbox"/>				63	25	QAT-900	35	BUSINESS	230
<input type="checkbox"/>				64	26	QAT-900	41	PRIMERA	120
<input type="checkbox"/>				66	17	BRU-2222	10	PRIMERA	200

Ilustración 26 Verificación en la base de datos

2.3.7. Error al actualizar (Obligatorio)

2.3.7.1. Asiento ya ocupado en el vuelo

Si al modificar un pasaje ponemos un asiento de un vuelo que ya esta ocupado nos mostrará un mensaje de error.

Por ejemplo: Vamos a modificar el pasaje modificado anteriormente (Modificar un pasaje (Obligatorio)).

<input type="checkbox"/>				60	22	QAT-900	25	BUSINESS	170
<input type="checkbox"/>				61	23	QAT-900	40	TURISTA	100
<input type="checkbox"/>				62	24	QAT-900	30	BUSINESS	220
<input type="checkbox"/>				63	25	QAT-900	35	BUSINESS	230
<input type="checkbox"/>				64	26	QAT-900	41	PRIMERA	120
<input type="checkbox"/>				66	17	BRU-2222	10	PRIMERA	200

Ilustración 27 Pasaje a modificar

Y le vamos a cambiar el **identificador de vuelo** y el **número de asiento** por uno que ya este ocupado para que nos aparezca el error.

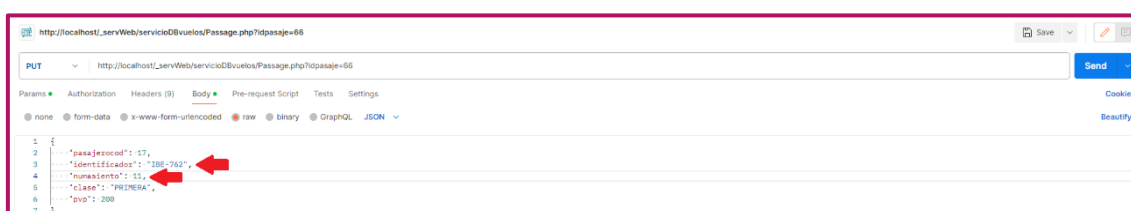


Ilustración 28 Datos a modificar

Como ese asiento ya esta ocupado en el vuelo nos aparecerá el error.

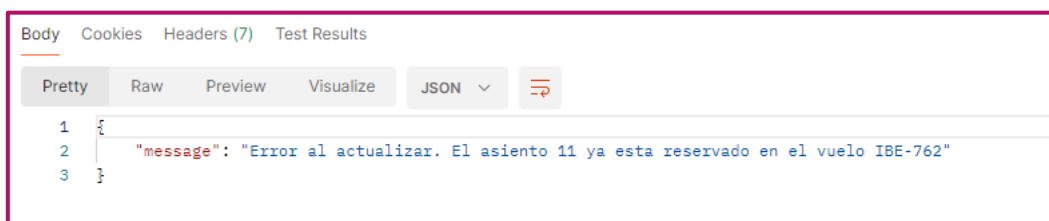


Ilustración 29 Resultado

2.3.8. Eliminar un pasaje (Obligatorio)

DELETE → Eliminar un pasaje mediante su código de pasaje.



Ilustración 30 URI para eliminar un pasaje

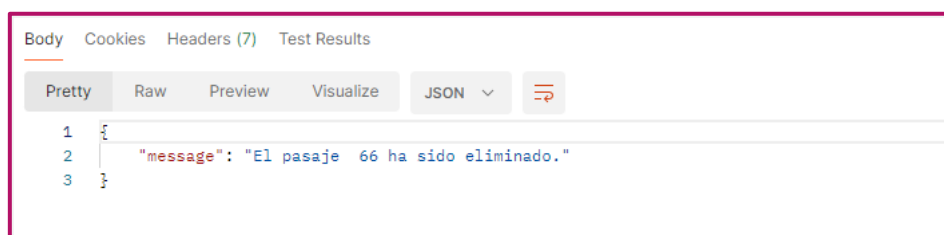


Ilustración 31 Resultado

Si vamos a la base de datos podremos ver que el pasaje se ha eliminado correctamente.

<input type="checkbox"/>	Editar	Copiar	Borrar	57	28	KLM-332	41	TURISTA	120
<input type="checkbox"/>	Editar	Copiar	Borrar	58	20	QAT-900	10	TURISTA	100
<input type="checkbox"/>	Editar	Copiar	Borrar	59	21	QAT-900	20	BUSINESS	170
<input type="checkbox"/>	Editar	Copiar	Borrar	60	22	QAT-900	25	BUSINESS	170
<input type="checkbox"/>	Editar	Copiar	Borrar	61	23	QAT-900	40	TURISTA	100
<input type="checkbox"/>	Editar	Copiar	Borrar	62	24	QAT-900	30	BUSINESS	220
<input type="checkbox"/>	Editar	Copiar	Borrar	63	25	QAT-900	35	BUSINESS	230
<input type="checkbox"/>	Editar	Copiar	Borrar	64	26	QAT-900	41	PRIMERA	120

Ilustración 32 Pasaje 66 eliminado de la base de datos

2.3.9. Error al eliminar un pasaje (Obligatorio)

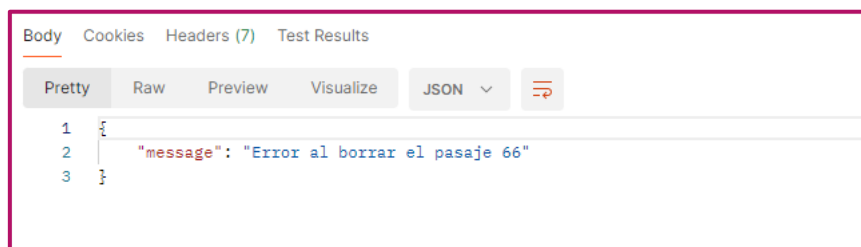
Si intentamos eliminar un pasaje que no existe nos aparecerá un mensaje de error.

Por ejemplo: Vamos a intentar eliminar el pasaje que hemos eliminado anteriormente (Eliminar un pasaje (Obligatorio)).



Ilustración 33 Eliminar pasaje que no existe

Como ya sabemos ese pasaje no existe, por lo tanto, se mostrará un mensaje de error.



```

Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "message": "Error al borrar el pasaje 66"
3 }
    
```

Ilustración 34 Mensaje de error

2.4. Pasajeros

2.4.1. Todos los pasajeros (Extra)

GET → Muestra los datos de todos los pasajeros.

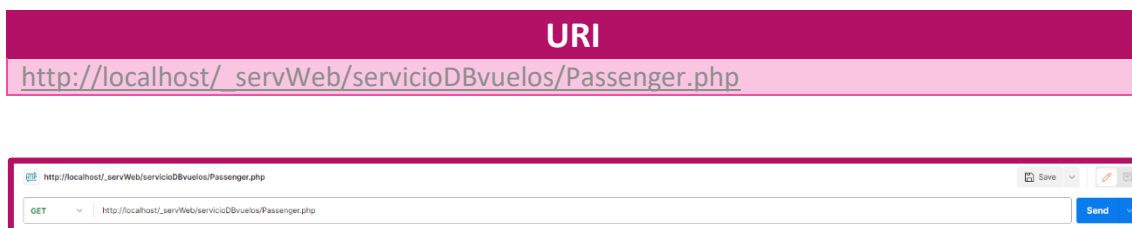
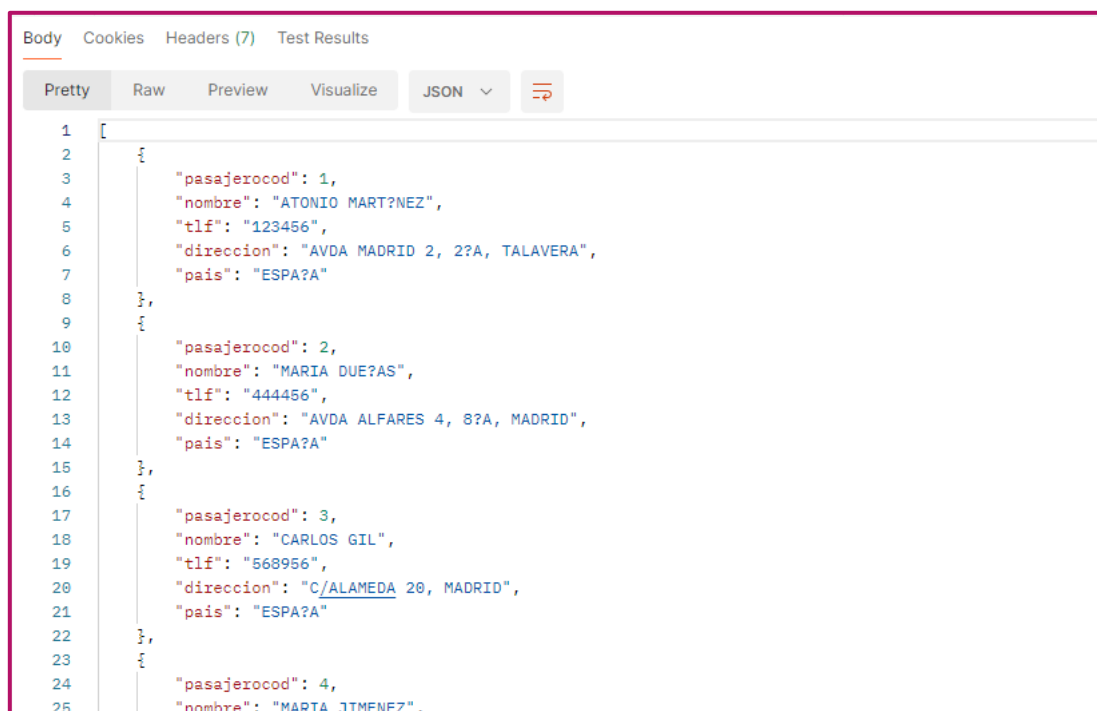


Ilustración 35 URI de todos los pasajeros



```

Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 [
2   {
3     "pasajero": 1,
4     "nombre": "ATONIO MART?NEZ",
5     "tlf": "123456",
6     "direccion": "AVDA MADRID 2, 2?A, TALAVERA",
7     "pais": "ESPA?A"
8   },
9   {
10    "pasajero": 2,
11    "nombre": "MARIA DUE?AS",
12    "tlf": "444456",
13    "direccion": "AVDA ALFARES 4, 8?A, MADRID",
14    "pais": "ESPA?A"
15  },
16  {
17    "pasajero": 3,
18    "nombre": "CARLOS GIL",
19    "tlf": "568956",
20    "direccion": "C/ALAMEDA 20, MADRID",
21    "pais": "ESPA?A"
22  },
23  {
24    "pasajero": 4,
25    "nombre": "MARIA JIMENEZ",
    
```

Ilustración 36 Resultado

2.4.2. Pasajero por código (Extra)

GET → Muestra los datos de un pasajero relacionado a un código.

URI

http://localhost/_servWeb/servicioDBvuelos/Passenger.php?pasajero=1



Ilustración 37 URI de un pasajero por código



Ilustración 38 Resultado