Ah, got it — you're working with **CyberSource's SOAP-style services using JSON payloads**, typically via the **SOAP Toolkit API over JSON**.

That means you're using CyberSource's **"Unified Payments JSON API"** that mimics SOAP field names (like `ccAuthReply`, `ccCaptureReply`) but communicates over JSON.

✅ Your Use Case in JSON API Terms:

1. Authorize $1 (no capture)

2. Later: Auth + Capture $500 using stored token

🟦 Step 1: $1 Auth-Only Request

```
{
  "merchantID": "your_merchant_id",
  "merchantReferenceCode": "auth-001",
  "card": {
    "accountNumber": "4111111111111111",
    "expirationMonth": "12",
    "expirationYear": "2030",
    "cvNumber": "123"
  },
  "purchaseTotals": {
    "currency": "USD",
    "grandTotalAmount": "1.00"
  },
  "ccAuthService": {
    "run": "true"
  },
  "tokenizationService": {
    "run": "true"
  }
}
```

You'll receive:

- `ccAuthReply.authorizationCode`
- `ccAuthReply.requestID`
- `tokenInformation.transientToken` or `subscriptionID` for future charges

You can void the $1 later if desired, using `ccVoidService`.

🟦 Step 2: New Auth + Capture for $500 Using the Token

This is not a capture of the $1 — it's a **new charge**, but tied to the original card via `subscriptionID`.

```
{
  "merchantID": "your_merchant_id",
  "merchantReferenceCode": "charge-500",
  "purchaseTotals": {
    "currency": "USD",
    "grandTotalAmount": "500.00"
  },
  "ccAuthService": {
    "run": "true"
  },
  "ccCaptureService": {
    "run": "true"
  },
  "recurringSubscriptionInfo": {
    "subscriptionID": "your_saved_token"
  }
}
```

The response will contain:

- `ccAuthReply.decision: ACCEPT`
- `ccCaptureReply.amount: "500.00"`
- `requestID`: for later voids or refunds

🧠 Key Points:

Step	Action	Notes
Step 1	`ccAuthService.run = true`	Auth $1
Step 1	`tokenizationService.run = true`	Get token for reuse
Step 2	`ccAuthService + ccCaptureService`	New full charge
Step 2	`recurringSubscriptionInfo.subscriptionID = token`	Use stored card

If you want to void the original $1 auth, just send:

```
{
  "merchantID": "your_merchant_id",
  "merchantReferenceCode": "void-auth-001",
  "ccVoidService": {
    "run": "true",
    "voidRequestID": "originalAuthRequestID"
  }
}
```

Would you like a working Node.js example using CyberSource’s SDK for this flow?