

BACKEND:

Se creó el archivo `trx.js`:

```
const { WebpayPlus } = require('transbank-sdk');
const { Options, IntegrationApiKeys, Environment, IntegrationCommerceCodes } =
  require('transbank-sdk'); // ES6 Modules
const Transaction = require('transbank-
sdk/dist/es5/transbank/webpay/webpay_plus/transaction');

let tx;

if (process.env.NODE_ENV === "production") {
  tx = new WebpayPlus.Transaction(new Options(IntegrationCommerceCodes.WEBPAY_PLUS,
  IntegrationApiKeys.WEBPAY, Environment.Integration));
} else {
  if (!global.__tx__) {
    global.__tx__ = new WebpayPlus.Transaction(new
Options(IntegrationCommerceCodes.WEBPAY_PLUS, IntegrationApiKeys.WEBPAY,
Environment.Integration));
  }
  tx = global.__tx__;
}

module.exports = tx;
```

Se importa el archivo para crear el ticket de vuelo y enviarlo al frontend mediante el endpoint `"flights/request/"`.

```
const ticket = await tx.create(String(purchase.id), "test-g9", amount,
"http://matiasoliva.me/purchase");
```

FRONTEND:

Mediante el endpoint `"flights/request/"` se recibe el ticket de vuelo y se extrae url y el token para realizar el pago. Este método es llamado cuando el cliente compra.

Se incluye el url entregado en un form con método POST y entregando el token como input. (Este form se manda al hacer click en un botón para pagar que se encuentra dentro del form).

```
<form action={data.url} method="POST">
  <input type="hidden" name="token_ws" value={data.token}/>
  //rest of the code
</form>
```

Se incluye además el endpoint `"flights/commit/"` donde se envía el token proveniente del ticket al backend y se recibe un mensaje de si la transacción fue exitosa o no. Este método se llama luego de finalizar la transacción.